International journal of Computer Science & Network Solutions November.2013-Volume 1.No3
http://www.ijcsns.com ISSN 2345-3397

A Secure Mechanism for Resilient of Data Mining-Based Fraud Detection

A. K.Rachel Praveena, B. Dr. G. Venkata Rami Reddy, C. K. Suresh Babu, D. G. Sudhakar[4]
M.TECH (CSE), School of IT, JNTUH,Hyderabad, India,
k.praveena3289@gmail.com
Associate Professor in CSE,School of IT,JNTUH,Hyderabad,India,
gvr.reddi@yahoo.co.in
Assistant Professor in CSE, School of IT, JNTUH, Hyderabad, India,
kare.suresh@yahoo.co.in
Lecturer in CSE, School of IT, JNTUH,Hyderabad, India,
sudhakar4321@gmail.com

Abstract

Identity Crime is used to detect fraudulence in credit card. The synthetic identity fraud utilizes credible but incorrect identities that are simple to create but more complicated to be appropriate on real time. Identity crime is completed in the combination of both synthetic as well as real identity theft. Credit Crime detection is extremely important characteristic of every computer applications. Particularly Credit crime is a lot reported crime in the literature. Credit application fraud is one of the examples of Credit crime. The obtainable applications do not apply data mining techniques intended for Credit crime detection has restrictions. To overcome the restrictions as well as address the Credit crime in the real world, this paper presents a Java based and user-friendly application based on top of the multilayered detection approach proposed by Phua et al. The layers consist of CD (Communal Detection) as well as SD (Spike Detection). The CD algorithm knows how to find social relationships inside the dataset whereas the SD algorithm finds spikes inside duplicates. Both communal detection as well as spike detection become aware of more types of attacks, enhanced account for changing legal behaviour, as well as remove the redundant attributes. The grouping of these algorithms knows how to detect several attacks. Our prototype application in Java illustrates how the Credit crime is detected. The results disclose that this application can be used in real world applications at the same time as supplement.

Keywords*:* Data mining based fraud, Anomaly detection, protection and data stream mining.

I. INTRODUCTION

Credit crime refers a form of theft identity of someone in addition to performing fraudulent behavior in the name of that person (victim). In such event, usually, the victim suffers from sudden consequences (Romanosky et al,2010). With modern technologies identity theft has develop into easy at the same time as its detection has become supplementary and more complicated. In the real world Credit crime can arise in two ways theft real identity or else making a synthetic identity of other person and abuse it. The Credit crime is augment for lots of reasons as well as the availability of real identity information of people over Internet. Through unsecured mailboxes along with social networking web applications, confidential data is made obtainable. Therefore the Credit crime is increased in the society. Another way is that the perpetrators are able to conceal their accurate identities. The domains in which the fraud can include telecommunications, credit, as well as insurance. This type 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 as well as tax returns. Therefore real consumers lose money moreover suffer from consequences. There are laws in developed countries to deal with such fraud cases. When organizations are subjected to Credit crimes, they experience great damage in terms of lost customers as well as economically (Romanosky et al,2010). Credit applications such as Internet based applications moreover paper form-based forms that capture users written requests for a variety of monetary reasons such as personal loans, mortgage loans, also credit cards offer chances for fraudulent people to commit Credit crime. Together real as well as synthetic identity frauds are part of credit application fraud (Bolton et al,2001). The patterns followed by the malicious people might alter from time to time. They are persistent in committing scam as they increase high monetary profit from it and also some applications which have duplicates or else share common contents.

The shared content may include exactly duplicates or else duplicates to various extents. In (Romanosky et al,2010) it is argued that unexpected spike in duplicates in extremely short time can represent successful credit application fraud. From fraudster's point of view it is difficult to avoid duplicates as they increase their achievement ratio. Comparatively the synthetic identity fraudsters enjoy low success ratio at the same time as the real identity fraudsters contain high success rate. This paper presents methods that support detection of Credit crime. The methods are based on the idea of white-listing in order to identify the spike in the similar applications. While social relationships are used in white-listing, it results in sinking false positives. On a set of attributes the process of detecting spikes with proper changes in suspicion scores can increase accurate positives. In case of synthetic identity fraud, the patterns obtained through data mining can offer required symptoms to identify crime early (Oscherwitz, 2005). Any security system is subjected to tradeoffs in general moreover achieving resilience is an significant aspect which throws some challenges (Schneier,2008). The detection systems require the protection mechanisms in depth at the same time as they need to withstand a variety of attacks. The data mining based protection has two specific challenges namely usage of quality data as well as adaptivity. When fraud behavior modifies, the system has to adapt to such changing behavior. Quality of data refers to the data which has no noise or else errors. There are many existing application used for fraud detection. Some of them are non-data mining based whereas others make use of data mining techniques; nevertheless they are not resilient in nature. This paper proposes resilience in addition to facing the challenges such as quality of data as well as adaptively. They are achieved by means of communal detection and spike detection algorithms. New methods are based on white-listing along with detecting spikes of related applications. White listing uses real social relationships on a fixed set of attributes. This reduces false positives by lowering a few suspicion scores. Detecting spikes in duplicates, on a variable set of attributes, enhances the true positives by adjusting suspicion scores properly. Throughout this paper, data mining is defined as the real-time search for patterns during a principled fashion. These patterns can be greatly indicative of early symptoms in identity crime, especially synthetic identity fraud.

Most important Challenges designed for Detection Systems are:
The two most advanced challenges for the data mining-based layers of defense are adaptively along with quality data. These challenges want to be addressed in order to condense fake positives.

Adaptively accounts for morphing rip-off behavior, as the challenge to observe fraud changes its behavior. But what is not noticeable, however equally essential, is the need to also account for unstable legal (or legitimate) behavior enclosed by a changing environment. In the credit application domain, varying legal behavior is exhibited by communal relationships (for example rising/falling numbers of siblings) furthermore it can be caused by peripheral events (for example introduction of organizational marketing campaigns). This way authorized behavior can be rigid to decide from fraud behavior. The detection system requests to work out carefulness by means of applications which reproduce communal relationships. It also needs to make allowance for certain peripheral actions.

Quality data are tremendously attractive for data mining along with data quality can be enhanced all the way through the real time removal of data errors (or noise). The detection system has to clean duplicates which have been reentered due to human error or else for other reasons. It also desires to disregard unnecessary attributes which have numerous missing values, as well as additional issues.

II. DATA MINING IN FRAUD DETECTION

For fraud detection, the client presently had a dissimilar data storage area somewhere model scoring was achieved. Based on the model score, information would be queried next to the data warehouse to produce the claims that were assumed. This process was disorganized for a number of reasons:

- Fraud detection investigation had to be conducted by a expert who approved the scores against the fraud detection group. The group achieves the investigation with evaluated qualities of the claims.

This was a incoherent procedure whereby the fraud detection data mining scores were not being improved based on search cost.

- Fraud detection was not accessible to unexpected alter during claim patterns. In favor of natural failure events, such as hurricanes, there would be a spike during related claims. The data mining score would not be capable to get used to such scenarios.
- Data mining was restricted to actuarial specialists also not day-to-day managers.

Primarily DWreview examined the text mining clarification provided by the most important data mining vendors. Whereas these verify to be well-off in features as well as they were awkward to use and lacked the assurance for ongoing daily usage. It became soon noticeable that it would not be cost useful to execute such solutions for the client's needs. A scripting engine was considered and urbanized for the data removal layer. The data removal layer was measured to pull reports, either manually or else a planned assignment, on or after the data warehouse repositories. The scripting words used with the data removal module are Perl, which makes the removal module enormously accessible for constructing changes by end-users.

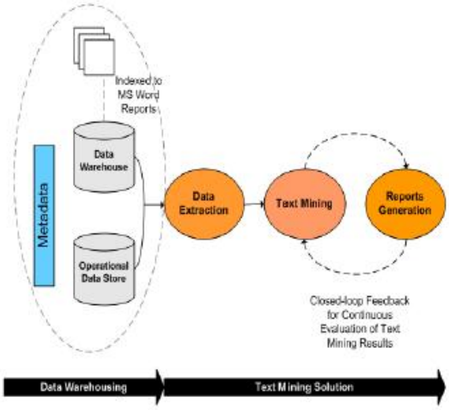


Figure.1. Text Mining modules

The text mining module includes the data mining scores based on conventional investigation of likelihood of fraud. The algorithms were convention developed based on text entered in the claims examiner's intelligence also information based on the claim. This model was developed mainly for the client by DWreview. The data mining representation can provide the client a determined advantage moreover the technical information is reserved as a intimately protected corporate secret.

III. BACKGROUND

Fraud detection has been around for many years, fraud behavior has been improved as the financial institutions are providing electronic payment options by issuing credit as well as debit cards. Banks and other such outfits are concerned about possible fraud. Fraud will decrease the image of such institutions as people will be not being able to use such electronic cards. Fraud detection has been a difficult job in credit applications. A lot of data mining algorithms came into subsistence in order to identify fraud. For instance K-Means algorithm along with Hidden Markov Model can successfully build a model which can identify credit card fraud. However, the algorithms were not resilient since it was not addressed expansively. Lot of effort on the fraud detection is proprietary in nature. For instance (IDAnalytics , 2008) described ID Score-Risk which gives a view of characteristics of credit applications and how they are similar to other industry provided characteristics. In further research work by name “Detect” policy rules are provided with respect to four categories to identify fraud. One such rule is to verify historical data with original credit application to ensure consistency. Case Based Reasoning (CBR) (Wong et al,2004) was also used to screen credit card applications. It can investigate difficult cases such as misclassified ones using accessible techniques or else methods. When compared with further algorithms, CBR has 20% higher true positives rate. In this paper SD and CD algorithms are even better than CBR for Credit crime detection. Several algorithms which are in similar lines for Peer Group Analysis and Break Point Analysis [2]. They examine behavior of accounts over a period of time as spending patterns change considerably and they can detect fraud or else identify the probability of fraud.

To uncover simulated anthrax attacks Bayesian networks (Goldenberg et al,2002) can be utilize as they effort on the data of emergency department. A survey of all such algorithms is made in (Wong et al,2004) which are meant for identifying suspicious activities. In order to track the symptoms of anthrax (Goldenberg et al,2002) used time series analysis. Many algorithms such as generalized linear models, exponential weighted moving averages along with control-chart-based statistics are explored in (Jackson et al,2007) 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.

IV. SYSTEM ARCHITECTURE

The architecture diagram (Fig 2) represents the overall structure of the system. The data is identified for the crime detection by means of the data mining algorithm communal detection as well as spike detection algorithm. These two algorithms come together to eliminate the negative false furthermore proceeded to the proposed system algorithm (i.e.) CBR algorithm. This algorithm retrieved along with diagnosis the datum. If the data is fraud it is thrown into the black list database and if the data is unique the data is stored in the database. The communal detection focused on attacks in the white list by fraudsters while they submit applications through synthetic relationship. The volume along with ranks of the white list's real communal relationships change over time, to construct the white list exercise warning with (more adaptive) changing legal behavior, the white list is constantly being reconstructed. The spike detection is attribute oriented.

It cannot be detected by fraud attribute will be updated frequently. The attributes used in spike detection will not be in communal detection. By using the spike detection as well as communal detection it detects the fraudsters in credit card application. In addition to communal detection as well as spike detection we use case based reasoning algorithm to make this approach more capable. CBR implements retrieval, diagnosis moreover resolution to make the data protected. The CBR used to examine with retrieval of data from the existing blacklist. The fraudulent datum is moved to the blacklist furthermore the original datum is stored in the database.

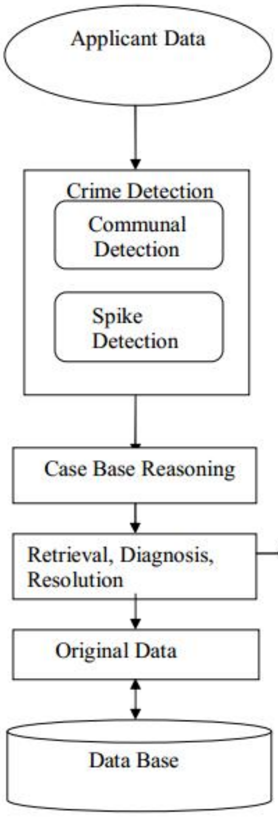


Figure.2. System Architecture Diagram

A. Crime Detection

The crime detection consists of the two algorithms, communal detection as well as spike detection. The communal detection detects the fraudsters and the detection is relationship oriented as well as attribute oriented. The spike detection detects the system fraudsters by updating the system attributes. These system finds the data whether the data is unique or not. These two detections are mostly involved in existing crime detection.

B. Finding Legitimate User

The CBR is used in the fraud detection system that the data is unique or not and the data is original or not by retrieving the data from the blacklist confirmation. This method discovers the fraudulent data by the artificial intelligence. The CBR algorithm involves with the data mining concept with match analysis

C. Blacklist Verification

With the provided sets of information are taken into consideration to keep away from the identity crime. The data is verified using the above algorithms to create the credit card application extremely efficient. If the data is original more processes will be imposed or else the data will be found as fraud and it will be enrolled in the black list.

V. 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

This algorithm helps for the bank when there are applications from the users specifically it is used to verify the duplicity of the users that is either by changing their name or else mobile number. The problem is that when the name is nearby same for pronouncing to the previously applied name from the same home phone number, same address as well as same locality then there could be a possibility that the user might be trying to do some scam with the card. Here in this method we need to white list the users whose data is not at all similar with the other data of the users. In case if the data is similar then we need to blacklist that user and go for the manual verification which is a step in advance than normal communal detection.

The need for communal detection is described here. When there are two credit applications where in similar kind of records exist with very minor changes, there are three possibilities. The first possibility is that there are twin brothers whose data is similar but slight change in the name. The second possibility is that a fraudster is attempting to get several credit cards from financial institution. Other possibility is that a person is applying double in order to get financial benefits. Communal Detection is an approach which can identify such scenarios. This algorithm compares data a variety of credit applications. It works on fixed set of attributes and it uses a white-list oriented approach. It finds self relationships as well as communal relationships between the applications. The communal relationships are nothing but records with near duplicate values on the chosen attributes (Jost, 2004). A white-list is constructed with entities that display more probabilities of communal relationships. The algorithm takes exponential smoothing factor, input size threshold, state of alert, string similarity threshold, attribute threshold, exact duplicate filter, link-types in current white-list, moving window as well as current application as input furthermore returns output as suspicion score, new parameter value and new white-list.

While Table 1 gives a summary of the CD algorithms six steps, the information in each step are presented below.

Step 1: Multi attribute link.

It finds attributes that exceed string similarity threshold; generate multi-attribute links against link types in current white-list when their duplicates' similarity is more than attribute threshold. The first step of the CD algorithm matches every current application's value against a moving window of previous applications' values to find links

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

Where ek is the single-attribute match between the current value and a previous value. The first case uses Jaro-Winkler(.) is case sensitive, cross matched linking current value as well as previous values from an additional similar attribute. The second case is a non match for the reason that values are not similar.

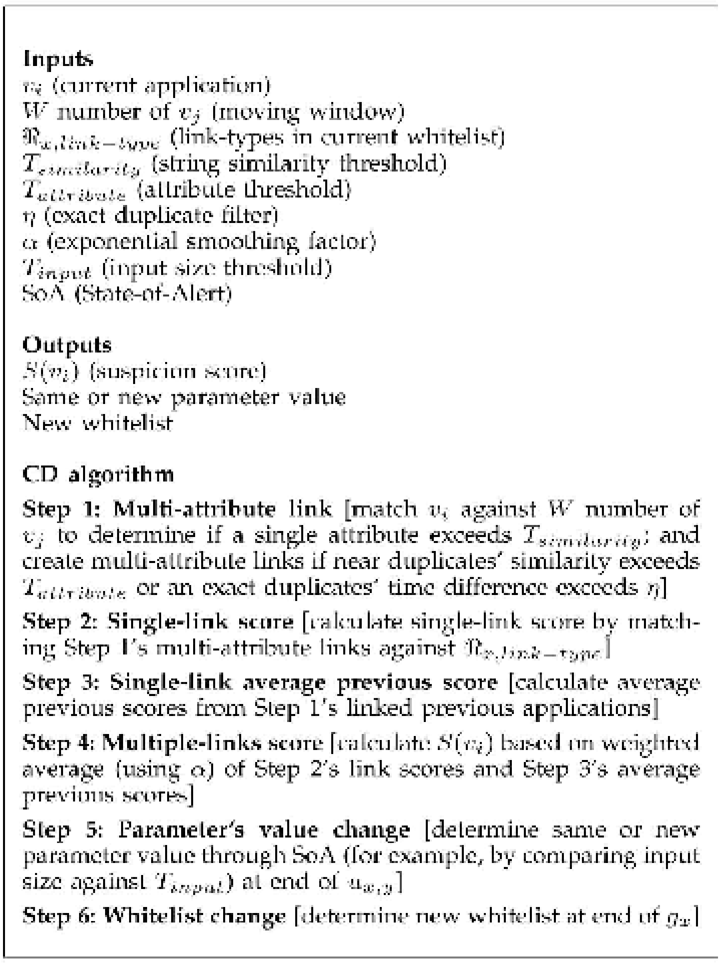
Step 2: Single-link communal detection.

Using Step1's multi-attribute links analyze single link score. The second step of the CD algorithm accounts for attribute weights moreover it matches all current application's link beside the white list to discover communal relationships furthermore it reduces their link score.

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

There are three cases in this formula. The first case uses attribute weights with given values as default. The second case is the gray list link score. The third case is used if there are no multi-attribute links.

TABLE I
Overview of Communal Detection Algorithm



Inputs
v_i (current application)
W number of v_j (moving window)
$\Re_{x,link-type}$ (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 $\Re_{x,link-type}$]

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

Step 4: Multiple-links score [calculate $S(v_i)$ 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 T_{input}) at end of $u_{x,y}$]

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

Step 3: Single-link average previous score.

Using before applications linked to Step1, analyze average prior scores. The third step of the CD algorithm is the calculation of all linked previous application's score designed for inclusion into the present application's score. The earlier scores act as the established baseline level.

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

In this equation, the first case computes each earlier application's average score while the second case is applied if there is no multi-attribute link.

Step 4: Multiple-links score.

Analyze suspicion score based on the result of Step 2 and Step 3.The fourth step of the CD algorithm is the calculation of all current application's score with every link along with previous application score.

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

Here it computes score of every current application using earlier application score and every link present over there.

Step 5: Parameter's value change.

Through State of Art find out new or same parameter value. By the end of the present micro discrete data stream, the adaptive CD algorithm concludes the State-of-Alert (SoA) moreover updates one random parameter's value such that there is a tradeoff among effectiveness through efficiency. This raises the tamper resistance during parameters.

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

The first case sets SOA to low when input size is high also output suspiciousness is low. The adaptive Communal Detection algorithm trades off one random parameter's usefulness for superior organization. For instance, a smaller moving window, fewer link types in the white list, if not a larger attribute threshold reduces the algorithm's usefulness although increase its efficiency.

Conversely, the second case sets SOA to high when its condition is the reverse of first case. The adaptive Communal Detection algorithm will trade off one random parameter's efficiency for effectiveness which develops security. The last case sets SOA to medium. The adaptive Communal Detection algorithm will not modify any parameter's value.

Step 6: White list change.

Determine new white-list, tamper-resistance is improved by constructing new white-list. By the end of the recent Mini discrete data stream, the adaptive Communal Detection algorithm creates the latest white list on the present Mini discrete stream's links. This increases the tamper-resistance within the white list.

Spike Detection

The spike detection process is essential in order to develop adaptively as well as resilience of the proposed solution for Credit crime detection. Communal detection has a restriction in the form of attribute threshold. The spike detection complements communal detection which providing attribute weights. Entry of new applications can also be modified using spike detection. This algorithm takes recent application, current step, other applications, time difference filter, string similarity threshold, moreover exponential smoothing factor as input and returns output as suspicion score along with attribute weights.

While Table 2 gives a summary of the SD algorithms five steps, the information in each step are presented below.

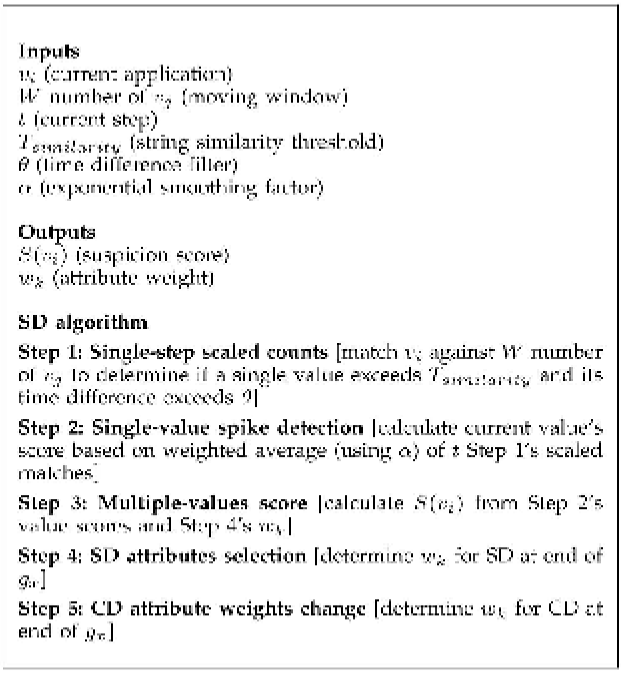
Step 1: Single-step scaled count.

The first step is to match current application's value with previous applications in order to discover links using the following equation.

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

Where ai,j is the single-attribute match between the current value with previous value. The first case uses Jaro-Winkler (.)(Gordon et al, 2007)(Clifton et al, 2012) , which is case sensitive furthermore cross-matched between current values through previous values behind an additional similar attribute. Time (.) which leftovers time difference in minutes. The second case exist a non match on behalf of the values that are not consistent or else continue too quickly.

TABLE II
Overview of Communal Detection Algorithm



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 v_i against W number of v_j to determine if a single value exceeds $T_{similarity}$ 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 $S(v_i)$ from Step 2's value scores and Step 4's w_k]

Step 4: SD attributes selection [determine w_k for SD at end of g_x]

Step 5: CD attribute weights change [determine w_k for CD at end of g_x]

Step 2: Single-value spike detection.

Based on Step 1's matches, compute current value's Score. The second step of the SD algorithm is the calculation of all single current value's score all through integrating each as well as every steps to find spikes. The before steps act at the same time as the established baseline level.

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

Where S (ai,k) is the current value score.

Step 3: Multiple-values score.

The third step of the SD algorithm is the calculation of every present application's score by means of all values' scores beside with attribute weights.

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

Where S(vi) is the SD suspicion score of the present application.

Step 4: SD attributes selection.

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

Where wk indicates Spike Detection attribute weight applied to the Spike Detection attributes. The first case is the average density of all attribute or else the sum of all value scores contained by a Mini discrete stream for one attribute, relative to the entire applications as well as attribute weights. In addition, the first case maintain only the best attributes' weights within the lower bound with upper bound, by setting redundant attributes' weights to zero. (Wheeler et al, 2000)(Wong et al, 2003)(Winkler , 2006)

Step 5: CD attribute weights change.

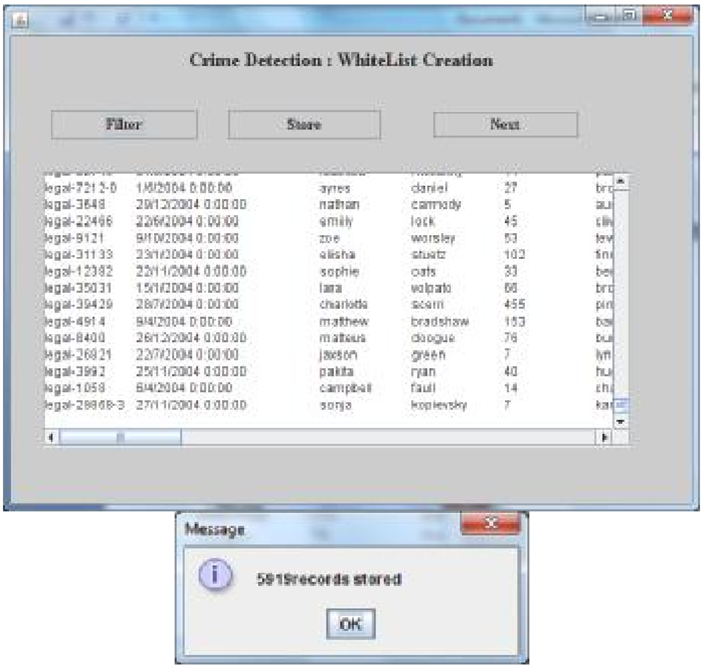
By the end of all present Mini discrete data stream, the fifth step of the SD algorithm updates the attribute weights used for CD.

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

Where wk is the SD attribute weight applied to the CD attributes.

VI. RESULTS

This is the communal-fraud-scoring-data and the file contains 21 attributes and greater than 6702 records. Next we are going to filter records that contain less number of attributes. Therefore, 5919 records are eliminated. After eliminating the false records we are going to store 5919 records into the database.



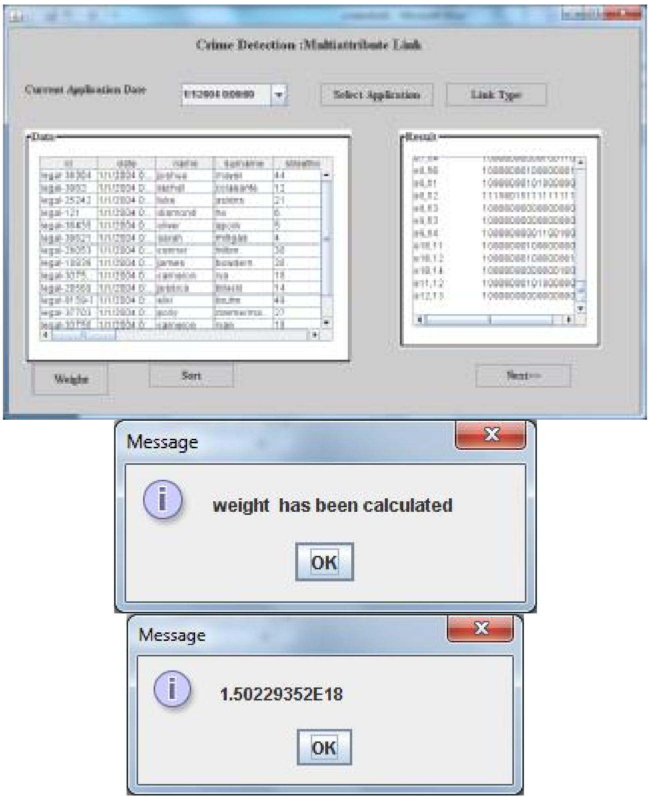
Crime Detection : WhiteList Creation

legal-7212-0	1/8/2004 0:00:00	ayres	daniel	27	bro
legal-3648	29/12/2004 0:00:00	nathan	carmody	5	au
legal-22466	22/6/2004 0:00:00	emily	lock	45	cli
legal-9121	9/10/2004 0:00:00	zoe	worsley	53	lev
legal-31133	23/1/2004 0:00:00	elisha	stuetz	102	fin
legal-12382	22/11/2004 0:00:00	sophie	oats	33	be
legal-35031	15/1/2004 0:00:00	lara	volpato	66	bro
legal-39429	28/7/2004 0:00:00	charlotte	scerri	455	pin
legal-4914	9/4/2004 0:00:00	matthew	bradshaw	153	ba
legal-8400	26/12/2004 0:00:00	matteus	doogue	76	bu
legal-26821	22/7/2004 0:00:00	jaxson	green	7	lyn
legal-3992	25/11/2004 0:00:00	pakita	ryan	40	hu
legal-1058	8/4/2004 0:00:00	campbell	faull	14	ch
legal-28868-3	27/11/2004 0:00:00	sonja	koplevsky	7	ka

Message

5919records stored

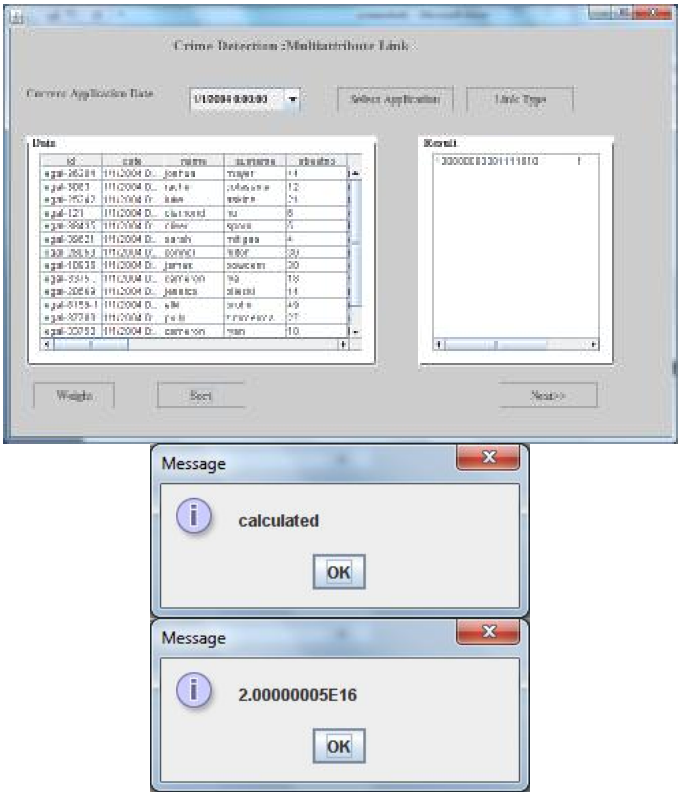
Select the date which contains only 15 records. For this date link type was generated. Link type was generated based on the similarity between these records. If the record contains high similarity then it is considered as crimes. Here if three attributes matched between two records then it is considered as a crime. Record 0 and 2 contains greater than 3 attributes similarly so it is crime. For this weight also calculated. This is the weight of the link types and this is called as white list. For this white list weight is calculated and after calculating weight, single link is generated and suspicion score is calculated. Based on suspicion score parameter are changed.



weight has been calculated

1.50229352E18

Finally spike detection (SD) also applied for this date.It contains 35 records.Link type and weight calculation was generated.Link type was generated if the record contains four attributes similarly.After calculating link type weight is calculated.

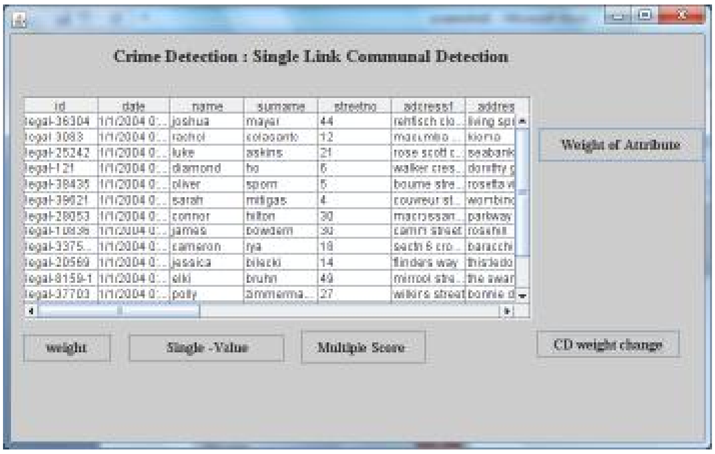


calculated

2.00000005E16

Based on the Communal Detection (CD) algorithm, update a Spike Detection (SD) and weight is updated. Weight is calculated and then multiple score. Based on the Spike Detection (SD) weight, update the

Communal Detection (CD) weight. At the end of every current data process, Spike Detection (SD) algorithm calculated and updates attribute weight for Communal Detection (CD).



Crime Detection : Single Link Communal Detection

id	date	name	surname	streetno	address1	addres
legal-36304	1/1/2004 0:...	joshua	majar	44	rehfisch clo...	living spr
legal 3083	1/1/2004 0:...	rachol	colosonto	12	macumbo ...	kiorno
legal-25242	1/1/2004 0:...	luke	askins	21	rose scott c...	seabank
legal-121	1/1/2004 0:...	diamond	ho	6	walker cres...	dorothy g
legal-38435	1/1/2004 0:...	oliver	sporn	5	bourne stre...	rosetta w
legal-39621	1/1/2004 0:...	sarah	milligas	4	couveur st...	wombinc
legal-28053	1/1/2004 0:...	connor	hilton	30	macrossan...	parkway
legal-10836	1/1/2004 0:...	james	bowdem	30	camm street	rosehill
legal-3375...	1/1/2004 0:...	cameron	rya	18	sectn 6 cro...	baracchi
legal-20569	1/1/2004 0:...	jessica	bilacki	14	flinders way	thistledo
legal-8159-1	1/1/2004 0:...	elki	bruhn	49	mirrool stre...	the swar
legal-37703	1/1/2004 0:...	polly	zimmerma...	27	wilkins street	bonnie d

VII. CONCLUSION AND FUTURE WORK

This paper focused on robust Credit crime detection. It has implemented algorithms to protect applications that occupy financial transactions. It proposed prototype application has numerous layers of defense using data mining which can be used in the real world credit applications or else for credit card fraud detection. The proposed prototype has several significant concepts such as quality of data, adaptively with multi-layered defense. The communal detection as well as spike detection concepts proposed by Phua et al. was used in the implementation of the Credit crime detection system. The application is tested with real datasets as well as synthetic datasets. The experimental results discovered that the proposed algorithms are robust and can be used in the real world credit applications. In future work we adaptive Communal Analysis Suspicion Scoring (CASS) algorithm to observe application streams to detect the changing attack patterns (new false negatives) which are in direct reply to our existing search parameters.

VIII. REFERENCES

i. Bolton, R. and Hand, D. 2001. Unsupervised Profiling Methods for Fraud Detection, Proc. of CSCC01.

ii. 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.

iii. 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.

iv. 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.

v. IDAnalytics. 2008. ID Score-Risk: Gain Greater Visibility into Individual Identity Risk. Unpublished.

vi. Goldenberg, A., Shmueli, G. and Caruana, R. 2002. Using Grocery Sales Data for the Detection of Bio-Terrorist Attacks, Statistical Medicine.

vii. Oscherwitz, T. 2005. Synthetic Identity Fraud: Unseen Identity Challenge, Bank Security News 3: p. 7.

viii. Schneier, B. 2008. Schneier on Security, Wiley, Indiana. ISBN-10: 0470395354.

ix. Romanosky, S., Sharp, R. and Acquisti, A. 2010. Data Breaches and Identity Theft: When is Mandatory Disclosure Optimal?, Proc. of WEIS10 Workshop, Harvard University.

x. Jost, A. 2004. Identity Fraud Detection and Prevention. Unpublished.

xi. 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.

xii. Wong, W. 2004. Data Mining for Early Disease Outbreak Detection, PhD thesis, Carnegie Mellon University.

xiii. 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.

xiv. Winkler, W. 2006. Overview of Record Linkage and Current Research Directions, Technical Report RR 2006-2, U.S.Census Bureau.

Authors Profile

K.Rachel Praveena has received her B.Tech degree in Computer Science and Engineering (2007-2011) from Mahaveer Institute of Science and Technology, Hyderabad. Currently she is pursuing M. Tech in Computer Science (2011-2013) from School of IT, JNTUH, Hyderabad, India.

Dr. G.Venkata Rami Reddy has received his Ph.D degree. He is presently Associate Professor in Computer Science and Engineering at School of Information Technology. He is more than 11 years of experience in Teaching, and Software Development. His areas of interests are: image Processing, Computer Networks, and Analysis of Algorithms, Data mining, Operating Systems and Web technologies.

K. Suresh Babu has done his M. Tech (Computer Science) from Central University, Hyderabad and presently pursuing Ph. D. from JNT University in the field of Network Security in MANETs. He has a teaching experience of 12years.His subjects of interests are Computer Networks, Network Security, Wireless Networks and Mobile Computing, Security in Mobile Computing. He published several papers in international journals and national journals, also participated and presented papers in International and National Conferences. He is presently course coordinator for M. Tech (Computer Science).He is also Program Officer for Nation Service Scheme(NSS) Unit at School of IT. He is also Cisco Certified Academy Instructor (CCAI).

G. Sudhakar he is presently Lecturer in Computer Science and Engineering at School of Information Technology, JNTUH, Hyderabad, India.