Unsupervised Identity Application Fraud Detection using Rule-based Decision Tree

Amany Abdelhalim Issa Traore

Department of Electrical and Computer Engineering
University of Victoria, P.O.Box 3055 STN CSC,
Victoria, B.C., V8W 3P6, Canada,
Phone: (250) 721-6036, Fax: (250) 721-6052
{amany, itraore}@ece.uvic.ca

Abstract

Identity fraud is becoming a growing concern for most government and private institutions. In the literature, identity fraud is categorized into two classes, namely application fraud and behavioral (or transactional) fraud. Most of the previous works in the area of identity fraud prevention and detection have focused primarily on credit transactional frauds. The work described in this paper is one of the very few works that focus on application fraud detection. We present an unsupervised framework to detect fraudulent applications for identity certificates by extracting identity patterns from the web, and crossing these patterns with information contained in the application forms in order to detect inconsistencies or anomalies. The outcome of this process is submitted to a decision tree classifier generated on the fly from a rule base which is derived from heuristics and expert knowledge, and updated as more information are obtained on fraudulent behavior. We evaluate the proposed framework by collecting real identity information online and generating synthetic fraud cases.

Keywords

Identity fraud, Application fraud, Fraud detection, Anomaly detection, Web mining, Rule-based Decision Tree, Data Mining.

1 INTRODUCTION.

Identity fraud is spreading fast and causing more and more damages both financially and sociologically. Identity fraud occurs when a criminal impersonates another individual by taking on that person's identity or by creating a fake identity for whatever reason [1] and [2].

Identity frauds can be categorized into two different types: transaction frauds and application frauds. Application fraud occurs when an individual or an organization applies for an identity certificate (e.g., passport, credit card etc.) using someone else's identity. Transaction fraud, also known as behavioral fraud, occurs when an identity thief performs some operations or transactions using fake or stolen identity.

Most of the research in identity fraud detection has focused so far on credit transactional fraud detection. Limited attention has been paid to application fraud detection, where only few papers have been published so far. Application fraud detection, however, is an important aspect of any sound and global strategy to combat identity fraud. Application fraud detection is a proactive measure that allows early screening of fraudsters, contributing as a result to cutting down significantly the effort and resources required to detect fraudulent transactions [3].

A wide variety of data sources may be used in isolation or in combination for application fraud detection. An alternative identity information source, which to our knowledge has not been deeply explored for application fraud detection is the web. The web is actually a federation of many different identity information data sources. Using the web, it may be possible to cross-check application forms with data from various collateral identity information sources; even though such information might be sparse or useful information could be missing.

In this work, we propose an application fraud detection framework that consists of two main components: an online identity mining module and a fraud detector. Our proposed online identity mining scheme is based on *white hat Google hacking*, in which identity information is collected through online search, targeting a specific individual (i.e. the applicant). The fraud detector is based on an intelligent unsupervised decision model that analyzes online identity information related to

the applicant and crosses such information with information contained in the application form, in order to detect and report possible inconsistencies or anomalies. Although various machine learning techniques (supervised or unsupervised) may be used in designing this kind of detector, the lack of genuine fraud data tends to hinder such process. A common challenge of data-mining based fraud detection research is the lack of publicly available real data for model building and evaluation. For privacy and competitive reasons, organizations are reluctant to release data related to fraudulent activities.

In order to compensate for the fact that labels may not be readily available, ideally such techniques should be unsupervised. Although many unsupervised detection frameworks have so far been proposed for transaction fraud detection, none of them apply specifically to application fraud detection. To our knowledge, the application fraud detection techniques proposed so far in the research literature are either supervised or semi-supervised.

The solution commonly adopted in the industry and in the literature to address this issue is to encode expert knowledge and past knowledge of fraudulent behavior into rule bases. However, due to the fast pace at which new fraud methods are created and used by fraudsters, the rule bases are submitted to constant changes and usually tend to grow at an accelerated rhythm, quickly reaching unmanageable size. This might not be conducive to timely decision, which is required in many business environments. In this case, a decision tree represents an effective alternative to rule-base reasoning for a quicker decision. This is because in order to be able to make a decision for some situations we need to decide the shortest and most efficient order in which tests should be evaluated. In those cases a decision structure (e.g. decision tree) is much quicker than a rule engine to reach a decision. Furthermore, decision trees are more readily understood by others in the organization than a set of rules. Consequently, they are more appropriate as a communication tool.

We use RBDT-1, a rule-based decision tree technique that we recently proposed in [5] to design our fraud detector. RBDT-1 is capable of generating a decision tree from a set of rules and has been shown to be more effective than other existing similar techniques [5].

The rest of the paper is structured as follows. In Section 2, we summarize and discuss related work. In Section 3, we present our approach and give an overview of the general architecture of the tool. In Section 4, we present our online identity information retrieval scheme. In Section 5, we present our identity fraud detection scheme. In Section 6, we conduct the evaluation of the proposed framework. Finally in Section 7, we make some concluding remarks.

2 RELATED WORK.

Although there is a large amount of published works on identity fraud detection, only a few of these works focus specifically on application fraud detection. Cross-referencing new applications with similar information from other databases is a common characteristic of most of the application fraud detection approaches proposed so far in the literature, including those in [6], [7] and [8].

Wheeler et al. in [6] applied case-based reasoning for credit card application fraud detection. The proposed system was used as reinforcement for an existing rule-based (RB) fraud detector. The input data to the system consists of pairs of database records, consisting in one hand of the application and on the other hand of fraud evidence produced by the RB system. The proposed case based reasoning framework consists of two decision-making modules, in charge of case retrieval and diagnosis, respectively. The retrieval component utilizes a weighting matrix and nearest neighbor matching to identify and extract appropriate cases to be used in the final diagnosis for fraud. The decision component utilizes a set of algorithms (i.e. probabilistic curve selection, best match algorithm, negative selection algorithm, density selection algorithm, and default goal) to analyze the retrieved cases and attempt to reach a final diagnosis. The results of the system showed that the performance of each algorithm employed in the fraud diagnosis process differs depending on the nature of the fraud case presented. In our work we also use application cross-referencing to capture similarities. However, our fraud detection algorithm does not need labels to make fraud or non-fraud decisions. While in the work of Wheeler and Aitken, the evidence has to be produced and tagged as fraud or non-fraud by a separate system, our proposed system uses unlabelled data in its decision-making process.

Phua *et al.* in [7] proposed a technique for detecting application fraud based on implicit links between new and previous applications. The proposed technique is similar to the weight matrix proposed in [6], except that in [7], in addition to basing their calculation of suspicion scores on matching attributes, they take into consideration temporal and spatial differences between the matching applications. The approach will be described in detail in section 6.3. A key difference between this approach and ours is that it requires labeling the data. Despite this important difference,

we evaluate our work by mainly comparing it to the approach proposed by Phua *et al.*, as explained later.

In [8], a system that detects subscription fraud in fixed telecommunications is proposed. The system consists of two main modules, a classification module and a prediction module. The purpose of the prediction module is to detect fraudulent customers when they attempt to subscribe to a fixed telecommunication service. To investigate the application for signs of fraud, the prediction module crosses the information available in the new application with information available in other sources. The prediction module consists of a multi-layer feed-forward neural network. The output units indicate one of two decisions for an application; either fraudulent or legitimate. Experimental validation of the proposed approach based on a labeled dataset of subscription examples showed that the prediction module was able to identify only 56.2% of the true fraudster cases while screening 3.5% of all the subscribers in the testing set. Like in [6] and [7], the proposed framework requires using labeled data and some form of supervision. Many criticisms can be made about using labeled data for fraud detection, including the limited efficiency of processing such data in event-driven environment, the cost, difficulty, and length of time needed to obtain such data, and the fact that the class labels can be inaccurate. In our work, we do not use or assume knowledge of existing fraudulent applications in the screening process. Instead, our proposed fraud detector processes unlabelled data from the identity information source.

3 GENERAL APPROACH.

Identity can be defined "as one or more pieces of information that cause others to believe they know who someone is" [9]. In [2], the notion of *identity certificate* is introduced.

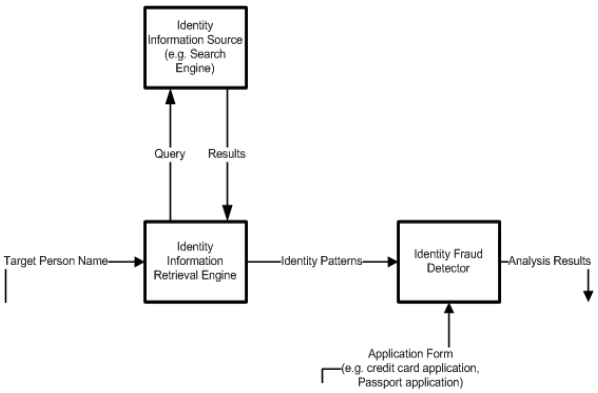

Figure 1. Fraud Detection Framework.

An *identity owner* applies for an identity certificate for various purposes in his life, administrative or business related. *Identity issuers*, represented by trusted government or private institutions, deliver identity certificates.

To obtain an identity certificate an individual submit an *application* to a certificate issuer providing required identification information. In doing so, he makes an *identity claim*. So, an *identity claim* occurs when an individual declares a specific identity, for instance, on an official document like a passport application, or a business document like a credit card application.

Our proposed identity fraud detection approach consists of screening a particular *identity claim* for possible inconsistencies by crossing corresponding information with related identity information patterns collected from selected identity information sources. Our application fraud detection framework consists of three main modules as illustrated by Figure 1: the identity information source, the identity information retrieval engine, and a fraud detector.

4 ONLINE IDENTITY INFORMATION RETRIEVAL.

Because of the huge amount of information available online, locating and sorting identity information related to a specific individual is a daunting task.

A key challenge in online targeted search is that a wide variety of document formats are used on the Internet (e.g., PDF, Doc, Excel, or Html). Some of these formats are not directly searchable, and require some form of pre-processing. Also our search space is not limited to a specific type of documents Furthermore it is difficult to establish the existence of values corresponding to the identity keywords located in the documents. The reason for such difficulty is that although some of the identity keywords targeted in our search have a fixed

number of characters, such as social security number or telephone number, no standard format is used to express corresponding values in the documents on the Internet.

Details about our algorithm describing the proposed targeted search strategy can be found in [10]. An important challenge that is addressed by our targeted search scheme is the fact that some of our identity keywords have homonyms. In many cases, sorting information pertaining to different homonyms is challenging. In the literature, one of the main approaches used to handle homonyms consists of associating semantic information with the keyword. The semantic-based approach, however, does not make sense when the homonyms are proper nouns. Since we are dealing, in our work, with targeted search, homonyms based on proper nouns are very important. We handle this kind of homonyms using our identity profile derivation strategy, outlined in the next section.

5 IDENTITY FRAUD DETECTOR.

We illustrate in this section our proposed identity fraud detection strategy and mechanisms.

Our identity fraud detection approach consists of two major phases: the derivation of individual profiles and the analysis of the derived profiles using rule-based decision tree. We illustrate each of these two phases in the following.

5.1 Shared Identity Information. Let P be the set of identity patterns returned by a targeted search. Each identity pattern $pi \in P$ is represented by a k-dimensional attribute vector $<p_{i1},..., p_{ik}>$. Possible examples of attributes include the following:

<Social security number, Date of birth, Address, Telephone number, Mother maiden name, Credit card number, Credit card expiry date, Credit card type, Credit card security number>

We exclude the name simply because at this stage we are dealing with identity information belonging to homonyms. A typical identity pattern returned from a targeted search might include only a subset of the k attributes set. The missing (or undefined) fields will simply be considered non-applicable (NA), and will not be used for the matching. Figure 2 illustrates sample identity patterns. In this example the following five identity attributes are considered: social security number (ssn), date of birth (dob), mother maiden name (mmn), address (addr) and phone number (tel). The retrieval engine uses a name and a set of keywords to search the identity information source (e.g., the Web), and returns a collection of identity information related to the identity claim corresponding to the application being checked. As a result the returned identity information may correspond to several different individuals, all having the same name. For instance, we can have in the search result several social security numbers corresponding to the same name, which necessarily means that different individuals actually share the searched name.

We identify the patterns related to the applicant by determining direct and indirect connections between the application pattern and the patterns retrieved from the identity information source.

Two patterns are directly connected if they share at least one attribute value. Two patterns p and q have an indirect connection, if there is a sequence of directly connected patterns linking them. In other words, there exists a sequence of patterns $p1 \ldots pn$, such that (p, p_1), (p_n, q), and (p_i, p_{i+1}), $\forall i \in \{1, \ldots, n-1\}$, are each a direct connection.

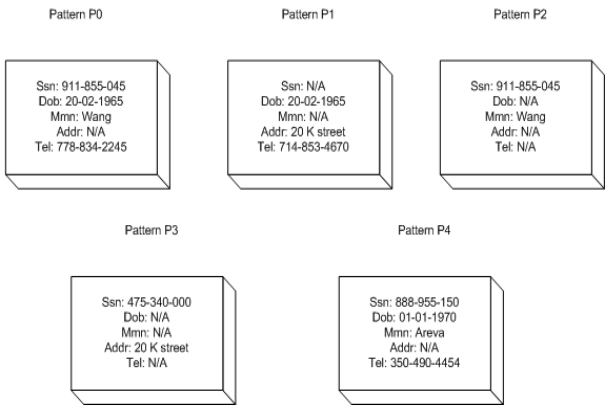

Figure 2. Examples of identity patterns based on 5 attributes.

For the purpose of fraud detection, we trim the returned set P of identity patterns, and retain only the patterns having direct or indirect connections with the application pattern; let P^+ denote the subset of P containing all such patterns. Let p_0 denote the application pattern and let G denote a set of patterns such that $G = P^+ \cup \{p_0\}$.

As an example, let's consider the sample patterns depicted by Figure 3.

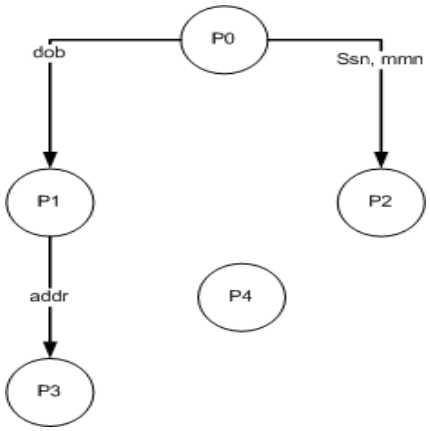

Figure 3. Direct and indirect connections between identity patterns.

Figure 3 depicts a tree structure exposing the direct and indirect connections between the sample patterns shown in Figure 2. We assume in this example that p_0 is the application pattern and the set of retrieved patterns is *P, P= {p1, p2, p3, p4}*. The set of connected patterns is *G, G= {p0, p1, p2, p3}*. For instance, patterns p_0 and p_2 share two attributes, namely *ssn* and *mmn*. Pattern p_4 has no connection (direct or indirect) with p_0, so it is excluded from the fraud analysis. It is assumed in this case that p_4 does not belong to the applicant but belongs to another person that shares his name.

For the purpose of fraud detection, we analyze the link between every pair of patterns from *G* which are directly connected. Specifically we convert every pair of patterns that are directly connected into a single feature vector characterizing the underlying relationship. For each pattern-pair $< p_i, p_j > \in G \times G$, we derive a feature vector $v_{ij} = [\delta_{ijl}]_{1 \le l \le k}$, such that

$$\delta_{ijl} = \begin{cases} ? \;\; if \;\; ((p_{il} = na) \;\; or \;\; (p_{il} = na)) \\ 1 \;\; if \;\; p_{il} = p_{jl} \\ 0 \;\; otherwise \end{cases} \quad (5.1)$$

Where "?" denote a missing value, which may be either 1 or 0, but this is unknown.

5.2 Fraud Detection. We present, in this section, our fraud detection approach, followed by an overview of the RBDT-1 method, and the initial rule base.

5.2.1 Approach. Our fraud detection approach consists of analyzing the coherence or consistency of the shared identity information between patterns. For instance, two identity certificates attributed to the same individual are expected to bear the same birth date and mother maiden name, when such information is available. The feature vectors describing patterns connections are used as basis for such analysis. As an outcome of the analysis, patterns connections are classified in one of four categories: *normal, suspicious-low, suspicious-high*, or *fraudulent* represented by the labels *N, S-, S+*, and *F*, respectively.

Our fraud detector is implemented as a decision tree that takes as inputs pattern-pair feature vectors, and provides as outputs fraud decisions. The tree is built and updated dynamically based on a rule base which encodes expert knowledge or common sense understanding of the notion of fraudulent or inconsistent behavior. No previous fraud instances should be required. The size of the decision tree resulting from such process is expected to be manageable because typically application forms involve sparse and limited identity attributes.

We use a new rule-based decision tree method named *RBDT-1* that we presented in [5] to derive the decision tree. In the rest of this subsection, we give an overview of the *RBDT-1* method, and then summarize and discuss a sample of the rules.

5.2.2 Rule Base. We considered ourselves an expert in this domain and based on our readings and research in this area we were capable of converting what we read from different sources into a set of rules that describe the normal, suspicions, fraudulent cases. Based on the five attributes patterns considered in our previous example, we derived 82 rules for the initial implementation of our proof concept.

While some of the rules are straightforward, many require some explanations. A straightforward fraud case is when the social security numbers match and either the dates of birth or the mother maiden names do not. Some other straightforward cases of fraud are when the home telephone numbers match while the addresses do not.

In both cases one could assume that the same individual is impersonating two different individuals: in the first case by changing either the mother maiden name or the date of birth, and in the second case by using different locations. Two straightforward normal cases are when either none of the attributes match or all the attributes match. We assume in the case when none of the attributes matches that the non-matching patterns belong to different individuals and have not been used by the applicant (in some past attempt to defraud the system). Obviously, the case when all the attributes match is regular. A variant of this case is when the telephone numbers do not match; this is considered

normal because it could simply be the case that the same individual owns several telephone lines.
According to the degree of rarity the case is classified as either highly suspicious or lightly suspicious. For instance, considering that we are dealing with homonyms, a case where the social security numbers do not match, while the addresses, dates of birth, and mother maiden names match is extremely unusual. Because this simply means that there are two different individuals (different ssn), who are homonyms, born the same day, from homonym mothers and living at the same location. This is highly suspicious and requires further verification.

If these two individuals (born on the same day with the same mother maiden name) were living in different locations, with different phone numbers, then the situation could be considered as a coincidence, although still uncommon. We classify such occurrence as lightly suspicious. Further investigation would also be required in this case as well, but such investigation does not have to be as thorough as in a highly suspicious case. The decision tree obtained, by applying the *RBDT-1* method to the rule base, is illustrated in Figure 4. RBDT-1 is capable of transforming a set of rules into a less complex set without reducing the accuracy. Thus, the resulting tree consists of 57 rules.

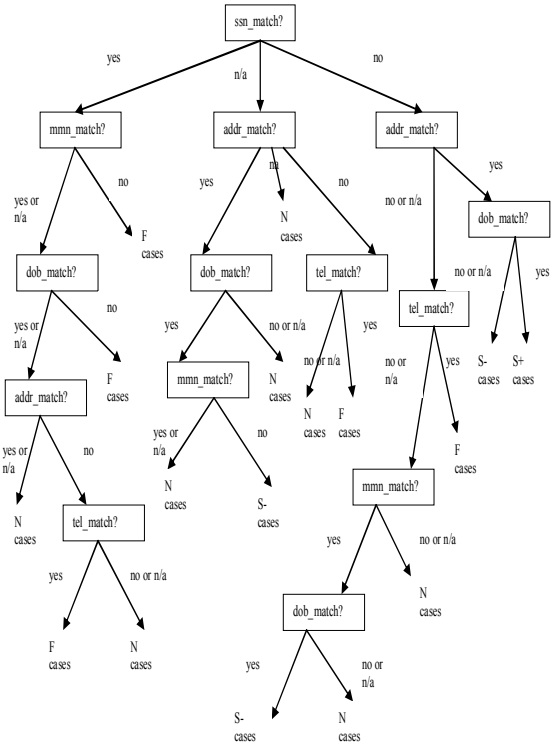

Figure 4. The decision tree produced by RBDT-1.

6 EVALUATION.

We present, in this section, the evaluation of the different components of our fraud detection framework. We start by presenting the evaluation method and data collection, and then we present and discuss the evaluation results obtained for the fraud detector.

6.1 Evaluation Method and Data. The most common way of evaluating frameworks like the one proposed in this paper consists of finding and using some public dataset containing real identity fraud information. As expected, however, the evaluation of our framework faces difficulty in obtaining real data to effectively analyze and test our system. Usually data containing real identity (fraud) information is restricted from being used due to privacy issues. An alternative approach may consist of generating synthetic fraud data. In this work, we collected real identity information online and then generated and injected synthetic fraud information - that was learned from related works [7] - in the data. Having, however, a dataset is not enough to carry out the evaluation. We need an unbiased mechanism to label the data. In the absence of a domain expert to label the data, we have decided to adopt a comparative labeling mechanism in this work. More specifically, we used the communal scoring approach by Phua *et al.* [7] described in the related work to produce an initial set of labels, and then compared the produced labels to the labels obtained with our proposed approach. Using such comparative evaluation allows us to assess the relative strength of our approach.

To obtain real identity information, we conducted some white hat Google hacking experiments over two weeks. The study allowed us to establish a database of online identity information related to 154 different individuals, which serves as basis for the validation of our identity fraud detection framework. Details of the study and corresponding results can be found in [10].

6.2 Application Fraud Data. In order to evaluate our application fraud detector, we need instances of labeled data both with normal, fraud and suspicious cases and then label the data with our proposed fraud detector and compare the results.

As mentioned above, due to privacy and confidentiality reasons, obtaining real identity fraud information is extremely difficult. Usually the available real data is encrypted and key identity attributes are removed which makes it ineffective for evaluating our detector.

To overcome this issue we assume that the data that we collected in our white hat Google hacking experiments are application forms submitted by applicants applying for an identity certificate, which yields 154 normal applications. In addition we used the names of each of the 154 applicants to search for identity patterns containing identity information that share the same name.

Since the data that we collected does not include fraud instances, we created some synthetic fraud data based on obvious fraud concepts. Specifically, the main obvious fraud concept used in our evaluation was based on the case where pair of applications bear the same social security number and have different dates of birth or mother maiden names. Based on these fraud concepts, we created synthetic data based on all the possible value combinations which corresponds to 45 fraud data instances. So, overall our evaluation dataset involved 199 data instances consisting of 154 normal cases and 45 fraud cases.

6.3 Data Labeling. We used a methodology based on a technique proposed by Phua *et al* in [10] for labeling the data instances that were described in the previous section as normal, fraud or suspicious.

Phua *et al.* technique is referred to (here) as CASS for communal analysis suspicion scoring. CASS is a technique for generating numeric suspicion scores for credit applications based on implicit links to other previous applications over both time and space.

Every new application v_i is pair-wise matched against previous scored applications within a window W.

With their technique, Phua *et al.* label a new application v_i by first linking it with an existing application v_j from either a black list or a white list or as anomalous, or by considering it unlinked. Linkage between v_i and v_j is denoted as $v_i \xrightarrow{L} v_j$, where L is a label, corresponding to *fraud*, or *normal* or *anomalous*. After establishing the linkage, the communal suspicion score $W_{communal_{ij}}$ for the linked application-pair is computed accordingly.

Phua *et al.* assume that the black list is made up of actual identity applications previously found to be fraudulent. As a result, any new application that contains similar identity information to those applications in the black list is considered as a fraud.

In CASS, the connectivity of a new application to the black list, white list, anomalous list or unlinked list is based on thresholds; T_{fraud} for the black list and T_{normal} and a set of relationships defining normal connections for the white list.

In the process of labeling our data using Phua *et al.*'s technique, we populated the black list with identity information of four synthetic fraud applications crafted based on the obvious fraud concept explained in the previous section. These will be used to match against our 45 synthetic fraud applications mentioned in the previous section. We populated the white list with 15 normal relationships from [16] after adapting them to fit the five attributes used in our fraud detector. We randomly selected 49 patterns from the identity patterns that we collected online and labeled them as normal to link them to the white list.

We assumed that the suspicion scores of the applications linked to the black list were very high, and those linked to the white list were very low. Thus, in this case we labeled a new application linked to an application in the black list with $W_{communal} = 1$ as a fraud application. We labeled the applications linked to the white list with $W_{communal} > 0.5$ and the unlinked applications as normal applications and labeled those linked as anomalous with $W_{communal} > 0$ as suspicious.

6.4 Results. To evaluate our application fraud detector we removed the labels of the data and allowed our system to produce its own labels, and then compared them to the data labels produced by CASS. The outcome of the comparison is the *Match Rate (MR)*, when the labels produced by both techniques coincide. The *Non Match Rate (NMR)*, which is the complement of the MR (NMR=1-MR) measures the disagreement between both techniques. Note that a high NMR doesn't necessarily mean a weakness of either of the techniques. A closer error analysis needs to be done to find out which technique is actually at fault.

We chose the values 2 and 3 for the normal and fraud thresholds, which are considered intermediate values since the application-pair matching score is based on five attributes. As a result of setting the thresholds for fraud and normal to a combination of the above two values (i.e. 2 and 3) we produced four different sets of labels for our data. We compared the labels produced by our proposed fraud detector to each of the four sets of labels and computed the match rate by our method

Based on the selected thresholds, the best match rate produced by our proposed fraud detector was 92% for the data labeled by CASS using a combination of $T_{normal} = 3$ for the white list and the anomalous and $T_{fraud} = 2$ for the black list.

To analyze this result, we have to discuss the labels produced by the CASS approach. By reviewing the 45 synthetic fraud applications that we created, we discovered that 12 fraud applications were labeled as unlinked, and as a result were considered as normal and got accepted. These 12 applications are cases where two applications share only one attribute value (below the black list threshold) that happens to be the social security number. Although these applications share only one attribute, they correspond to an obvious fraud case because the social security number is a unique attribute and hence requires that the rest of the attributes to be identical. Thus, in that sense our application fraud detector produces a correct fraud label for all the 45 fraud applications which, however, does not match the CASS labels. CASS produces the wrong labels because, as explained above, the method depends on a threshold which is not effective in detecting such fraud cases.

For the 154 normal applications, while our application fraud detector labels them as normal, CASS labeled 3 of the 154 applications as anomalous and the rest as normal. Examination of these 3 CASS-labeled anomalous applications shows that while the threshold for a linkage between each of the applications and a normal application is met there is no corresponding matching relationship in R. This could possibly be addressed by expanding the initial set of normal relationships used for the evaluation.

Overall our proposed system performs well when assessed against data labeled using CASS, which underscores its potential as a strong fraud detector.

7 Conclusion.

We have presented in this paper an unsupervised application fraud detection framework that analyzes online identity patterns using an intelligent decision model to identify fraudulent applications. We defined heuristic rules for detecting fraudulent applications and used a rule-based decision tree method for transforming the rules into a decision tree. To evaluate our fraud detector we used a mix of real data collected online and synthetic data to induce some fraud cases. The data was treated as identity applications and labeled using a technique called CASS as normal, fraud or suspicious. The data was labeled four times based on the CASS approach by using different combinations of normal and fraud thresholds and compared to the labels produced by our fraud detector. The best match rate produced by our detector was 92%. The non-matching cases happened to be related to labeling errors by the CASS approach. Overall our approach shows strong promise for online identity application fraud detection.

8. REFERENCES

[1] Crown (2002) 'Identity fraud: A study', Economic and Domestic Secretariat Cabinet Office, www.homeoffice.gov.uk/docs/id_fraud-report.PDF.

[2] WenJie Wang; Yufei Yuan; Archer, N,"A Contextual Framework for Combating Identity Theft", Security & Privacy Magazine, IEEE Volume 4, March-April 2006 pp: 30 – 38.

[3] Bolton, R. J. and Hand, D. J. (2001) 'Unsupervised profiling methods for fraud detection', *Credit Scoring and Credit Control, Vol. 2, pp.*5–7.

[4] I. F. Imam, and R. S. Michalski "Should decision trees be learned from examples of from decision rules?", Source Lecture Notes in Computer Science. In Proceedings of the 7th International Symposium on Methodologies, 689, 1993, pp. 395–404.

[5] Abdelhalim, A.,Traore, I. and Sayed, B. 'RBDT-1: a New Rule-based Decision Tree Generation Technique', *3rd International Symposium on Rules, Applications and Interoperability,* Las Vegas, Nevada, USA: Nov 5-7, 2009.

[6] Wheeler, R. and Aitken, S. (2000) 'Multiple algorithms for fraud detection', *Knowledge-Based Systems, Vol. 13, No. 3*, pp.93–99.

[7] Phua, C., Gayler, R., Lee, V. and Smith-Miles, K. (2009). On the Communal Analysis Suspicion Scoring for Identity Crime in Streaming Credit Applications. European Journal of Operational Research, 195, pp. 595-612.

[8] Estevez, P. A., Held, C. M. and Perez, C. A. (2006) 'Subscription fraud prevention in telecommunications using fuzzy rules and neural networks' *Expert Systems with Applications, Vol. 31, No. 2*. (in press).

[9] Chan, P. K., Fan, W., Prodromidis, A. L. and Stolfo, S. J. (1999) 'Distributed data mining in credit card fraud detection', *Intelligent Systems, IEEE, Vol. 14, No. 6, pp.*67–74.

[10] Abdelhalim, A. and Traore, I. (2007) 'The impact of Google hacking on identity and application fraud', *Proc. IEEE Pacific Rim Conference, Communications, Computers and Signal Processing*, 240–244.