Volume : 2 | Issue : 7 | July 2013 • ISSN No 2277 - 8179

Credit Crime Detection By Using Multilayer System

Engineering

KEYWORDS : Fraud detection, data mining, anomaly detection, Credit crime detection

G ANIL KUMAR M.Tech (cse), GATES Institute of Technology, Anantapur, A.P., India

D VENKATESH Prof & DEAN , Department of CSE&IT, GATES Institute of Technology, Gooty,Anantapur,A.P.,India

KANTE RAMESH Associate Prof & HEAD , Department of CSE, GATES Institute of Technology, Gooty, Anantapur, A.P

ABSTRACT

Credit Crime detection is very essential aspect of all computer applications. Especially Credit crime is much reported crime in the literature. Credit application fraud is one of the examples of Credit crime. The existing applications that do not use data mining techniques for Credit crime detection have limitations. To overcome the limitations and address the Credit crime in the real world, this paper presents a Java based, user-friendly application based on the multilayered detection approach proposed by Phua et al. The layers include CD (Communal Detection) and SD (Spike Detection). The CD algorithm can find social relationships in the dataset while the SD algorithm finds spikes in duplicates. The combination of these algorithms can detect many attacks. Our prototype application in Java shows how the Credit crime is detected. The results reveal that this application can be used in real world applications as supplement.

I. INTRODUCTION

Credit crime refers a form of stealing identity of someone and performing fraudulent activities in the name of that person (victim). In such incidents, generally, the victim suffers from unexpected consequences [1]. With modern technologies identity theft has become easy while its detection has become more and more difficult. In the real world Credit crime can occur in two ways. Stealing real identity or making a synthetic identity of other person and misuse it. The Credit crime is increase for many reasons including the availability of real identity information of people over Internet. Through unsecured mailboxes and social networking web applications, confidential data is made available. Thus the Credit crime is increased in the society. Another way round is that the perpetrators are able to hide their true identities. The domains in which the fraud can take place include telecommunications, credit, and insurance. This kind of fraud is prevalent and costly.

Stolen identity information can be used by people with malicious intentions for the purpose of payment card fraud, home equity and tax returns. Thus real consumers lose money and suffer from consequences. There are laws in developed countries to deal with such fraud cases. When organizations are subjected to Credit crimes, they suffer great damage in terms of lost customers and also economically [1]. Credit applications such as Internet based applications and paper form-based forms that capture users written requests for various monetary reasons such as personal loans, mortgage loans, and credit cards provide chances for fraudulent people to commit Credit crime. Both real and synthetic identity frauds are part of credit application fraud [8]. The patterns followed by the malicious people might change from time to time. They are persistent in committing fraud as they gain high monetary benefits from it. There might be some applications which have duplicates or share common contents. The shared content might have exactly duplicates or duplicates to some extent. In [1] it is argued that sudden spike in duplicates in very short time can represent successful credit application fraud. From fraudster's point of view it is hard to avoid duplicates as they increase their success ratio. Comparatively the synthetic identity fraudsters enjoy low success ratio while the real identity fraudsters have high success rate. This paper presents methods that support detection of Credit crime. The methods are based on the concept of white-listing in order to detect the spike in the similar applications. As social relationships are used in white-listing, it results in reducing false positives. On a set of attributes the process of detecting spikes with appropriate changes in suspicion scores can increase true positives. In case of synthetic identity fraud, the patterns obtained through data mining can provide necessary symptoms to identify crime early [4].

Any security system is subjected to tradeoffs in general and achieving resilience is an important aspect which throws some challenges [5]. The detection systems need the defense mechanisms in depth as they need to withstand a variety of attacks. The data mining based security has two specific challenges namely usage of quality data and adaptivity. When fraud behavior changes, the system has to adapt to such changing behavior. Quality of data refers to the data which has no noise or errors. There are many existing application for fraud detection. Some of them are non-data mining based while others make use of data mining techniques. However, they are not resilient in nature. This paper proposes resilience besides facing the challenges such as quality of data and adaptivity. They are achieved using communal detection and spike detection algorithms. The former is a white-list based approach while the latter is an attribute-oriented approach.

II. RELATED WORK

Fraud detection has been around for many years. Fraud behavior has been increased as the financial institutions are providing electronic payment options by issuing credit and debit cards. Banks and other such outfits are worried about possible fraud. Fraud will reduce the image of such institutions as people will be not be able to use such electronic cards. Fraud detection has been a challenging job in credit applications. Many data mining algorithms came into existence in order to detect fraud. For instance K-Means algorithm along with Hidden Markov Model can effectively build a model which can detect credit card fraud. However, the algorithms were not resilient as it was not addressed comprehensively. Lot of work on the fraud detection is proprietary in nature. For instance [6] described ID Score-Risk which gives a view of characteristics of credit applications and how they are similar to other industry provided characteristics. In other research work by name "Detect" policy rules are provided with respect to four categories to identify fraud. One such rule is to check historical data with new credit application to ensure consistency. Case Based Reasoning (CBR) [7] was also used to screen credit card applications. It can analyze hardest cases such as misclassified ones using existing techniques or methods. When compared with other algorithms, CBR has 20% higher true positives rate. The SD and CD algorithms in this paper are even better than CBR for Credit crime detection. Some algorithms which are in similar lines are Peer Group Analysis and Break Point Analysis [8]. They observe behavior of accounts over a period of time. As spending patterns change dramatically, they can detect fraud or identify the probability of fraud.

To uncover simulated anthrax attacks Bayesian networks [11] can be used as they work on the data of emergency department. A survey of all such algorithms is made in [10] which are meant for identifying suspicious activities. In order to track the symptoms of anthrax [11] used time series analysis. Many algorithms such as generalized linear models, exponential weighted moving averages and control-chart-based statistics are explored in [12] with respect to the detection of bio-terrorism. The SD algorithm implemented in this paper can be compared with Exponentially Weighted Moving Average (EWMA) with respect to performing linear forecasting.

III. IDENTITY DETECTION METHODS

This section provides information about the two algorithms that work together to detect identity fraud. The communal detection and spike detection methods are presented in this section.

Communal Detection

The need for communal detection is described here. When there are two credit applications where in same kind of records exist with very slight changes, there are three possibilities. The first possibility is that there are twin brothers whose data is same except slight change in the name. The second possibility is that a fraudster is attempting to get multiple credit cards from financial institution. Other possibility is that a person is applying twice in order to get monetary benefits. Communal Detection is an approach which can detect such scenarios. This algorithm compares data of various credit applications. It works on fixed set of attributes and uses white-list oriented approach. It finds self and communal relationships among applications. The communal relationships are nothing but records with near duplicate values on the selected attributes [13]. A white-list is constructed with entities that exhibit more probabilities of communal relationships. The algorithm takes state of alert, input size threshold, exponential smoothing factor, exact duplicate filter, attribute threshold, string similarity threshold, link-types in current white-list, moving window and current application as input and returns output as suspicion score, new parameter value and new white-list. The algorithm is as shown in listing 1.

Step 1: Find attributes that exceed string similarity threshold; create multi-attribute links against link – types in current white-list when their duplicates' similarity is more than attribute threshold.
Step 2: Using Step1's multi-attribute links calculate single link score.
Step 3: Using previous applications linked to Step1, calculate average prior scores.
Step 4: Calculate suspicion score based on the result of Step 2 and Step 3.
Step 5: Through State of Art find out new or same parameter value.
Step 6: Determine new white-list
Listing 1 – Communal detection algorithm.

As can be seen in listing 1, the communal detection algorithm whose functionality is described her. The first step is to match current application's value with other applications in order to find links using the following equation.

$$e_k = \begin{cases} 1 & \text{if } Jaro - Winkler(a_{i,k}, a_{j,k}) \geq T_{similarity} \\ 0 & \text{otherwise} \end{cases}$$

It is the first case and as per the Jaro – Winkler [14]. It is case sensitive. For non-match values, the second case is:

$$e_{i,j} = \begin{cases} e_1 e_2 \ldots e_N & \text{if } T_{attribute} \leq \sum_{k=1}^{N} e_k \leq N-1 \\ & \text{ or } [\sum_{k=1}^{N} e_k = N \\ & \text{ and } Time(a_{i,k}, a_{j,k}) \geq \eta] \\ \varepsilon & \text{otherwise} \end{cases}$$

Here the Time is the difference in time in minutes. In Step 2, single link communal detection is made using

$$S(e_{i,j}) = \begin{cases} \sum_{k=1}^{N}(e_k \times w_k) \times w_z & \text{if } e_{i,j} \in \Re_{x,link-type} \\ & \text{ and } e_{i,j} \neq \varepsilon \\ \sum_{k=1}^{N}(e_k \times w_k) & \text{if } e_{i,j} \notin \Re_{x,link-type} \\ & \text{ and } e_{i,j} \neq \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

There are three cases in this formula. The first case uses attribute weights with given values as default. The second case is the graylist link score. The third case is used if there exists no multi-attribute link. In Step 3, single link average previous score is computed as follows.

$$\beta_j = \begin{cases} \frac{S(v_j)}{E_O(v_j)} & \text{if } e_{i,j} \neq \varepsilon \\ & \text{ and } E_O(v_j) > 0 \\ 0 & \text{otherwise} \end{cases}$$

In this equation, the first case computes each previous application's average score while the second case is applied if there is no multi-attribute link. In Step 4 multiple-links score is computed as follows.

$$S(v_i) = \sum_{v_j \in K(v_i)} [S(e_{i,j}) + \beta_j]$$

Here it computes score of every current application using previous application score and every link present over there. The Step 5 computes parameter's value change as follows.

$$\text{SoA} = \begin{cases} \text{low} & \text{if } q \geq T_{input} \text{ and } \Omega_{x-1} \geq \Omega_{x,y} \\ & \text{ and } \delta_{x-1} \geq \delta_{x,y} \\ \text{high} & \text{if } q < T_{input} \text{ and } \Omega_{x-1} < \Omega_{x,y} \\ & \text{ and } \delta_{x-1} < \delta_{x,y} \\ \text{medium} & \text{otherwise} \end{cases}$$

It has three cases. In the first case the SoA is set to low when suspiciousness of output is low. In the second case the SoA is set to high when it experiences conditions which are opposite to the first case. Medium is the value set to SoA in the last case. In Step 6 tamper-resistance is improved by constructing new white-list.

Spike Detection

The spike detection process is required in order to improve adaptivity and also resilience of the proposed solution for Credit crime detection. Communal detection has a limitation in the form of attribute threshold. The spike detection complements communal detection which providing attribute weights. Entry of new applications can also be adapted using spike detection. This algorithm takes current application, other applications, current step, string similarity threshold, time difference filter, and exponential smoothing factor as input and returns output as suspicion score and attribute weights. The algorithm is as given in listing 2.

Step 1: Match current value with previous values
Step 2: Based on Step 1's matches, compute current value's score
Step 3: Calculate multiple-values score
Step 4: Find suitable SD attributes
Step 5: Determine the attribute weights for CD
Listing 2 – Spike Detection Algorithm

As can be seen in listing 2, the spike detection algorithm has five important steps. In Step 1 it matches present value with previous values.

$$a_{i,j} = \begin{cases} 1 & \text{if } Jaro - Winkler(a_{i,k}, a_{j,k}) \geq T_{similarity} \\ & \text{ and } Time(a_{i,k}, a_{j,k}) \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

The first case uses Jaro – Winkler as proposed in [14] while the second case is for non-match. In the Step 2 performs single value spike detection as follows.

$$S(a_{i,k}) = (1-\alpha) \times s_t(a_{i,k}) + \alpha \times \frac{\sum_{\tau=1}^{t-1} s_\tau(a_{i,k})}{t-1}$$

In Step 3 multiple-values score is computed as follows.

$$S(v_i) = \sum_{k=1}^{N} S(a_{i,k}) \times w_k$$

SD attribute selection is made in Step 4 as follows.

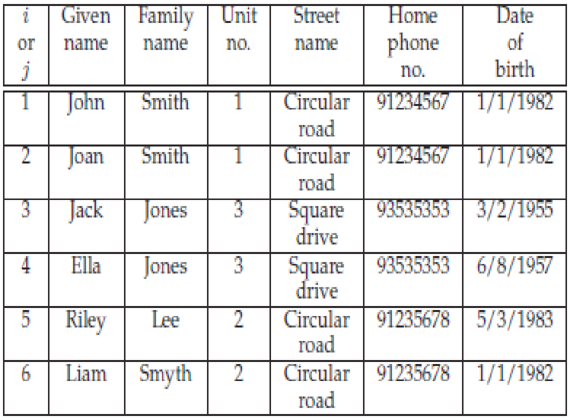
$$w_k = \begin{cases} 1 & \text{if } \frac{1}{2\times N} \leq \frac{\sum_{i=1}^{p\times q} S(a_{i,k})}{i\times\sum_{k=1}^{N} w_k} \\ & \leq \frac{1}{N} + \sqrt{\frac{1}{N} \times \sum_{k=1}^{N}(w_k - \frac{1}{N})^2} \\ 0 & \text{otherwise} \end{cases}$$

Average density of each attribute is computed in first case and finally retains weights of the best attributes. In Step 5 attributes weight change of CD is computed as follows.

$$w_k = \frac{\sum_{i=1}^{p\times q} S(a_{i,k})}{i \times \sum_{k=1}^{N} w_k}$$

IV. EXPERIMENTAL RESULTS

The environment used to develop the proposed prototype application is JDK 1.6, NetBeans IDE, a PC with 2GB RAM and Core 2 Dual processor. The dataset used for experiments is as shown in table 1. It has 6 attributes pertaining to 6 credit applications.



i or j	Given name	Family name	Unit no.	Street name	Home phone no.	Date of birth
1	John	Smith	1	Circular road	91234567	1/1/1982
2	Joan	Smith	1	Circular road	91234567	1/1/1982
3	Jack	Jones	3	Square drive	93535353	3/2/1955
4	Ella	Jones	3	Square drive	93535353	6/8/1957
5	Riley	Lee	2	Circular road	91235678	5/3/1983
6	Liam	Smyth	2	Circular road	91235678	1/1/1982

Table 1- dataset with 6 credit applications

Many experiments have been made to demonstrate the claims in this paper. They experiments are named as CD – base line, SD-baseline, CD-adaptive, SD-adaptive, No-white-list, CD-SD resilient, and CD-SD resilient best. The experiments named No-whitelist, CD-baseline, and CD-adaptive are used to demonstrate the reduction of false positives while other experiments are to demonstrate Credit crime detection with resilience. The experimental results are shown in fig. 1.

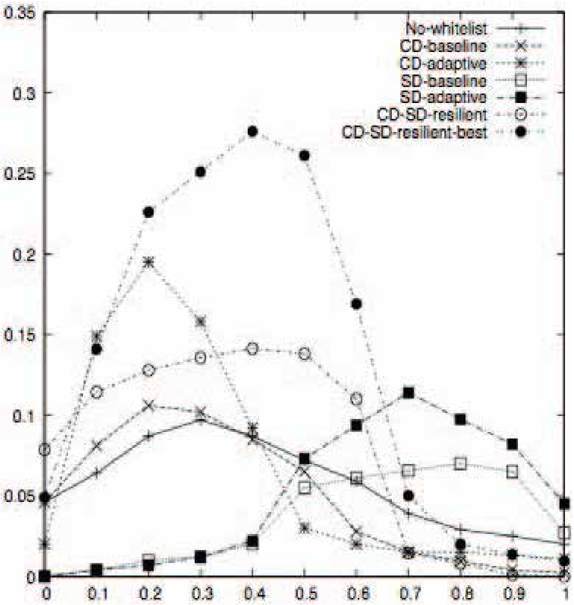


Fig. 1 – Results of CD and SD experiments Using F-measure

As can be seen in fig. 1 results are inferior without white-list. When compared with CD-baseline experiment, no-whitelist experiment performs poorly. CD Adaptive performs better than CD baseline. SD adaptive has higher performance than SD baseline.

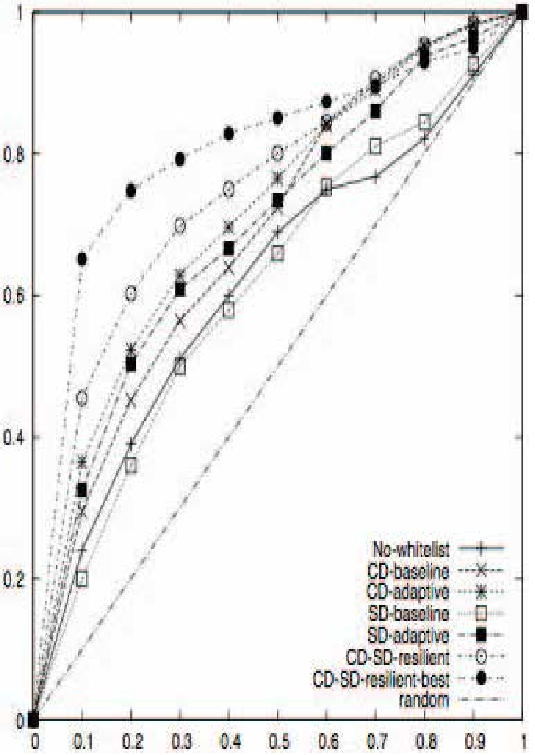


Fig. 2 – ROC Curves of CD and SD experiments

As can be seen in fig. 2, with respect to ROC curves, results are inferior without white-list. When compared with CD-baseline experiment, no-whitelist experiment performs poorly. CD Adaptive performs better than CD baseline. SD adaptive has higher performance than SD baseline.

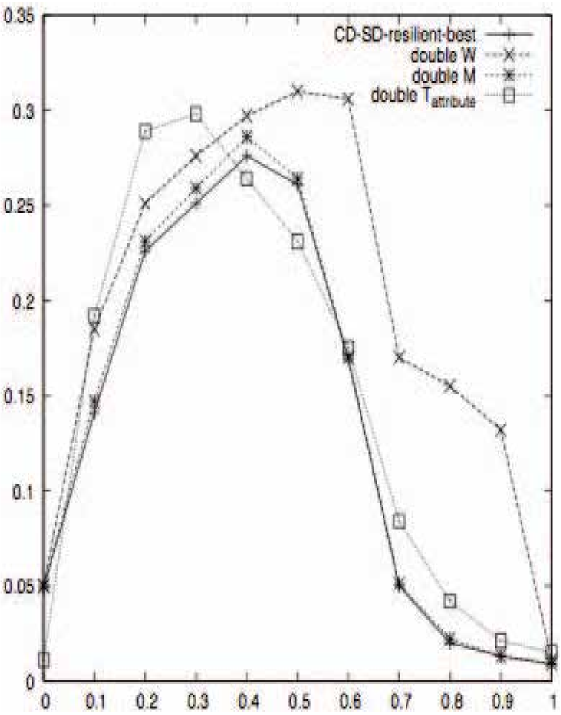


Fig. 3 – F-measure Curve of CD-SD resilient best parameters

As can be seen in fig. 3, it shows results of doubling and also CD-SD resilient best. With respect to F-measure, doubling is proved to be the most influential parameters.

V. CONCLUSION

This paper focused on robust Credit crime detection. It has implemented algorithms to safeguard applications that involve monetary transactions. It proposed prototype application has many layers of defense using data mining which can be used in the real world credit applications or for credit card fraud detection. The proposed prototype has many important concepts such as quality of data, adaptivity and multi-layered defense. The communal detection and spike detection concepts proposed by Phua et al. were used in the implementation of the Credit crime detection system. The application is tested with real and synthetic datasets. The experimental results revealed that the proposed algorithms are robust and can be used in the real world credit applications.

REFERENCE

[1] Romanosky, S., Sharp, R. and Acquisti, A. 2010. Data Breaches and Identity Theft: When is Mandatory Disclosure Optimal?, Proc. of WEIS10 Workshop, Harvard University. | [2] Gordon, G., Rebovich, D., Choo, K. and Gordon, J. 2007. Identity Fraud Trends and Patterns: Building a Data-Based Foundation for Proactive Enforcement, Center for Identity Management and Information Protection, Utica College. | [3] Clifton Phua, Member, IEEE, Kate Smith-Miles, Senior Member, IEEE, Vincent Lee, and Ross Gayler, "Resilient Credit crime Detection", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING VOL.24 NO.3 YEAR 2012. | [4] Oscherwitz, T. 2005. Synthetic Identity Fraud: Unseen Identity Challenge, Bank Security News 3: p. 7. | [5] Schneier, B. 2008. Schneier on Security, Wiley, Indiana. ISBN-10: 0470395354. | [6] IDAnalytics. 2008. ID Score-Risk: Gain Greater Visibility into Individual Identity Risk. Unpublished. | [7] Wheeler, R. and Aitken, S. 2000. Multiple Algorithms for Fraud Detection, Knowledge-Based Systems 13(3): pp. 93-99. DOI: 10.1016/S0950-7051(00)00050-2. | [8] Bolton, R. and Hand, D. 2001. Unsupervised Profiling Methods for Fraud Detection, Proc. of CSCC01. | [9] Wong, W., Moore, A., Cooper, G. and Wagner, M. 2003. Bayesian Network Anomaly Pattern Detection for Detecting Disease Outbreaks, Proc. of ICML03. ISBN: 1-57735-189-4. | [10] Wong, W. 2004. Data Mining for Early Disease Outbreak Detection, PhD thesis, Carnegie Mellon University. | [11] Goldenberg, A., Shmueli, G. and Caruana, R. 2002. Using Grocery Sales Data for the Detection of Bio-Terrorist Attacks, Statistical Medicine. | [12] Jackson, M., Baer, A., Painter, I. and Duchin, J. 2007. A Simulation Study Comparing Aberration Detection Algorithms for Syndromic Surveillance, BMC Medical Informatics and Decision Making 7(6). DOI: 10.1186/1472-6947-7-6. | [13] Jost, A. 2004. Identity Fraud Detection and Prevention. Unpublished. | [14] Winkler, W. 2006. Overview of Record Linkage and Current Research Directions, Technical Report RR 2006-2, U.S. Census Bureau.