Resilient Data Stream Mining using Spike detection

Chandana Suresh[1], Betam Suresh[2]

[1] *Chandana Suresh pursuing M.Tech(CSE), Mother Theresa Educational Society Group of Institutions, Nunna, Vijayawada. Affiliated to JNTU-Kakinada, A.P, India*

[2] *Betam Suresh is working as Head Of Department (CSE) in Vikas Group of Institutions, Nunna, Vijayawada. Affiliated to JNTU-Kakinada, A.P, India*

***Abstract*— Crime is well known, and it may exist in any form and our proposed work focus on the Credit Card related crime which shows how effectively the crime rate can be stopped. The existing system of detection is done manually and may not give feasible results for the Banks. To limit this crime in Credit card related issues we have proposed a system where in two ways the crime rate can be reduced and is far better than the data mining detection of business rules. The two techniques that help to cut down the crime rates are Spike Detection and Community Identification or also called as Hamlet Detection. Community Identification helps find the relationships through which they can prevent the issue of credit cards. Spike Detection in this project helps the application to detect the crime being done when the transaction is being done using that specific card. Together, these techniques can detect more types of attacks, better account for changing legal behavior, and remove the redundant attributes. Various test results will be conducted to carry out the process of Crime Detection. We are performing this process on Credit Card because it best suits this, as the money on Credit Card is Banks money to which security can be given from bank side and so we are implementing the above mentioned techniques and the adequate results were captured.**

***Keywords*— Crime, Spike Detection, Hamlet Detection, data mining, Credit Card.**

I. INTRODUCTION

Data Mining is sometimes also called as knowledge discovery. Data Mining is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. We can do a lot many tasks using the Data Mining concept and few of them I will list out below,

Data mining is primarily used today by companies with a strong consumer focus - retail, financial, communication, and marketing organizations. It enables these companies to determine relationships among "internal" factors such as price, product positioning, or staff skills, and "external" factors such as economic indicators, competition, and customer demographics. And, it enables them to determine the impact on sales, customer satisfaction, and corporate profits. Finally, it enables them to "drill down" into summary information to view detail transactional data.

With data mining, a retailer could use point-of-sale records of customer purchases to send targeted promotions based on an individual's purchase history. By mining demographic data from comment or warranty cards, the retailer could develop products and promotions to appeal to specific customer segments.

For example, Blockbuster Entertainment mines its video rental history database to recommend rentals to individual customers. American Express can suggest products to its cardholders based on analysis of their monthly expenditures.

Wal-Mart is pioneering massive data mining to transform its supplier relationships. Wal-Mart captures point-of-sale transactions from over 2,900 stores in 6 countries and continuously transmits this data to its massive 7.5 terabyte Teradata data warehouse. Wal-Mart allows more than 3,500 suppliers, to access data on their products and performs data analyses. These suppliers use this data to identify customer buying patterns at the store display level. They use this information to manage local store inventory and identify new merchandising opportunities. In 1995, Wal-Mart computers processed over 1 million complex data queries.

As we have seen above the importance of Data Mining in real time scenario. In the same manner the bank also does implement the concept of Data Mining i.e., it has a very huge database which contains all the details of the customers and also trading related data which is concerned to the bank. With

this large data in its database it is able to do the process i.e. fetch the data for a particular customer and perform his requested transaction efficiently. Bank has got two sectors dealing with the money i.e. one which is the people's money it saves and does the transaction as per their request second is the bank's money which is given to the user in the form of Credit Card. Our main focus is on the bank's money i.e. to give security to the bank for its card which when gone to wrong user should stop the transaction with not much of the usage. Security is given to the card with spike detection and the community detection comes into picture where the card for a particular user is applied.

Credit cards all have to deal with the identity detection i.e. whether a card is being used by the correct user or not. Credit applications are Internet or paper-based forms with written requests by potential customers for credit cards, mortgage loans, and personal loans. Credit application fraud is a specific case of identity crime, involving synthetic identity fraud and real identity theft.

As in identity crime, credit application fraud has reached a critical mass of fraudsters who are highly experienced, organized, and sophisticated. Their visible patterns can be different to each other and constantly change. They are persistent, due to the high financial rewards, and the risk and effort involved are minimal. Based on anecdotal observations of experienced credit application investigators, fraudsters can use software automation to manipulate particular values within an application and increase frequency of successful values. Duplicates (or matches) refer to applications which Share common values. There are two types of duplicates: exact (or identical) duplicates have the all same values; near (or approximate) duplicates have some same values (or characters), some similar values with slightly altered spellings, or both. This paper argues that each successful credit application fraud pattern is represented by a sudden and sharp spike in duplicates within a short time, relative to the established baseline level.

It will be shown later in this paper that many fraudsters operate his way with these applications and that their characteristic pattern of behavior can be detected by the methods reported. In short, the new methods are based on white-listing and detecting spikes of similar applications. White-listing uses real social relationships on a fixed set of attributes. This reduces false positives by lowering some suspicion scores. Detecting spikes in duplicates, on a variable set of attributes, increases true positives by adjusting suspicion scores appropriately. Throughout this paper, data mining is defined as the real-time search for patterns in a principled (or systematic) fashion. These patterns can be highly indicative of early symptoms in identity crime, especially synthetic identity fraud.

Main challenges for detection systems:

Resilience is a term which mostly suits for an application and it is expected that the application which is developed it must be resilient. The above statement means that the application must be flexible as per the requirement.

Resilience is the ability of a material to absorb energy when it is deformed elastically, and release that energy upon unloading. The system designed should be flexible for the various attacks that can be encountered in the process.

Adaptivity accounts for morphing fraud behavior, as the attempt to observe fraud changes its behavior. But what is not obvious, yet equally important, is the need to also account for changing legal (or legitimate) behavior within a changing environment. In the credit application domain, changing legal behavior is exhibited by communal relationships (such as rising/falling numbers of siblings) and can be caused by external events (such as introduction of organizational marketing campaigns). This means legal behavior can be hard to distinguish from fraud behavior, but it will be shown later in this paper that they are indeed distinguishable from each other.

Existing identity crime detection systems

There are non-data mining layers of defense to protect against credit application fraud, each with its unique strengths and weaknesses. The first existing defense is made up of business rules and scorecards. In Australia, one business rule is the hundred-point physical identity check test which requires the applicant to provide sufficient point-weighted identity documents face-to-face. They must add up to at least one hundred points, where a passport is worth seventy points. Another business rule is to contact (or investigate) the applicant over the telephone or Internet. The above two business rules are highly effective, but human resource intensive. To rely less on human resources, a common business rule is to match an application's identity number, address, or phone number against external databases. This is convenient, but the public telephone and address directories, semi-public voters' register, and credit history data can have data quality issues of accuracy, completeness, and timeliness.
In addition, scorecards for credit scoring can catch a small percentage of fraud which does not look creditworthy; but it also removes outlier applications which have a higher probability of being fraudulent. The second existing defense is known fraud matching. Here, known frauds are complete applications which were confirmed to have the intent to defraud and usually periodically recorded into a blacklist. Subsequently, the current applications are matched against the blacklist. This has the benefit and clarity of hindsight because patterns often repeat themselves. However, there are two main problems in using known frauds. First, they are untimely due to long time delays, in days or months, for fraud to reveal it, and be reported and recorded. This provides a window of opportunity for fraudsters. Second, recording of frauds is highly manual. This means known frauds can be incorrect, expensive, and difficult to obtain, and have the potential of breaching privacy. In the real-time credit application fraud detection domain, this paper argues against the use of classification (or supervised) algorithms which use class labels. In addition to the problems of using known frauds, these algorithms, such as logistic regression, neural networks, or Support Vector Machines (SVM), cannot achieve

scalability or handle the extreme imbalanced class in credit application data streams. As fraud and legal behavior changes frequently, the classifiers will deteriorate rapidly and the supervised classification algorithms will need to be trained on the new data. But the training time is too long for real-time credit application fraud detection because the new training data has too many derived numerical attributes (converted from the original, sparse string attributes) and too few known frauds. This paper acknowledges that in another domain, real-time credit card *transactional* fraud detection, there are the same issues of scalability, extremely imbalanced classes, and changing behavior.

Background work:

Algorithms Used in the Implementation:

1. Hamlet or Communal Detection
2. Spike Detection

Hamlet Detection:

This algorithm helps for the bank when there are applications from the users i.e. it is used to check the duplicity of the users i.e. either by changing their name or mobile number. The problem is that when the name is nearby same for pronouncing to the already applied name from the same address, same home phone number and same locality then there could be a chance that the user might be trying to do some fraud with the card. Here in this process we need to whitelist the users whose data is not at all matching with the other data of the users. In case if the data is matching then we need to blacklist that user and go for the manual verification which is a step ahead than normal communal detection.

To account for legal behavior and data errors, Communal Detection (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, neighbours, or friends. The family member relationship can be further broken down into more detailed relationships such as husband-wife, parent-child, brother-sister, and male-female cousin (or both male, and 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 behaviour (for simplicity, self-relationships are regarded as communal relationships).

This algorithm to design takes nine inputs, gives three outputs in six steps i.e.,

Inputs

v_i (Current application)
W number of v_j(moving window)
$R_{x,link-type}$(Link-types in current whitelist)
$T_{similarity}$(String similarity threshold)
$T_{attribute}$(Attribute threshold)
η (exact duplicate filter)
α (exponential smoothing factor)
T_{input} (Input size threshold)
SoA (State-of-Alert)

Outputs

$S(v_i)$ (suspicion score)
Same or new parameter value
New whitelist

CD algorithm

Step 1: Multi-attribute link [match v_i against W number of v_j To determine if a single attribute exceeds $T_{similarity}$; and create multi-attribute links if near duplicates' similarity exceeds $T_{attribute}$or an exact duplicates' time difference exceeds η]

Step 2: Single-link score [calculate single-link score by matching
Step 1's multi-attribute links against, $R_{x,link-type}$

Step 3: Single-link average previous score [calculate average previous scores from Step 1's linked previous applications]

Step 4: Multiple-links score [calculate S (vi) based on weighted average (using α) of Step 2's link scores and Step 3's average previous scores]

Step 5: Parameter's value change [determine same or new parameter value through SoA (for example, by comparing input size against *Tinput*) at end of *ux,y*]

Step 6: Whitelist change [determine new whitelist at end of *gx*]

I o r j	Given name	Family name	Unit no.	Street name	Home phone	DOB
1	Joan	Smith	1	Cross road	9009134567	1/1/1986
2	John	Smith	1	Cross road	9009134567	2/1/1986
3	Jack	Jones	2	Circular road	9292001234	3/2/1967
4	Ella	Jones	2	Circular road	9292001234	2/4/1980
5	Ram	Williams	3	Square road	8989891234	4/4/1954
6	Sham	Williams	3	Square road	8989891234	3/4/1978

Fig. 1 Sample of 6 credit applications with 6 attributes

Above table provides a sample of 6 credit applications with 6 attributes, to show how communal relationships are extracted from credit applications. With the above six steps we will create a white list of the users.

From the above data we can white list the users whose data is genuine i.e. if the data is not matching with the other user who has applied for the card with the same bank.

Spike Detection:

This algorithm is applicable on the list where the users are provided with the credit card. In this algorithm we are going to track the usage details and accordingly we are going to decide some rules for that user for his credit card usage.

Spike Detection algorithm is explained as follows,

Inputs

v_i (Current application)
W number of v_j(moving window)
t (current step)
$T_{similarity}$(String similarity threshold)
θ (time difference filter)
α (Exponential smoothing factor)

Outputs

$S(v_i)$ (suspicion score)
w_k(Attribute weight)

SD algorithm

Step 1: Single-step scaled counts [match *vi* against *W* number of *vj* to determine if a single value exceeds *Tsimilarity* and its time difference exceeds θ]

Step 2: Single-value spike detection [calculate current value's score based on weighted average (using α) of *t* Step 1's scaled matches]

Step 3: Multiple-values score [calculate *S*(*vi*) from Step 2's value scores and Step 4's *wk*]

Step 4: SD attributes selection [determine *wk* for SD at end of *gx*]

Step 5: CD attribute weights change [determine *wk* for CD at end of *gx*]

The details of the above five steps is discussed below.

Step 1: Single-step scaled counts

$$a_{i,j} = \begin{cases} 1 \; if \; ai,k \; aj,k > T \; and \; Time \geq \\ 0 \; otherwise \end{cases}$$

where ai,j is the single-attribute match between the current value and a previous value. The first case uses Jaro-Winkler, which is case sensitive, and can also be cross-matched between current value and previous values from another similar attribute, and Time which is the time difference in minutes. The second case is a non-match because the values are not similar, or recur too quickly.

Step 2: Single-value spike detection

The second step of the SD algorithm is the calculation of every current value's score by integrating all steps to find spikes. The previous steps act as the established baseline level.

$$S_{(a_{i,k})} = (1-\alpha) \times s_t(a_{i,k}) + \alpha \times \frac{\sum_{T=1}^{t-1} s_T(a_{i,k})}{t-1}$$

Step 3: Multiple-values score

The third step of the SD algorithm is the calculation of every current application's score using all values' scores and attribute weights.

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

Step 4: SD attributes Selection

At the end of every current Mini-discrete data stream, the fourth step of the SD algorithm selects the attributes for the SD suspicion score. This also highlights the probe-reduction of selected attributes.

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

Step 5: CD attribute weights change

At the end of every current Mini-discrete data stream, the fifth step of the SD algorithm updates the attribute weights for CD.

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

Where wk is the weight applied to the community detection attributes.

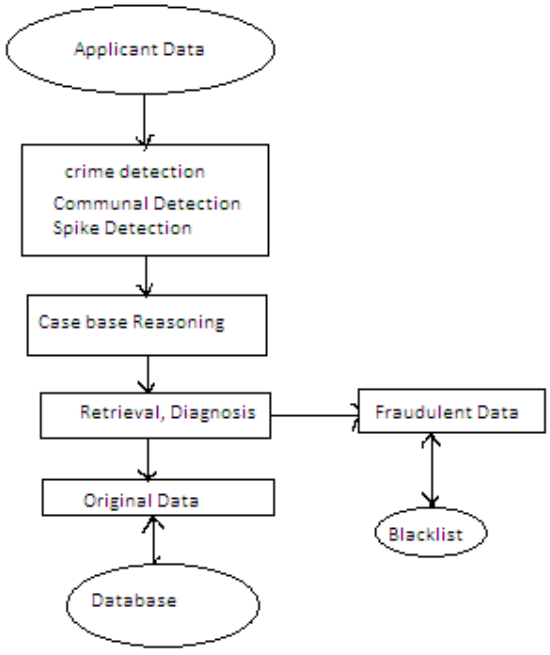
Therefore the above five steps shows the implementation process of Spike Detection. Spike Detection helps bank to stop the wrong transaction or an illegal transaction being done by an unauthenticated user.

We are setting up some rules where the user cannot proceed further if the transaction is not as per the regulations of the bank. We are mainly putting a condition i.e.

1. We are allowing a user to do transactions using the card allotted to him and meanwhile the bank is mining the purchases.

2. Then using Spike detection bank is calculating an average limit for a user and setting some upper boundary for that card.
3. If the limit of purchase is crossing the upper boundary then the transaction will be blocked and the user need to go for verification purpose i.e. the rule set by bank software using Spike detection.

With the above process the application is saving the bank money by going into wrong user hand. By this we can say that both CD and SD will help the organization to detect the crime related with the Credit Cards.



System Architecture Diagram

Conclusion:

The system detects the fraud detection online credit card application. This system is used to avoid the duplicates from the fraudsters while applying the credit card and also to save the bank money. Data mining algorithms used in this system are communal detection and spike detection which is used to detect the fraud in Credit Cards. We have worked on this concept and also made some test practices which is giving us appropriate result as expected. So it is suggested to go with both the data mining algorithms for best output where we can stop the online fraud for those applying the credit cards repeatedly and also the credit card usage by an unauthenticated user.

REFFRENCES

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

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

P. Brockett, R. Derrig, L. Golden, A. Levine, and M. Alpert, "Fraud Classification Using Principal Component Analysis of RIDITs,"The J. Risk and Insurance, vol. 69, no. 3, pp. 341-371, 2002, doi: 10.1111/1539-6975.00027.

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 Discoveryand Data Mining (KDD '04), 2004, doi: 10.1145/1014052.1014063.

P. Christen and K. Goiser, "Quality and Complexity Measures for Data Linkage and Deduplication," Quality Measures in DataMining, F. Guillet and H. Hamilton, eds., vol. 43, Springer, 2007, doi: 10.1007/978-3-540-44918-8.

C. Cortes, D. Pregibon, and C. Volinsky, "Computational Methods for Dynamic Graphs," Computational and Graphical Statistics,vol. 12, no. 4, pp. 950-970, 2003, doi: 10.1198/1061860032742.

Experian. Experian Detect: Application Fraud Prevention System, Whitepaper, http://www.experian.com/products/pdf/experian_detect.pdf, 2008.

T. Fawcett, "An Introduction to ROC Analysis," Pattern Recognition Letters, vol. 27, pp. 861-874, 2006, doi: 10.1016/j.patrec. 2005.10.010.

AUTHORS

Chandana Suresh

Pursuing M.Tech (CSE) Mother Theresa Educational Society Group of Institutions, Nunna, Vijayawada. Affiliated to JNTU-Kakinada, A.P. India

Betam Suresh

Working as Head of Department CSE, Vikas Group of Institutions, Nunna, Vijayawada, Affiliated to JNTU-Kakinada, A.P. ,India