COMPUTERS & SECURITY XXX (2015) 1–12

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

BankSealer: A decision support system for online banking fraud analysis and investigation

Michele Carminati [a,*], *Roberto Caron* [a], *Federico Maggi* [a], *Ilenia Epifani* [b], *Stefano Zanero* [a]

[a] *Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Italy*
[b] *Politecnico di Milano, Dipartimento di Matematica, Italy*

ARTICLE INFO

Article history:
Received 20 December 2014
Received in revised form
18 March 2015
Accepted 2 April 2015
Available online xxx

Keywords:
Internet banking
Fraud detection
User profiling
Decision support system

ABSTRACT

The significant growth of online banking frauds, fueled by the underground economy of malware, raised the need for effective fraud analysis systems. Unfortunately, almost all of the existing approaches adopt black box models and mechanisms that do not give any justifications to analysts. Also, the development of such methods is stifled by limited Internet banking data availability for the scientific community. In this paper we describe BankSealer, a decision support system for online banking fraud analysis and investigation. During a training phase, BankSealer builds easy-to-understand models for each customer's spending habits, based on past transactions. First, it quantifies the anomaly of each transaction with respect to the customer historical profile. Second, it finds global clusters of customers with similar spending habits. Third, it uses a temporal threshold system that measures the anomaly of the current spending pattern of each customer, with respect to his or her past spending behavior. With this threefold profiling approach, it mitigates the under-training due to the lack of historical data for building well-trained profiles, and the evolution of users' spending habits over time. At runtime, BankSealer supports analysts by ranking new transactions that deviate from the learned profiles, with an output that has an easily understandable, immediate statistical meaning.

Our evaluation on real data, based on fraud scenarios built in collaboration with domain experts that replicate typical, real-world attacks (e.g., credential stealing, banking trojan activity, and frauds repeated over time), shows that our approach correctly ranks complex frauds. In particular, we measure the effectiveness, the computational resource requirements and the capabilities of BankSealer to mitigate the problem of users that performed a low number of transactions. Our system ranks frauds and anomalies with up to 98% detection rate and with a maximum daily computation time of 4 min. Given the good results, a leading Italian bank deployed a version of BankSealer in their environment to analyze frauds.

* *Corresponding author.*
E-mail addresses: michele.carminati@polimi.it (M. Carminati), roberto.caron@mail.polimi.it (R. Caron), federico.maggi@polimi.it (F. Maggi), ilenia.epifani@polimi.it (I. Epifani), stefano.zanero@polimi.it (S. Zanero).
http://dx.doi.org/10.1016/j.cose.2015.04.002
0167-4048/

1. Introduction

The popularity of Internet banking has led to an increase of frauds, perpetrated through cyber attacks, phishing scams and malware campaigns, resulting in substantial financial losses (Wei et al., 2013; Bolton and David). In 2013, Kaspersky Lab[1] detected 28.4 million attacks using financial malware, with a 27.6% increase over 2012. The number of users targeted in attacks involving financial malware also rose by 18.6% to 3.8 million. A similar trend characterizes online banking frauds which increased 30% in 2012–2013.[2]

Internet banking frauds are difficult to analyze and detect because the fraudulent behavior is dynamic, spread across different customer profiles, and dispersed in large and highly imbalanced datasets (e.g., web logs, transaction logs, spending profiles). Despite the importance of the problem, the development of new online banking fraud decision support systems is made difficult by the limited availability of transactions and fraud datasets, due to privacy concerns. As a consequence, only a limited amount of research deals with fraud detection in online banking. Commercial systems do exist, but they offer limited insight in their inner workings due to obvious intellectual property concerns. We noticed that most existing approaches build black box models that are not very insightful for analysts in the subsequent manual investigations, making the process less efficient. In addition, systems based on baseline profiling are not adaptive, and do not take into account cultural and behavioral differences that vary from country to country. Instead of focusing on *pure detection* approaches, we believe that more research efforts are needed toward systems that *support* investigations. Cooperating with a leading security company which helps banks build fraud detection systems and processes, we had the unique opportunity to work on a real-world, anonymized dataset of Internet banking transactions.

In this paper we present a detailed description of BankSealer (Carminati et al., 2014), a decision support system for online banking fraud analysis and investigation that automatically ranks frauds and anomalies in transactions. Most of the development was driven by the analysis of the dataset itself. BankSealer uses a combination of advanced data mining, statistical, and mathematical techniques to automatically rank transactions on the basis of the risk of being fraudulent. During a training phase, it builds a local, global, and temporal profile for each user. The local profile models past user behavior to evaluate the anomaly of new transactions by means of a novel algorithm that uses the Histogram Based Outlier Score (HBOS). The global profiling clusters users according to their transactions features via an iterative version of Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and compute the anomaly with the Cluster-Based Local Outlier Factor (CBLOF). The temporal profile aims to model transactions in terms of time-dependent attributes. For this, we design a series of thresholds and measure the anomaly in terms of the percentage gap from the thresholds once they are exceeded. We handle the concept drift of the scores with an exponential decay function that assigns lower weights to older profiles.

We tested the BankSealer on real-world data, injecting a realistic set of attacks (e.g., credential stealing, banking trojan activity, and frauds repeated over time) built in collaboration with domain experts. Our system ranked fraud and anomalies with up to 98% detection rate.

In summary, our main contributions are:

1. An in-depth analysis of a real-world online banking dataset, in which we highlight the aforementioned challenges and the importance of dealing with dataset scarcity in this research field.
2. A general framework for online semi-supervised outlier-detection based on a combination of different models to discover different types of frauds. Our approach has a score with a clear statistical meaning, is adaptive to non-stationary sources and can deal with concept drift and data scarcity.
3. An almost exhaustive evaluation through a set of realistic attacks and in a real-world setting, thanks to the deployment to a large national bank.

2. Online banking fraud detection: goals and challenges

Our goal is to support the analysis of (novel) frauds and anomalies. Hence, we do not want to focus on a classifier but provide the analysts with a ranked list of transactions, along with the risk score. The rationale behind this design decision is that analysts must investigate reported alerts in any case: therefore, the focus is on collecting and correctly ranking evidence that support the analysis of fraudulent behavior, rather than just flagging transactions.

From a literature review (described in Section 6) and a real-world dataset obtained from a large national bank (described in Section 3), we found peculiar characteristics that make the analysis of this data particularly challenging. First and foremost, the distribution of attributes values is imbalanced and skewed (non-symmetric), which makes it difficult to approximate with most common statistical distributions, and unusable with most statistical methods to explain or predict trends and outliers. A second troublesome characteristic is the prevalence of users who perform a low number of transactions – an issue not considered in previous literature. Finally, the system must adopt a simple design and must be able to handle the high load of transactions avoiding high computational and spatial complexity.

Given the scarcity of labeled datasets, such a system must be able to work in an unsupervised or semi-supervised fashion (we can assume that no fraud exists in this dataset, as indicated by our collaborators). This conflicts with the requirement of the system being able to provide "readable" evidence to corroborate each alert. These peculiarities have remarkable implications for the typical statistical and data mining methods used in the outlier detection field.

[1] Kaspersky Lab – Financial cyber threats in 2013 – Available at http://goo.gl/8iaDcU.

[2] Symantec – Internet security threat Report 2013 – Available at http://goo.gl/hDgafz.

Please cite this article in press as: Carminati M, et al., BankSealer: A decision support system for online banking fraud analysis and investigation, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.002

3. Dataset analysis

Our system design is guided by an in-depth analysis of a real-world dataset, that is paramount for our work and provides useful insights for future research.

3.1. Dataset description

We obtained a dataset of transactions from a large national bank, collected between December 2012 and August 2013. The dataset was anonymized by removing personally identifiable information, and substituting it with randomly-generated unique values to ensure our analysis could still link values that happened to be equal.

The data contains customer transactions related to **Bank transfers** (i.e., money transfers from any account of the bank to any other account), **Prepaid cards** (i.e., transactions to top up credit on prepaid cards) **Phone Recharges** (i.e., transaction to refill prepaid cellphone accounts).

Table 1 summarizes the number of transactions and customers involved.

The selection of relevant features is a particularly important step. Beyond the obvious ones (such as **Amount**, **IP** address of the customer, and **Timestamp** of the transaction), we selected the following attributes, based on a preliminary analysis of data:

- **CC_ASN**: the country from which the customer makes their connection, based on the Autonomous System.
- **UserID**: unique ID associated to a user.
- **IBAN**, **IBAN_CC**: the identifier of the beneficiary account, and country.
- **Card type** (i.e., the circuit), and **number** of the prepaid card.
- **Phone operator**, and **number** of the beneficiary of the top-up.

3.2. Attribute distribution

To measure the quality of the dataset and of attributes, we make an exploratory analysis on their values. We show the results on the bank transfer data for brevity, but similar results are obtained for the other contexts.

Fig. 1 shows the distribution of the transaction amounts. We can see that the majority of transactions has low amounts, and that there are peaks at "round" amounts. Initially, we decide to discretize numerical attributes using a standard equi-frequency binning technique. However, the last bin covers a very large range of values, due to the distribution being long-tailed. This does not allow us to discriminate between spending pattern. Thus, we decide to "break down" the last bin by re-applying the same technique, producing the binning shown in Fig. 2. This static binning is necessary due to the fact that our approach allows the updating of the model for handling the concept drift.

If we observe the distribution of the transactions over the hours of each day, the majority of the operations are executed during working hours. We apply a discretization to transform such timestamps in a categorical value, by splitting the day in early morning, morning, afternoon, evening, and night, as shown in Fig. 3.

In general, we observe that common attributes to the online banking services under analysis (i.e., Amount, IP, Timestamp), show a strongly skewed and imbalanced distribution. In addition, the majority of categorical attributes have an irregular and noisy trend with a high cardinality associated to a few values.

We notice a striking dissimilarity between bank-wise vs. user-wise attribute distributions. When analyze globally (i.e., bank-wise), certain attributes exhibit uniform distributions; when analyzed locally (i.e., user-wise), the very same attributes show an imbalanced, skewed distribution, often with more than one modality. This motivates our approach in building user based profile (see Section 4).

A challenging aspect is the abundance of users who perform few transactions, insufficient to build user profiles in a reasonable time frame. Unfortunately, none of the previous works in the area addresses this problem.

Table 1 – Number of transactions, customers and attributes for each type of transaction. Attributes in bold are hashed for anonymity needs.

Dataset	Attributes	Users	Transactions
Bank transfers	Amount, CC_ASN, **IP**, **IBAN**, IBAN_CC, Timestamp	92,653	718,927
Phone recharges	Amount, CC_ASN, **IP**, Phone operator, **Phone number**, Timestamp	29,298	100,688
Prepaid cards	Amount, Card type, **Card number**, CC_ASN, **IP**, Timestamp	16,814	71,362

3.3. Correlation and dependence analysis

We determine to what extent features are directly correlated or dependent on each other. Attributes that share the same information (e.g., Phone operator and Phone number) and attributes derived from computations (e.g., ASN from IP) are obviously correlated. Apart from these, computing correlation on non-homogeneous values requires us to use approximated methods (Myers and Well, 2003). In particular, we use the *point biserial* r_{pb} *methods* to study the correlation between quantitative attributes (e.g., "Amount") and the categorical ones (CC_ASN, IBAN_CC, etc). For the correlation between categorical attributes, we use the *Kendall-tau rank correlation coefficient* and the *Spearman's rank correlation coefficient*, by sorting the analyzed attributes according to the frequency of each value in the dataset.

We obtain values near to zero for all coefficients and, hence, the features under analysis can be considered not to be directly correlated.

To study dependence, we evaluate non-parametric tests (for a general overview see Conover, 1999). Pearson's χ^2 test for independence, for instance, has a pre-requisite of the contingency matrix having at least 80% of cells with more than five observations. However, in our case the contingency matrix has more than 50% of cells with no observations, due to the very high degree of freedom of the cardinality of our attributes. For the same reason, it is impossible to apply Yate's

Please cite this article in press as: Carminati M, et al., BankSealer: A decision support system for online banking fraud analysis and investigation, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.002

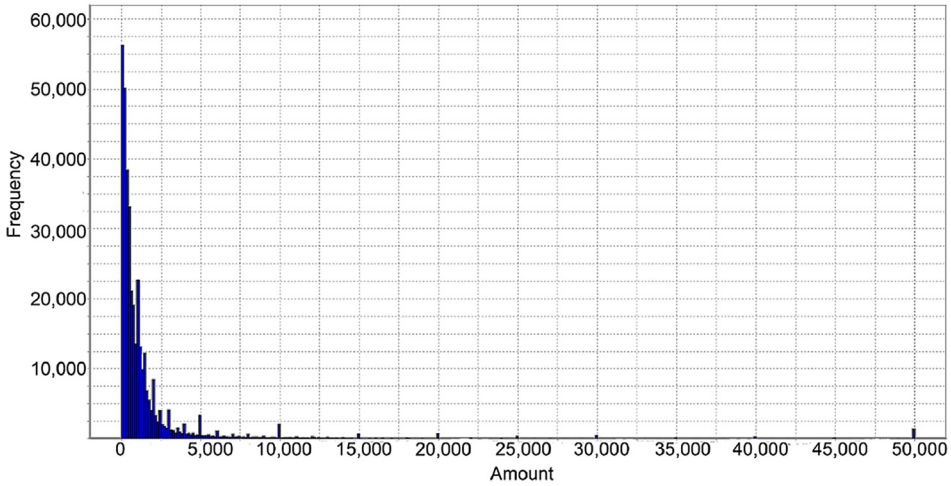

Fig. 1 – Distribution of the transaction amounts.

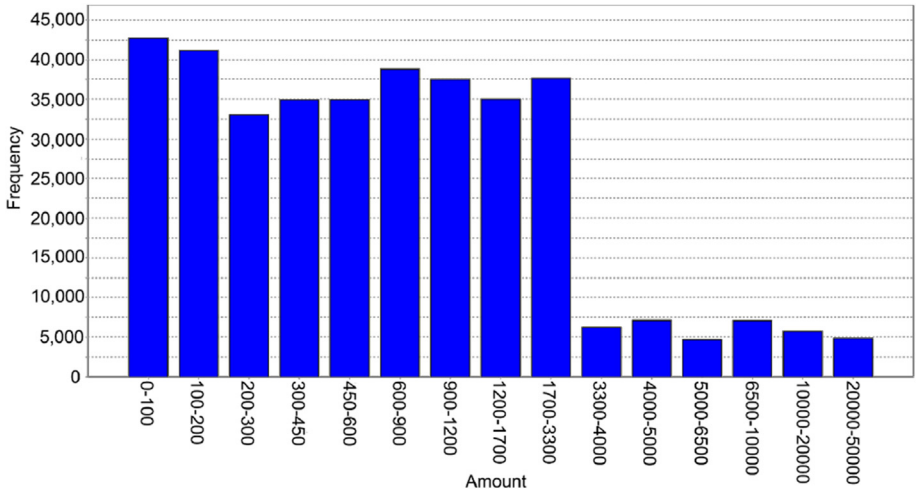

Fig. 2 – Discretization of amount distribution.

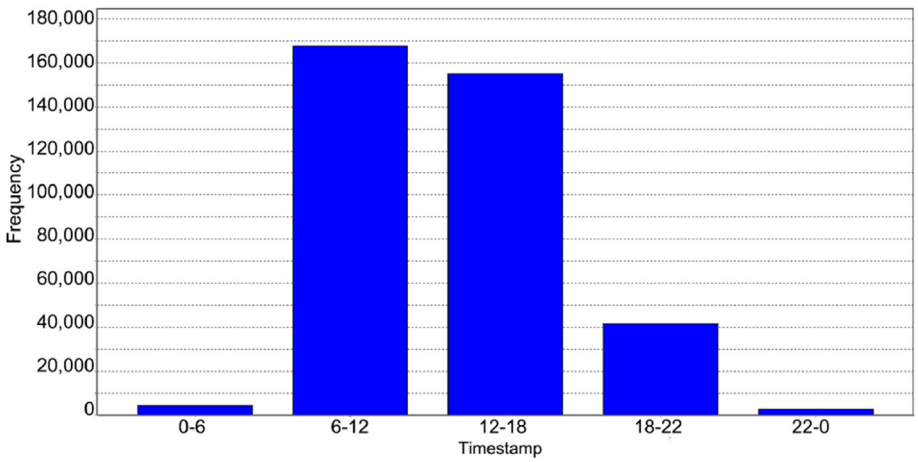

Fig. 3 – Discretization of the number of transaction per day.

Please cite this article in press as: Carminati M, et al., BankSealer: A decision support system for online banking fraud analysis and investigation, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.002

continuity correction factor. Fisher's exact test, which is based on the hypergeometric distribution and does not suffer the limitations of Pearson's χ^2 test, is computationally unfeasible for matrices with high dimensionality and sparsity. Finally, the mutual information test (or G-test), which measures how much knowing one random variable gives information about another random variable, also requires the contingency matrix to have no null values.

In conclusion, it is not easy to estimate the dependence and the correlation between the attributes. The main obstacle is represented by the extremely sparse, imbalanced distribution of the dataset and by the high cardinality of the attributes. However, in the light of the obtained results, we decide to work under the hypothesis of independent and uncorrelated attributes. This approximation allows a much easier visualization and interpretation of models and results, on the top of a reduced temporal and spatial complexity.

3.4. *Clustering analysis*

We want to evaluate the feasibility of finding "classes" of users and quantifying the similarity between profiles in order to separate anomalous users from normal ones. In Fig. 4 we show the Principal Component Analysis (PCA) on two dimensions that we have applied to the users profiles. As it can be seen, they do not seem to form distinct groups. Instead, they tend to congregate in one dense cloud of points, with several outliers points and small groups around it. The results are confirmed by the application of the Hopkins' statistic (Banerjee and Dave, 2004), which measures the clustering tendency. It is clear from the first results that cluster users is not an easy task, because of the homogeneity of users' behavior.

Despite the discouraging results of the PCA, clustering the profiles using a basic *agglomerative hierarchical clustering* led to a satisfactory outcome. After testing the Euclidean distance, we switch to the Mahalanobis distance (Mahalanobis, 1936) which operates with scale-invariant datasets. This means that it manages the differences of scale between the components of the vector representing the profile.

In Fig. 5 we present the dendrograms relative to the application of the hierarchical clustering algorithm. The vertical axes express the linkage (measure of similarity) at which elements are joined: the lower the linkage, the more similar they are. As it can be seen, there are a lot of elements joined with a high similarity. These elements compose the large cluster of very similar profiles observed before. Hence, the majority of users yield densely "connected" areas of the dendogram. On the other hand, users with rare spending profiles tend to form small, isolated groups. These aspects simplify the outlier detection of anomalous users, yet do not create a sharp distinction between users.

In order to exploit the different zones of density of the large cluster, we apply DBSCAN (Ester et al., 1996), which is a density-based clustering algorithm: it grows regions with sufficiently high density into clusters, defined as maximal sets of density-connected points, and discovers clusters of arbitrary shape in spatial databases with noise. We execute several iterations of the DBSCAN algorithm, varying the ε parameter which indicates the maximum distance from a point within which we can consider another point density connected to it. We observe that for low values of ε, we discover only the clusters closer to the center of our data group, while all the external data points are considered noise. For higher values, we find groups for the external data points, while the major group of our data is considered a single huge cluster. By manual inspection, we verify that the generated clusters are reasonable (i.e., composed by similar users). For these reasons, we design a variant of the DBSCAN algorithm (explained in Section 4.2) which tries to separate zones with different density in the big cluster of data by executing multiple iterations of DBSCAN, using increasing ε values.

To evaluate the quality of this clustering and to find a stopping criterion we use the Davies–Bouldin index (Davies and Bouldin, 1979). A low value means good clustering quality (i.e., high intra-cluster similarity and low inter-cluster similarity). In Fig. 6 we show the trend of the Davis–Bouldin index, as the number of clusters and the ε parameter vary. As the number of cluster grows, the index has an increasing trend, reaches a global maximum and then decreases. In other

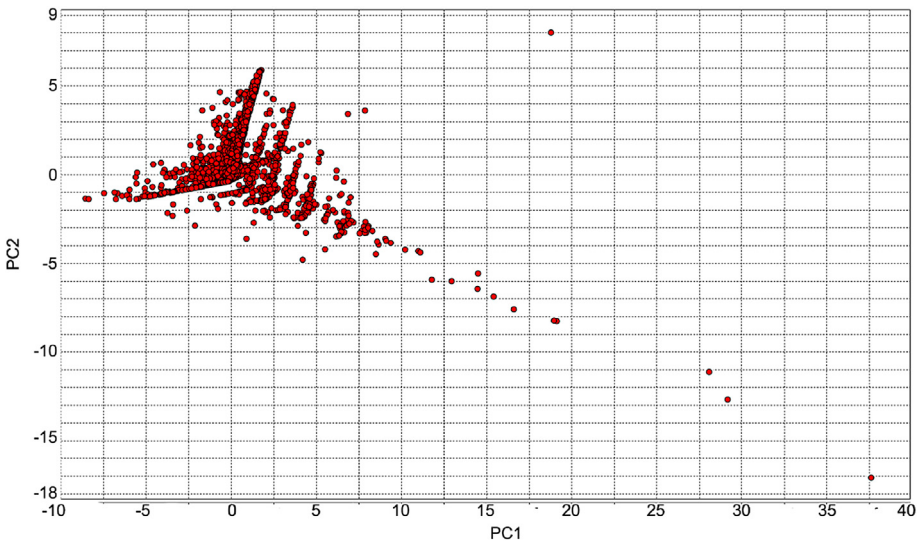

Fig. 4 – PCA of the dataset on two dimensions.

Please cite this article in press as: Carminati M, et al., BankSealer: A decision support system for online banking fraud analysis and investigation, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.002

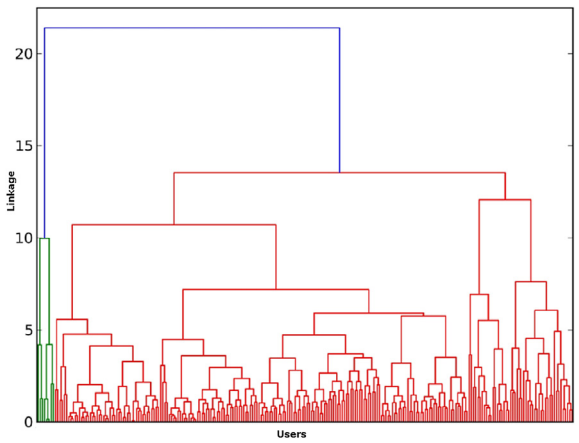

Fig. 5 – Dendrogram representing the hierarchical clustering using the Mahalanobis distance. The different colors represent the clusters obtained cutting dendrograms at 75% linkage.

terms, we have good clustering results either with just a few clusters, or with many clusters. On the right hand side, the lowest value of the index is where we have a high value of the ε parameter and, hence, one large cluster and a few anomaly points or small clusters around it. We obtain the same results applying the Dunn index (Dunn, 1973), which verify the quality of clusters in terms of the ratio between the minimal inter-cluster distance to maximal intra-cluster distance.

In order to understand if user behavior can be modeled, and to give an explanation to clustering results, we analyze the data trying to extract any underlying distribution, by performing the Anderson–Darling best fitting test for normal, exponential, extreme value, log-normal, and Weibull distributions (Anderson and Darling, 1952), using the variant for multivariate distributions. In spite of all our efforts, we were not able to detect any of the aforementioned distributions that can describe the behavior of users. It explains the complexity of finding clear categories.

4. Approach and system description

In this section we describe the main features of BankSealer (for technical details see Carminati et al., 2014), a general framework for online banking fraud and anomaly detection that synthesizes relevant information for each user and transaction. The objective of our system is to be a Decision Support System, able to improve the speed and accuracy of the detection of frauds by the bank analysts characterizing the users of the online banking web application by means of a local, a global and a temporal *profile*, which are built during a training phase. As depicted in Fig. 7, the training phase takes as input a list of *transactions*. Each type of profile extracts different statistical *features* from the transaction attributes, according to the type of model built. BankSealer works both under semi-supervised and unsupervised assumptions and once the profiles are built, it processes new transactions and ranks them according to their anomaly score and the predicted risk of fraud. The *anomaly score* quantifies the statistical likelihood of a transaction being a fraud w.r.t. the learned profiles. The *risk of fraud* prioritizes the transactions by means of anomaly score and amount.

4.1. Local profiling

The goal of this profiling is to characterize each user's spending patterns.

During training, we aggregate the transactions by user and approximate each feature distribution by a histogram. More precisely, we calculate the empirical marginal distribution of the features of each user's transactions. This representation is simple, readable and effective.

At runtime, we calculate the anomaly score of each new transaction using the HBOS (Goldstein and Dengel, 2012) method. The HBOS computes the probability of a transaction according to the marginal distribution learned. We improve the HBOS to account for the variance of each feature and to allow the analyst to weight the features differently according to the institution's priorities.

Training and Feature Extraction. The features are the actual values of all the attributes listed in Table 1. During training we estimate the marginal distribution of each feature using uni-variate histograms. However, we do not consider correlation between features in order to gain lower spatial complexity and better readability of the histograms. Univariate histograms are indeed directly readable by analysts who get a clear idea of the typical behavior by simply looking at the profile. In addition, they easily allow to compute the anomaly score as the sum of the contributions of each feature,

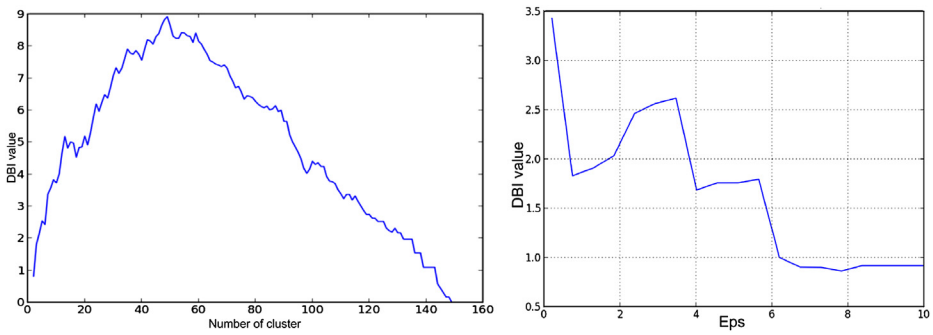

Fig. 6 – Application of the Davis–Bouldin index on the as the number of clusters grows (left) and as the ε parameter varies (right).

Please cite this article in press as: Carminati M, et al., BankSealer: A decision support system for online banking fraud analysis and investigation, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.002

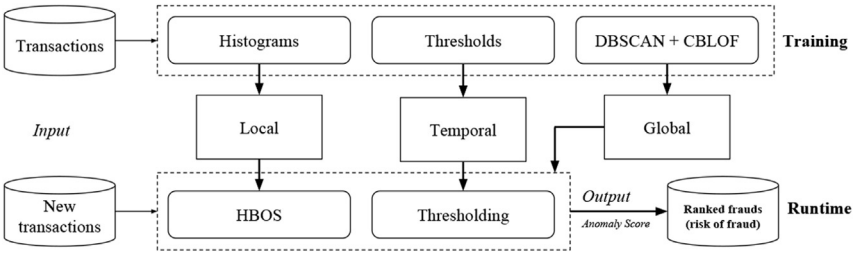

Fig. 7 – BankSealer architecture.

giving an intuitive justification of the resulting anomaly score. For *categorical attributes* (e.g., IP, CC), we count the occurrences of each category. For *numerical attributes* (e.g., Amount, timestamp) we adopt a static binning and count how many values falls inside each bin. After this, we estimate the marginal distribution of the features, computing the relative frequency.

Runtime and Anomaly Score Calculation. We score each new transaction using HBOS (Goldstein and Dengel, 2012). It considers the relative frequency of each bin to quantify the log-likelihood of the transaction to be drawn from the marginal distribution. In other words, for each feature t_i of the transaction t we calculate $\log 1/hist_i(t_i)$, where $hist_i$ indicates the frequency of the i-th feature. The resulting values are summed to form the anomaly score $HBOS_i(t)$. Finally, we compute the risk of fraud multiplying the anomaly score by the transaction amount.

Feature Normalization, Weighting and Rare Values. One of the main drawbacks of the original HBOS is that it does not take into account the variance of the features: we apply a min–max normalization to the histogram, where the minimum is zero, and the maximum is the highest bin.

In addition, we add a weighting coefficient w_i to each feature to allow the analyst to tune the system according to the institution's priorities. In our experiments, however, we fix all the weights at 1, except for IP and IBAN, which are fixed at 0.5 because of their high variance.

To mitigate the problem of feature values never occurred during training for a user (i.e., zero frequency within the local profile), we compute the frequency of unseen values as $k/1-f$, where f is the frequency of that value calculated within a particular cluster, if the global profiling is able to find a cluster for that user. Otherwise, f is calculated on the entire dataset. This method quantifies the "rarity" of a feature value with respect to the global knowledge. The parameter k is an arbitrarily small, non-zero number. In our experiments we set it to 0.01.

Under-trained and New Users. An under-trained user is a user that performed a low number of transactions. In BankSealer this value is a parameter, which empirically we set at $T = 3$ as this is enough to get rid of most of the false positives due to under-training. For under-trained users, we consider their global profile (see Section 4.2) and select a cluster of similar users. For each incoming transaction, our system calculates the anomaly score using the local profile of both the under-trained user and the k nearest neighbor users. For new users, we adopt the same strategy. However, given the absence of a global profile, we consider all the users as neighbors.

4.2. *Global profiling*

The goal of this profiling is to characterize "classes" of spending patterns. During training, we first create a global profile for each user and then cluster the resulting profiles using an iterative version of the DBSCAN. Finally, for each global profile we compute the CBLOF score, which tells the analyst to what extent a profile is anomalous with respect to its closest cluster. The global profile is also leveraged to find local profiles for under-trained or new users. The rationale is that users belonging to the same cluster exhibit spending patterns with similar local profiles.

Training and Feature Extraction. Each user is represented as a feature vector of the six components: average transaction amount, sum of the transaction amounts, average timespan between consecutive transactions, number of transaction executed from foreign countries, number of transaction whose beneficiary account is in a foreign country (only for bank transfers), number of transaction executed.

To find classes of users with similar spending patterns, we apply an iterative version of the DBSCAN, using the Mahalanobis distance between the aforementioned vectors. To mitigate the drawbacks of the classic DBSCAN when applied to skewed and imbalanced datasets such as ours, we run 10 iterations for decreasing values of ε, which is the maximum distance to consider two users as connected (i.e., density similar). At each iteration, we select the largest cluster and apply DBSCAN to its points with the next value of ε. The smaller clusters at each iterations are preserved. We stop the iterations whenever the number of clusters exhibits an abrupt increase (i.e., a knee). In all of our experiments, we empirically observed that this happens at 0.2. As a result, we obtain a set of clusters, which contain similar user profiles.

Anomaly Score Calculation. The global profile is used to assign each user profile a global anomaly score, which tells the analyst how "uncommon" their spending pattern is. For this, we compute the unweighted-CBLOF (Amer and Goldstein, 2012) score, which compute the anomaly as the minimum distance of a user profile from the centroid of the nearest largest cluster, considering small clusters as outliers with respect to large clusters: the more a user profile deviates from the dense cluster of "normal" users, the higher the anomaly score will be.

4.3. *Temporal profiling*

The goal of this profiling is to deal with frauds that exploit the repetition of legitimate-looking transactions over time. We

Please cite this article in press as: Carminati M, et al., BankSealer: A decision support system for online banking fraud analysis and investigation, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.002

construct a temporal profile for each user having a sufficient amount of past transactions, because occasional transactions have a high variance, unsuitable for this kind of analysis. We use a time window, which size can be easily chosen given the hardware resources available (see Section 5). Within such time window, during training, we aggregate the transactions of each user over time with a daily sampling frequency and calculate the sample mean and variance of the numerical features. These are used as thresholds during runtime to calculate the anomaly score.

Training and Feature Extraction. For each user, we extract the following aggregated features: *total amount*, *total* and *maximum daily number of transactions*. During training, we compute the mean and standard deviation for each feature, and set a threshold at mean plus standard deviation.

Runtime and Anomaly Score Calculation. At runtime, for each user and according to the sampling frequency, we calculate the cumulative value for each of the aforementioned features. Then, we sum the positive delta between each cumulative value and the respective threshold to form the anomaly score.

4.4. *Profile updating*

We update the profiles and scores using an exponential discount factor, expressed in terms of a time window W and its respective sampling frequency. Every month we recursively count the values of the features in the previous months discounted by a factor $\lambda = e^{-\tau/W}$, where W ~ 1 year. The rationale is that business activities are typically carried out, throughout a year, with a monthly basis. The parameter τ/W influences the speed with which the exponential decay forgets past data. We empirically set $\tau = 5$, because it seems to best discount past data with respect to time and sampling windows.

5. Experimental evaluation

The goals of our evaluation is to measure (1) the effectiveness and (2) the computational resource requirements of BankSealer in correctly ranking fraudulent transactions never seen before. With respect to Carminati et al. (2014), we repeated the experimental evaluation on a larger dataset, using mixed fraud scenarios to provide further empirical evidence of the viability of our approach. In addition, we provide more details on the effectiveness of our system.

5.1. *Dataset description and fraud scenarios*

In order to build a more consistent model and to reduce the noise due to under-trained and new users, we consider a larger dataset than the one used in Carminati et al. (2014), by considering 9 months of data collected between December and August 2013: we use the firsts months for training, and the last month for testing.

The dataset (described in Section 3) is unlabeled, but it contains no known frauds, as confirmed by the bank. As shown in Table 1, it consists of 718,927 *bank transfers* (92,653 users), 71,362 *prepaid cards* transactions (16,814 users), and 100,688 *phone recharge* (29,298 users) transactions.

As explained in Carminati et al. (2014), the evaluation of BankSealer is particularly difficult because, like any unsupervised analysis tool, it produces novel knowledge. Therefore, we rely on the expertise of domain experts (bank operators) to enrich our testing dataset with synthetic frauds based on fraud scenarios that replicate the typical real attacks performed against online banking users. We focus on the most important and challenging fraud schemes nowadays, those driven by banking trojans (e.g., ZeuS, Citadel) or phishing. Table 2 shows the amounts for each dataset and scenario.

Scenario 1: Info stealing. The trojan modifies the login form to deceive the victim into entering an one time password (OTP) along with the login credentials. This information is use by the fraudster to execute a transaction (with a high amount) towards his account, where the victim never sent money to. We test both the case of the connection coming from a national and foreign IP address. To inject the fraud, we randomly choose a victim from the testing dataset and used a random timestamp for the transaction.

Scenario 2: Transaction Hijacking. The trojan, not the fraudster, hijacks a legitimate bank transfer by manipulating the victim's browser. The challenge is that the connection comes from the victim's computer and IP address. Moreover, we execute the fraudulent transaction within 10 min from a real one, to emulate a session hijacking.

Scenario 3: Stealthy Fraud. The strategy of the fraudster is to execute a series of low–medium amount transactions, repeated daily for one month during working hours, to better blend in. We analyze three cases (very low, low and medium daily amounts). We use the same number of users of the previous scenarios, each performing 30 fraudulent transaction.

Mixed Scenarios: Information stealing and Transaction Hijacking. In addition to considering each scenario independently (as done in Carminati et al., 2014), we evaluate BankSealer with respect to frauds evenly generated from the first two scenarios to provide a more realistic analysis and to give an empirical evidence of the feasibility of our approach.

5.2. *Evaluation approach and metrics*

For the evaluation we followed the same criteria described in Carminati et al. (2014). After training, we inject n fraudulent transactions in the testing dataset. Then, we use the local

Table 2 – Amount transferred for each dataset and scenario. For the bank transfers dataset, the money can be transferred to a national or foreign account, whereas for the phone recharges and prepaid debit cards the money is charged on card.

Fraud scenario	Amount transferred (€)		
	Bank transfers	Phone recharges	Prepaid cards
1: Information stealing	10,000–50,000	250–255	750–1000
2: Transaction hijacking	10,000–50,000	250–255	750–1000
3: Stealthy fraud			
very low amount	50–100	5–10	50–100
low amount	100–500	10–25	100–250
medium amount	500–1000	25–50	250–500

Please cite this article in press as: Carminati M, et al., BankSealer: A decision support system for online banking fraud analysis and investigation, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.002

profiles to rank transactions, and the temporal profiles to rank users, according to the respective anomaly scores. The global profiles are used to mitigate under-training. We analyze the top n transactions (or users) in the ranking, where n is the number of injected transactions (or users). In our case, n accounts for 1% of the testing dataset. Depending on the specific scenario, a fraud may trigger either the local or temporal profile, or both. We count as true positives the number of fraudulent transactions (or users) in the top n positions, and the remainder ones (to the whole n) are false positives. All tests are repeated 10 times and the results are averaged, to avoid biases.

The **overall results**, summarized in Figs. 8 and 9, are consistent with the ones obtained in Carminati et al. (2014), with BankSealer outperforming the state of the art. For instance, Wei et al. (2013) detects up to 60–70% of the frauds with an unreported precision. Remarkably, the effect of under-training is almost negligible.

Experiment 1: Well-trained Users. We first test BankSealer without the noise due to non-well-trained users.

As Table 3 shows, the combination of local and temporal profiles guarantees that frauds are ranked high at either transaction level, thanks to the local profiles, or user level, thanks to the temporal profile.

The results on the information stealing frauds (**Scenario 1**) are very promising. In fact, BankSealer reach a detection rate of 98%, 95%, and 91% and a precision of 97.6%, 94.7%, and 90.7% in the bank transfer, phone recharges, and prepaid cards dataset, respectively.

Transaction hijacking frauds (**Scenario 2**) are particularly challenging, because the malware does not alter the overall amount of transactions performed: It leverages existing transactions by diverting them to a different recipient. The IP address is one of those usually used and, in the case where the recipient fraudulent account is national, these transactions

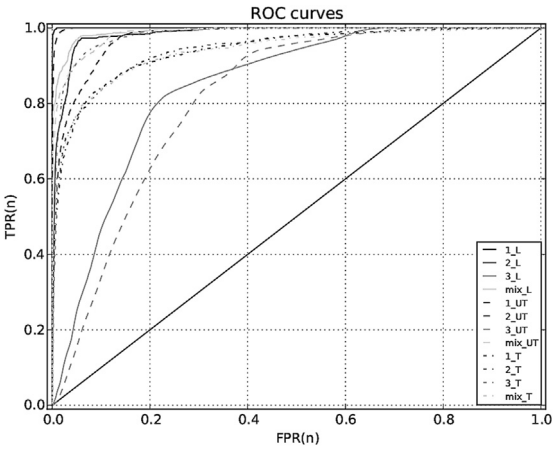

Fig. 8 – True positive rate (TPR) and false positive rate (FPR) for $n \in [1, N]$, where N is the size of the testing dataset. The label "UT" stands for "under-training", "L" for local profile, and "T" for temporal profile. BankSealer shows similar performances in Scenario 1, 2, and mixed, with a high detection rate for low value of n (~90%). Scenario 3 is the most challenging and reach, thanks to the temporal profile, a detection rate of 74%.

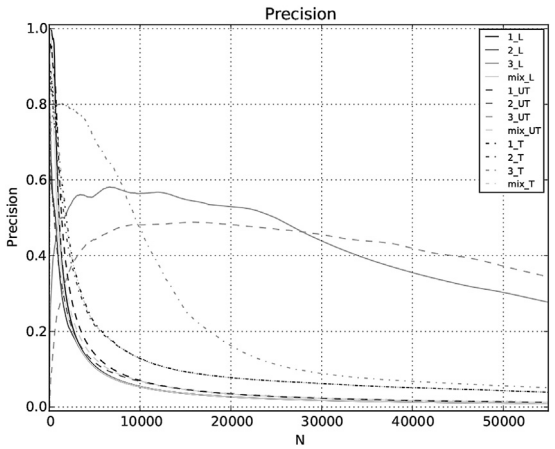

Fig. 9 – Precision for $n \in [1, N]$, where N is the size of the testing dataset. The label "UT" stands for "under-training", "L" for local profile, and "T" for temporal profile. BankSealer shows high precision in Scenario 1, 2, and mixed for low value of n (~90%). The temporal profile improves the overall effectiveness in Scenario 3 up to 80% of precision.

blend in quite easily. However, even for this last case, thanks to the temporal profile anomaly score BankSealer correctly ranks 59% of the frauds, up to 80% with 77.6% of precision for bank transfer dataset.

Stealthy frauds (**Scenario 3**) are also challenging: the local profile performs well when the recipient account is foreign, or with phone recharge and prepaid card frauds. Interestingly, stealthy frauds involving very low amounts (50–100€) are correctly ranked better than transactions involving low amounts (100–500€). The reason is because the very-low amounts are rarer in the dataset, and thus obtain higher anomaly scores. In this scenario, thanks to the temporal profiling, BankSealer correctly ranks up to 74% of the frauds (74.9% of precision) for bank transfer dataset, 100% (99.8% of precision) for the phone recharges dataset, and 93% (92.9% of precision) for the prepaid cards dataset.

As expected, in **Mixed scenarios** BankSealer try to mediate the good performances obtained in scenario 1 with the lower detection rate obtained in scenario 2 reaching a true positive rate near 80%.

Experiment 2: Under-trained and New Users. We evaluate the capabilities of the global profile to lookup a good replacement local profile for under-trained and new users. We proceed similarly to what we did in the previous experiment, injecting 1% of fraudulent transactions, but we spread the injections evenly across well trained, under-trained, and new users.

Table 4 summarizes the percentage of correctly ranked transactions overall, for well-trained users only, for under-trained uses only, and finally for new users only. Performance is similar to the previous experiment, even if the percentage of correctly ranked frauds is obviously lower due to the additional noise.

The fact that under-trained sometimes obtain better ranking than well-trained users, especially when in the attack scenario the frauds are masked to be similar to common transactions, is an artifact due to the fact that in under-

Please cite this article in press as: Carminati M, et al., BankSealer: A decision support system for online banking fraud analysis and investigation, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.002

Table 3 – Experiment 1 results on transactions and users. Blank cells indicate inapplicable dataset–scenario combinations (e.g., phone recharge transactions have no IBAN, phone recharge or prepaid card transactions are only nation-wise). Values in bold represents best results obtained between the local profile (Transactions) and the temporal profile (Users) for each dataset and scenario.

Fraud scenario	Correctly ranked frauds (%)					
	Bank transfers		Phone recharges		Prepaid cards	
	Transactions	Users	Transactions	Users	Transactions	Users
1: Information stealing						
foreign IP, IBAN	**98**	61	**95**	56	**90**	30
foreign IP, national IBAN	**92**	61				
national IP, foreign IBAN	**98**	60	**93**	62	**91**	30
national IP and IBAN	**91**	63				
2: Transaction hijacking						
foreign	**80**	59				
national	28	**59**	70	**71**	**60**	33
3: Stealthy fraud						
foreign, very low amount	**70**	67				
foreign, low amount	**69**	**69**				
foreign, medium amount	71	**73**				
national, very low amount	40	**64**	82	**99**	77	**88**
national, low amount	37	**72**	82	**99**	74	**88**
national, medium amount	40	**74**	83	**100**	82	**93**
Mixed frauds						
national/foreign	**83**	62	**84**	58	**76**	30
national	**70**	63	**81**	63	**70**	28

trained users' profiles even frauds designed to appear as legitimate transactions can receive a high score if the (few) transactions already observed for them are very different from the injected ones. Frauds injected in new users, instead, are ranked incorrectly when are designed to be similar to legitimate transactions. This is due to the fact that, for new users, transactions are tested against the average profile of all transactions in the dataset, and thus transaction with common attributes will receive low scores. In the experiments on the phone recharges and prepaid card dataset, we obtain a lower percentage of correctly ranked frauds than those in Table 3. On the other hand, for the stealthy fraud (**Scenario 3**) the percentages are considerably lower. A factor is the huge number of under-trained users in these two datasets.

Table 4 – Experiment 2 results on well-trained, under-trained, new users only, and overall. As in Table 3, blank cells indicate inapplicable dataset–scenario combinations.

Fraud scenario	Correctly ranked frauds (%)											
	Bank transfers				Phone recharges				Prepaid cards			
	Overall	Well trained	Under-trained	New	Overall	Well trained	Under-trained	New	Overall	Well trained	Under-trained	New
1: Information stealing												
foreign IP, IBAN	96	96	99	93	71	81	95	11	65	67	80	0
foreign IP, national IBAN	75	81	91	52								
national IP, foreign IBAN	95	98	100	85	59	81	95	0	69	71	77	0
national IP and IBAN	72	84	93	42								
2: Transaction hijacking												
foreign	65	46	90	60								
national	25	17	61	3	22	13	51	0	30	18	64	0
3: Stealthy fraud												
foreign, very low amount	59	44	90	44								
foreign, low amount	68	39	91	60								
foreign, medium amount	68	42	93	70								
national, very low amount	31	20	72	5	35	39	72	3	46	40	93	0
national, low amount	31	28	70	3	36	41	36	0	47	35	98	35
national, medium amount	35	25	74	7	39	35	81	1	60	53	92	5
Mixed frauds												
national/foreign	71	73	85	58	50	60	84	8	32	33	63	0
national	60	67	76	43	43	48	58	3	26	26	44	0

Please cite this article in press as: Carminati M, et al., BankSealer: A decision support system for online banking fraud analysis and investigation, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.002

Experiment 3: Performance and Resource Requirements. To test the performance of BankSealer, we measured both the computational requirements at runtime (as this is a constraint for the practical use of the system in production), and peak memory requirements at training time (as this is a constraint on the dimension of the dataset that can be handled).

For computational power requirements, we test the time to analyze one day and one month of data, both with and without the handling of under-trained and new users explained in Section 4.1. Our experiments have been executed on a desktop-class machine with a quad-core, 3.40 Ghz Intel i5-3570 CPU, 16 GB of RAM, running Linux 3.7.10 × 86_64. Processing times are taken using the *time* library. The results are listed in Table 5. As we can see, the processing time varies on the basis of the context being tested, and there is a significant difference induced by the handling of the bank transfer dataset and under-trained/new users. In production BankSealer will analyze transactions day by day. Therefore, the maximum time required would be 4 min per day for the bank transfers context. In conclusion, BankSealer is suitable for online fraud monitoring.

We test the scalability of the system by measuring RAM consumption at training time, which is the most memory-intensive phase. We use the bank transfers dataset, the largest one. We rely on memory-profiler and psutil. As Fig. 10 shows, the peak RAM consumption increases almost linearly with the number of days, and quadratically with the number of users. This is expected, as the most memory-intensive data structure is the distance matrix, a square matrix of the size of the number of users.

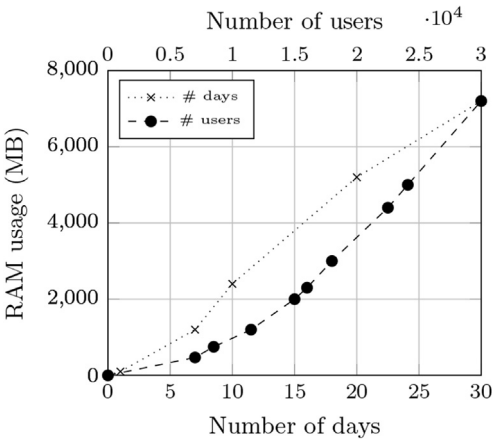

Fig. 10 – RAM requirements for increasing values of W and users profiled (left) Time requirements for runtime analysis of different testing interval.

6. Related work and discussion

Fraud detection, mainly focused on credit card fraud, is a wide research topic, for which we refer the reader to Chandola et al. (2009), Phua et al. and Bolton and David.

Limiting our review to the field to banking fraud detection, supervised approaches based on contrast patterns and contrast sets (e.g., Bay and Pazzani, 2001) have been applied. Along a similar line Aggelis (2006) proposed a rule-based Internet banking fraud detection system. The proposed technique does not work in real time and thus is profoundly different from ours. Also, supervised techniques require labeled samples, differently from BankSealer.

The unsupervised approach presented in Wei et al. (2013) is interesting as it mitigates the shortcomings of contrast pattern mining by considering the dependence between events at different points in time. However, Wei et al. (2013) deal with the logs of the online banking web application, and thus does not detect frauds as much as irregular interactions with the application. Among the unsupervised learning methods, Mhamane and Lobo (2012) proposed an effective detection mechanism to identify legitimate users and trace their unlawful activities using Hidden Markov Model HMMs. Kovach and Ruggiero (2011) is based on an unsupervised modeling of local and global observations of users' behavior, and relies on differential analysis to detect frauds as deviations from normal behavior. This evidence is strengthened or weakened by the users' global behavior. The major drawback of this approach is that the data collection must happen on the client side, which makes it cumbersome to deploy in large, real-world scenarios. In general, a major difference between existing unsupervised and semi-supervised approaches and BankSealer is that they do not give the analyst a motivation for the analysis results, making manual investigation and confirmation more difficult.

The main barrier in this research field is the lack of publicly available, real-world frauds and a ground truth for validation. Indeed, we had to resort to synthetically generated frauds. The absence of non-anonymized text fields does not allow us to analyze, for instance, their semantics. In future extensions, BankSealer will compute the models on the bank side and export privacy-preserving statistics for evaluation.

The prototype is also constrained by the RAM consumption of the clustering phase. This technical limitation can be mitigated by applying a distribute version of presented algorithms.

Table 5 – Computation time required at runtime under various conditions. In the typical use case, the system works on a daily basis, thus requiring 6 min (worst case).

Testing interval	Elapsed time		
	Bank transfers	Phone recharge	Prepaid cards
1 day, no under-trained/new	1′00″	0′18″	0′07″
1 day, under-trained/new	4′00″	0′24″	0′10″
1 month, no under-trained/new	6′00″	0′30″	0′12″
1 month, under-trained/new	93′00″	2′30″	1′00″

7. Conclusions

BankSealer is an effective online banking semi-supervised and unsupervised fraud and anomaly detection approach that helps the analyst in *understanding* the reasons behind fraud alerts. We developed it based on real-world (albeit anonymized) data and requirements.

Please cite this article in press as: Carminati M, et al., BankSealer: A decision support system for online banking fraud analysis and investigation, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.002

We performed an in-depth technical analysis of the dataset, which allowed us to understand its main features, to generalize them and to develop BankSealer in a data-driven way. This allowed us to mitigate challenges such as the scarcity of training data and their extreme statistical imbalance.

We evaluated the developed system through real-world data and a set of realistic attacks, validated by domain experts.

BankSealer is currently deployed as a pilot project in the large national bank with which we cooperated in building it. Thanks to the data we are receiving and recording from this deployment, a short-term future development is to consider the feedback given by the analyst on the detected anomalies to improve the results.

Other future expansions are a semantic analysis of the text attributes, and a more precise estimation of the number of transactions required to fully train a profile.

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007–2013) under grant agreement nr. 257007, as well as from the TENACE PRIN Project (n. 20103P34XC) funded by the Italian Ministry of Education, University and Research.

REFERENCES

Aggelis V. Offline internet banking fraud detection. In: ARES, IEEE Computer Society; 2006. p. 904–5.

Amer M, Goldstein M. Nearest-neighbor and clustering based anomaly detection algorithms for RapidMiner. 2012. p. 1–12.

Anderson TW, Darling DA. Asymptotic theory of certain "Goodness of Fit" criteria based on stochastic processes. Ann Math Stat 1952;23(2):193–212.

Banerjee A, Dave R. Validating clusters using the Hopkins statistic. In: Fuzzy systems, 2004. Proc. 2004 IEEE Intl. Conf. on, vol. 1; 2004.

Bay SD, Pazzani MJ. Detecting group differences: mining contrast sets. Data Min Knowl Discov 2001;5(3):213–46.

R. J. Bolton, David, Statistical fraud detection: a review, Stat Sci 17.

Carminati M, Caron R, Maggi F, Epifani I, Zanero S. BankSealer: an online banking fraud analysis and decision support system. In: ICT systems security and privacy protection – 29th IFIP TC 11 international conference, SEC 2014, proceedings, Springer Berlin Heidelberg; 2014.

Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. ACM Comput Surv 2009;41:15:1–15:58.

Conover W. Practical nonparametric statistics. Wiley series in probability and statistics. 3rd ed. New York, NY [u.a.]: Wiley; 1999.

Davies David L, Bouldin Donald W. A cluster separation measure. IEEE Trans Pattern Anal Mach Intell April 2 1979;PAMI-1:224–7. http://dx.doi.org/10.1109/TPAMI.1979.4766909. issn 0162-8828.

Dunn Joseph C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. Taylor & Francis; 1973.

Ester Martin, Kriegel Hans Peter, Sander Jörg, Xu Xiaowei. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (Kdd), vol. 96; 1996. p. 226–31.

Goldstein M, Dengel A. Histogram-based Outlier Score (HBOS): a Fast unsupervised anomaly detection algorithm. 2012.

Kovach S, Ruggiero W. Online banking fraud detection based on local and global behavior. In: ICDS 2011: the Fifth Intl. Conf. on digital society; 2011. p. 166–71.

Mahalanobis PC. On the generalized distance in statistics. In: Proc. of the National Institute of Science of India; 1936. p. 49–55.

Mhamane S, Lobo L. Internet banking fraud detection using HMM. In: Computing Communication Networking Technologies (ICCCNT), 2012 Third Intl. Conf. On; 2012. p. 1–4.

Myers JL, Well AD. Research design and statistical analysis. New Jersey: Lawrence Erlbaum Associates; 2003.

C. Phua, V. C. S. Lee, K. Smith-Miles, R. W. Gayler, A Comprehensive survey of data mining-based fraud detection research, CoRR.

Wei W, Li J, Cao L, Ou Y, Chen J. Effective detection of sophisticated online banking fraud on extremely imbalanced data. World Wide Web 2013;16(4):449–75.

Michele Carminati holds an M.Sc. in Computer Engineering (cum laude) from Politecnico di Milano. Since November 2013 he is a PhD student in Computer Engineering at Politecnico di Milano. His research interests are mainly focused on computer security and in particular on financial malware analysis and Internet banking fraud detection.

Roberto Caron received an M.Sc. in Computer Engineering both from Politecnico di Milano. Since November 2013 he works as a consultant at Reply S.p.A., a large Italian system integrator.

Federico Maggi is an Assistant Professor at Dipartimento di Elettronica, Informazione e Bioingegneria of Politecnico di Milano in Italy. He holds a Ph.D. degree in Computer Engineering (cum laude) from the same university. His current research interests revolve around web and mobile security, and anomaly detection.

Stefano Zanero holds a Ph.D. degree in Computer Engineering (cum laude) from Politecnico di Milano, where he is currently a tenured assistant professor. His research interests focus on systems security, malware analysis, and in general data analysis applied to security.

Ilenia Epifani is a tenured assistant professor in probability and mathematical Statistics at Politecnico di Milano. She holds a PhD in Statistics from the University of Trento, Italy.