Early Security Classification of Skype Users via Machine Learning

Anna Leontjeva*
University of Tartu
Ülikooli 2, Estonia
anna.leontjeva@ut.ee

Moises Goldszmidt
Microsoft Research
Silicon Valley
1288 Pear Avenue
Mountain View, CA 94043
moises@microsoft.com

Yinglian Xie
Microsoft Research
Silicon Valley
1288 Pear Avenue
Mountain View, CA 94043
yxie@microsoft.com

Fang Yu
Microsoft Research
Silicon Valley
1288 Pear Avenue
Mountain View, CA 94043
fangyu@microsoft.com

Martín Abadi
Microsoft Research
Silicon Valley
1288 Pear Avenue
Mountain View, CA 94043
abadi@microsoft.com

ABSTRACT

We investigate possible improvements in online fraud detection based on information about users and their interactions. We develop, apply, and evaluate our methods in the context of Skype. Specifically, in Skype, we aim to provide tools that identify fraudsters that have eluded the first line of detection systems and have been active for months. Our approach to automation is based on machine learning methods. We rely on a variety of features present in the data, including static user profiles (e.g., age), dynamic product usage (e.g., time series of calls), local social behavior (addition/deletion of friends), and global social features (e.g., PageRank). We introduce new techniques for pre-processing the dynamic (time series) features and fusing them with social features. We provide a thorough analysis of the usefulness of the different categories of features and of the effectiveness of our new techniques.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Security and Protection; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*user profiles and alert services*

Keywords

fraud detection; machine learning; social graph

*Corresponding author. This work was done while Anna was an intern at Microsoft Research.

AISec'13, November 4, 2013, Berlin, Germany.

http://dx.doi.org/10.1145/2517312.2517322

1. INTRODUCTION

Fraud detection has attracted considerable attention in both academia and industry (e.g., [1, 26, 8]). Techniques for fraud detection rely on a wide variety of data, often tied to specific applications. Each application may in fact give rise to several different kinds of data, even more so as fraud schemes evolve over time. In addition, some of the data may be voluminous, incomplete, and not fully reliable. Therefore, one strategic element in fraud detection is the development of approaches for fusing disparate information sources, and for making sense of the aggregate information, robustly and at scale.

In this paper, we propose and evaluate an approach to this problem based on supervised machine learning. The approach combines information from diverse sources such as static user profiles, time series that represent user activities, and the results of algorithms that analyze user social connections. Separately, these sources can be insufficient for fraud detection. The data are often sparse, containing missing values, and the abnormal patterns associated with attacks may manifest themselves in different parts of the data. Our work explores a new way to fuse the data sources, synergistically, for the purpose of fraud detection.

We develop and study our approach in the context of Skype. Skype is one of the largest providers of Internet communication software, and is the target of varied fraudulent activities. Accordingly, Skype employs sophisticated techniques for detecting and thwarting fraud. As a result, the majority of fraudulent users are detected within one day.

The aim of our work is to go beyond the present, sophisticated defenses, and to detect "stealthy" fraudulent users, namely, those that manage to fool those defenses for a relatively long period of time. Our concrete objective is to catch these stealthy fraudulent users within the first 4 months of activity. Our results indicate that, with our methods, we are able to detect 68% of these users with a 5% false positive rate; and we are able to reduce by 2.3 times the number of these users active for over 10 months.

We analyze a large dataset from Skype. This dataset represents a carefully anonymized snapshot that contains

the Skype contact network, time series of the utilization of Skype products, and time series of the social/contact request activity of users. It does not contain information about individual calls and their contents.

To this data, we apply techniques and algorithms for supervised learning, but only after non-trivial pre-processing. As will be seen, we have relatively short time series to work with. In order to extract as much information as possible from them, we use a set of Hidden Markov Models (HMMs) that produce odds of fraudulent vs. normal activities (both in terms of service utilization and social activities). A similar approach was used by one of the authors [12] for the purpose of predicting failure in datacenter disks. The application to fraud detection, and its evaluation in the context of real data (Skype), is new. Similarly, we adapt techniques from work on social graphs [17] in order to estimate the reputation of users according to their social connections. The application to fraudulent activity in Skype is also new, as is the combination with time series.

In summary, the contributions of this paper are:

- an architecture and methods to process and fuse information from a variety of sources in order to identify fraudulent users;
- an evaluation of the efficacy of the methods on real data;
- the quantification of the impact of each one of the different sources of information for the task of detecting fraudulent users.

The rest of the paper is organized as follows. Section 2 briefly reviews the relevant literature. Section 3 formally states the problem that we address, and describes the input data and its features. Section 4 describes our approach. Section 5 presents experimental results. Section 6 summarizes our main conclusions and discusses directions for further research.

2. RELATED WORK

In this section we briefly review approaches and techniques related to our study. We concentrate on research in machine learning and in social network analysis, with a focus on work on fraud detection.

Skype allows users to communicate with each other via text messaging, audio, and video calls. It supports both free services as well as paid-for products and subscriptions. Skype is not a telecommunication provider but, in some ways, fraud in Skype resembles fraud in telecommunication services. Previous work (e.g., [15, 14, 9]) has extensively studied fraud in telecommunications, and in some cases has explored data mining and machine learning techniques [25, 10, 20]. However, these previous studies have mostly leveraged the static user profiles and usage features for detection. In our work, we consider usage features based on time series, which provide richer and more fine-grained information than static usage features represented by simple statistics (e.g., mean and standard deviations). In addition, we study a broader set of features, including local and global social features.

In Skype, users add each other in friend lists and employ multiple channels to communicate. Thus, the Skype communication graph can be viewed as a social network graph, to some extent. Social network features were studied in the literature [21, 28, 7] as a tool for fraud detection. Their value motivates us to explore them within a general machine learning framework.

In order to fuse dynamic time-series usage features and other static profile features (for both training and detection), we combined the use of Hidden Markov Models [24] and the (log-odds) comparison to normal users and a classification framework based on Random Forest [3]. A number of previous studies discuss the combination of different machine learning methods [16, 5]. In particular, the general subject of classifier combinations has been considered and justified theoretically by Kittler and Hatef [19]. Furthermore, our approach can be regarded as a simple way of cascading classifiers advocated by Viola and Jones in the context of vision applications [27], except that we are using the cascade to transform different inputs rather than to select regions of the feature space.

In this paper, we do not address how fraudsters might adapt and attempt to evade our detection techniques (cf. [22, 18]). We hope that our use of a large number of features would make evasive actions rather costly. Further investigation of such questions may be worthwhile.

Despite the existence and the deployment of various approaches to fraud detection, many financial institutions and companies still rely on manual review in addition to automatic screening, spending more than half of their fraud management budget on review-staff costs. Recent reports indicate that many financial companies lose 0.9% of their online revenue to fraud [1], suggesting that fraud detection is still an important problem that requires improved solutions.

3. PROBLEM STATEMENT, DATA, FEATURES

Fraud is commonly defined as intentionally deceiving another person or organization and causing them to suffer a loss. In this study, we define a fraudulent user as a registered user who intentionally deceives another user or a service provider, causing them to suffer a loss. There is a wide variety of fraud schemes [23]. The kinds of fraud relevant to Skype include, in particular, credit card fraud and other online payment fraud, as well as account abuse such as spam instant messages.

Our aim is to catch those fraudsters that elude the first line of defenses at Skype. We define our target as those fraudsters that engage in activity for over K months (after creating their accounts), where K is a parameter which in this study we will set to 4 months. Our strategy is to combine information from different kinds of activities, both social (e.g., requests to be accepted as contact) and on the use of Skype products. To this end, we cast the problem of fraud detection as that of automated pattern classification.

Specifically, we are interested in automatically deriving a function f (the classifier) that, given a user u in the set of users U, classifies the users as either *fraudulent* or *normal*; that is, $f : U \rightarrow \{\text{fraud}, \text{normal}\}$. Each user u_i is represented as a *feature vector* $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})$, where the features are the characteristics of the user extracted from an input dataset D. The features thus represent the information that our classifier is combining in order to detect the fraudsters.

Table 1: Sets of features (with activity logs in *italic*)

Profile set	gender
	age
	country
	OS platform
	...
Skype product usage	*connected days*
	audio call days
	video call days
	chat days
Local social activity	*additions by a user*
	deletions by a user
	additions of a user
	deletions of a user
	accept rate (%)
	degree
Global social activity	full contact graph

Figure 1 depicts the entire workflow of the process. In the rest of this section we describe the first two levels, i.e., the data and the feature set, leaving the pre-processing of the features for the next section. The measures for the *accuracy* of our classifier, namely, how good is f at detecting fraudulent users, will be presented in Section 5.1.

The input dataset D is a snapshot of the data collected by Skype that encompasses all registered users and part of their activity both in terms of the usage of Skype products and in terms of the user-to-user contact-list requests over a period of time.

Skype takes the privacy of its users very seriously, and we implemented rigorous and carefully considered safeguards throughout this study in order to protect the privacy of Skype users. For example, all Skype IDs were anonymized using a one-way cryptographic salted hash function. None of the Skype usage data contained information about individual Skype communications, such as the parties involved in a communication, the content of communications, or the time and date of communications. Rather, it merely contained the number of days in each month that a Skype user used a particular communications feature, such as Skype chat, Skype video calls, and Skype In and Skype Out calls. Furthermore, the study's data was maintained on separately administered computer system, and access to it was strictly limited to the study's authors. Finally, we plan to erase the data when it is no longer needed for research.

Relying on the user-to-user contact requests, we built a directed graph that consists of 677.8 million nodes and 4.2 billion directed edges. We also have timestamps of edge creations and deletions. Edge creation means that a user (the sender) sends a friendship request to another user (the receiver). Once the request is sent, we count it as an edge creation from the sender to the receiver. After the request, there are two scenarios: either the receiver accepts it or not. In case of acceptance, we consider it as another edge creation from the receiver to the sender. We handle edge deletions by the same principle.

Additionally, we have 10.8 million labels for fraudulent users, which we use both in training, to induce the classifier, and for testing its accuracy (see Subsection 5.1).

The labels further identify four different types of fraud schemes. Skype's definitions and procedures for these four different types of schemes is internal confidential information, which therefore we do not discuss further. Moreover, in our work, we choose not to rely on Skype's informal intent in those definitions, nor on Skype's software for each of the different types, in order to develop robust, self-contained methods.

From dataset D we extract information to construct a feature vector $(x_{i1}, x_{i2}, \ldots, x_{im})$ for each user. We divide the features into different types according to the kind of information they provide as well as how much processing is needed to transform raw data into the inputs to the classifier. Table 1 summarizes the various types that we proceed to describe.

- Features in the first set are directly extracted from the profile information provided by the users at account-registration time. They include user age, gender, country of residence, etc. Such information is not compulsory and it is not verified by Skype. For many users, this data is incomplete or totally missing.

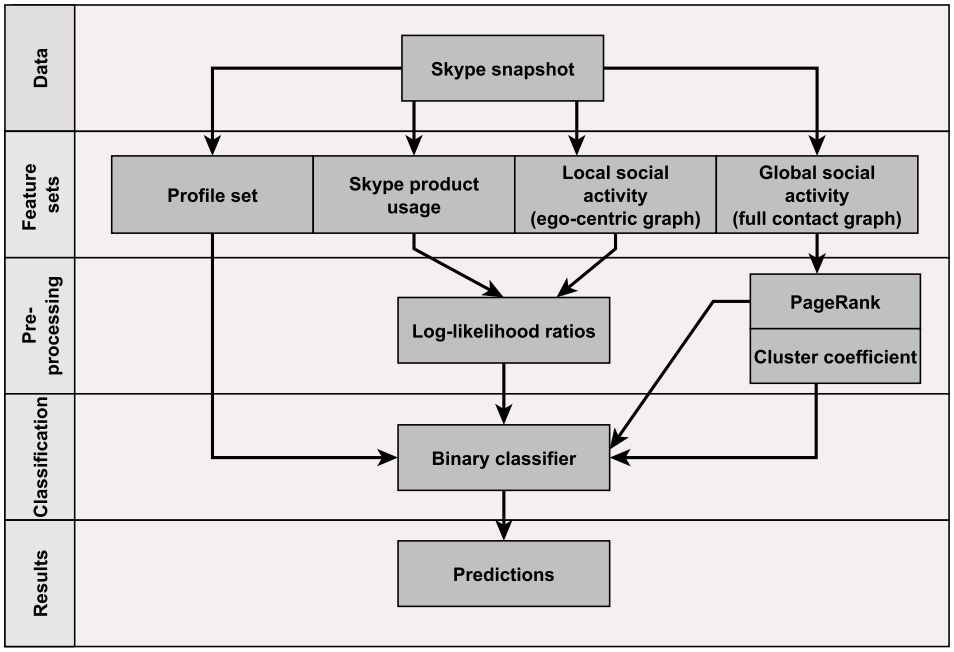
- Features in the second set encode information about *Skype product usage*. We refer to this set of features as *activity logs*: the total number of days per month a user was active using a specific product. For example, *connected days* represent the number of days per month a user was logged in Skype, while *video calls* shows on how many days the user made a video call. Note that these features are in the form of time series, and will be further processed before we use them as input to our classifier (see Figure 1 and Section 4). We introduce some notation: consider a fixed set of *activities* $\{a_1, a_2, \ldots, a_L\}$. For each user u_i and each activity a_j we have an *activity log* $\ell(u_i, a_j) = (y_1, y_2, \ldots, y_K)$, where y_k represents the value of activity a_j for user u_i during the k-th month since its account creation. Because of the limitations of our dataset, activity logs do not

  indicate which pairs of users actually communicate, nor distinguish a user who makes hundreds of calls per day from a user who makes just one.

- Features in the third set, which we name *local social activity*, capture information about the "social" activity from each user's perspective or so-called *egocentric network* [11]. From the snapshot of the social network, we extract information about additions and deletions of contacts by a user as well as how many contacts added or deleted this user. These features are also expressed as monthly *activity logs*, where the value is the number of times a user was involved in the activity, e.g., how many times the user was deleted during the month. Note that it does not necessarily mean that the user deletes the initiator of the deletion in return. We also include further information such as degree and users' acceptance rate. Degree is the number of contacts in the users' list and acceptance rate is the percentage of outbound friend requests being accepted by others.

- Features in the last set represent *global social activity*. To compute these features, we need a full social graph of users based on contact-list requests. In particular,

Figure 1: Entire workflow for the classification process.



we compute the *PageRank* and the *local clustering coefficient* for each user. These features provide information at a much more global scale than those extracted from the egocentric network (see subsections 4.3) and 4.2).

As we will see in Section 5, it is the combination of these different types of features that will enable a better detection of fraudulent activity.

4. THE CLASSIFIER AND ITS INPUTS

In our initial experiments, we benchmarked several classifiers, including Random Forest, SVM, and logistic regression (using both lasso and ridge regularization). While a thorough comparison of these classifiers is beyond the scope of this paper, and in particular we did not attempt to optimize their parameters, we estimated their accuracy via 5-fold cross-validation. Since Random Forest yielded superior results (by about 10%), we decided to adopt it for this work.

Random Forest, first introduced by Breiman and Cutler, is a modification of bagging that builds an ensemble of decision tree classifiers [3]. The idea behind the algorithm is to reduce variance by reducing the correlation between trees. This reduction is achieved by growing unpruned trees with random feature selection. The final classification result is derived by combining the classification results of individual trees through major voting [13]. Since each tree is trained independently, the training procedure is fast even for large data sets with many features.

The various pieces of information described in Section 3 constitute the inputs to our classifier. The user-profile features are in the form of categorical data that can be directly fed into the classifier. The remaining features (the activity logs and the local and the global social features) require further processing before they can be fed into the classifier. Part of the novelty in our approach stems from this pre-processing, which we explain in the rest of the section. The classifier is trained in the usual way using 5-fold cross-validation.

4.1 Log-likelihood ratios of activity logs

The user activity logs include information about Skype product usage as well as the addition and deletion of contacts (see Section 3). We represent this information as a set of time series. In order to combine the time series with other static, categorical features, we further transform the data into a set of features represented as log-likelihood ratios.

We perform this data transformation for each type of user activity (e.g., audio calls, video calls, contact additions) separately. This process consists of the following steps:

1. Given a specific type of user activity (e.g., audio calls), build two different models of activities, one for the normal users and the other for the known fraudsters, based on training data.

2. For each user (to be classified), produce a score using the above two models, representing how close this particular user is to the activities of a fraudster vs. the activities of a normal user.

3. Feed this score into the classifier, where the score will be combined with other features (and the scores from other activities) in order to perform the classification.

This approach leverages the information from an entire time series in order to produce a score that will be input to the classifier. It is fundamentally different from previous approaches that summarize a time series using simple statistics such as its mean or its standard deviation, and it can yield better results [12].

One way to instantiate the steps above is to use probabilistic models for the activity logs. Let $P_F(\ell(u_i, a_j))$ denote the probability that the data in the activity log $\ell(u_i, a_j)$ was generated from a fraudulent user, and correspondingly let $P_N(\ell(u_i, a_j))$ be the probability that the data was generated by a normal user. Then the score we want in Step 2 above is the log-likelihood ratio:

$$LLR(u_i, a_j) = \log\left(\frac{P_F(\ell(u_i, a_j))}{P_N(\ell(u_i, a_j))}\right) \quad (1)$$

This ratio reflects how much more probable it is that the data in $\ell(u_i, a_j)$ comes from a fraudster than from a normal user. The bigger LLR, the more evidence that user u_i acts like a fraudster.

What is left for us to describe is how we automatically get P_F and P_N from training data. There are many viable statistical models for this task; in this paper we explore the use of Hidden Markov Models (HMMs) [24]. It is beyond the scope of this paper to provide a tutorial on HMMs, as these are well known. However, we explain these models informally, and describe the specific way in which we use them and parameterize them for our purposes.

An HMM is a statistical model in which the system being modeled is assumed to produce output signals according to a Markov process governed by unobservable (hidden) states. In our case, the observable signals are the levels of user's activity $O_j = (O_1, O_2, O_3, \ldots)$. Thus, O_j corresponds to the outputs in the activity log reflected in the raw data—one for each month. The hidden states correspond to the user "intensity" for that month. As explained below, we consider two possible states: high intensity and low intensity (i.e., the user plans a high or a low level of engagement for the month).

In order to specify an HMM mathematically, we need three components:

1. the probability distribution of the initial state: $P(S_1 = s)$, for all $s \in S$,
2. the transition probabilities between states: $P(S_{t+1} = s'|S_t = s)$, for all $s, s' \in S$,
3. the emission probabilities (of the observable data) given a specific state: $P(O_t = o|S_t = s)$, for all $s \in S$, $o \in O$.

Given these, $P(\ell(u_i, a_j))$ with say $\ell(u_i, a_j) = (y_1, y_2, y_3, y_4)$ is equal to

$$\sum_{S_1,S_2,S_3,S_4} P(S_1)P(y_1|S_1)(\prod_{1\le t\le 3} P(S_{t+1}|S_t)P(y_{t+1}|S_{t+1})) \quad (2)$$

In our case, after some initial experimentation, we settled for the following parameters and distributions:

- The user can be in one of two hidden states (which represent intensity levels), and the initial distribution $P(S_1)$ is a binomial distribution with parameter p_i.
- Correspondingly, the transition probabilities $P(S_{t+1} = s'|S_t = s)$ are also binomial distributions with parameters $\{p_{s1}, p_{s2}\}$.
- We discretize the space of observables (level of activity) into three possible ones: O_1 (no activity in this month), O_2 (between one and five days of activity), and O_3 more than 5 days of activity in the month. Correspondingly, the conditional probability distributions are multinomials with parameters $\{p_{s1}(O_1), p_{s1}(O_2), p_{s1}(O_3), p_{s2}(O_1), p_{s2}(O_2), p_{s2}(O_3)\}$.

We use the standard Baum-Welch algorithm described by Rabiner [24] in order to fit the maximum likelihood parameters listed above, with the same training data as for the rest of the classification training. Once the parameters are fitted, the computation of Equation 2 employs a standard dynamic programming algorithm [24].

Thus, we are essentially cascading models. The log-likelihood computations for each activity log (Eq. 1) provide a pre-classification from which the scores can be used and compared at the next level by the final classifier.

4.2 PageRank

The PageRank algorithm is widely used for ranking Web pages based on Web link structures [4]. Pages with high PageRank scores are usually authoritative pages, with either many incoming links or links from other important pages. Pages with low PageRank scores are usually of low importance or spam pages. Recently, PageRank has also been used for identifying spammers on social graphs [6, 17]. In that work, the PageRank algorithm is run on a reversed email graph. More specifically, if user A sends an email to user B, one places an edge from B to A in the reversed email graph. A spammer that sends many spam emails but receives very few emails will have a high number of incoming edges in the reserve email graph, hence will likely have a high PageRank score.

In our work, we adopt a similar approach and compute PageRank scores on the reversed Skype user contact graph. More specifically, each user represents a node in the graph. If user u_i sends a friend request to user u_j, we place a link from user u_j to user u_i. Thus, users with higher PageRank scores are likely to be those that sent out a large number of friend requests. On such a reversed contact graph, we assign a uniform score to each user initially, then perform iterative PageRank computation until the PageRank scores converge.

In each iteration, each user propagates her scores to neighbors (friends). At the end of the iteration, a user u's new PageRank score $R_{u,i+1}$ is computed as:

$$R_{u,i+1} = 1 - d + d \sum_{\{X:e_{Xu}\in E\}} \frac{R_{X,i}}{outdegree(X)}$$

where d is the damping factor usually set to be 0.85 [4], $R_{X,i}$ is the score of the user X after the previous iteration, and $\{X : e_{Xu} \in E\}$ is the set of users in the graph having directed edges pointing to u (friends who received contact requests from u).

Thus, we compute a PageRank score for every user on the Skype communication graph. This score will be the input to the classifier that represents the user's global social activity.

4.3 The local clustering coefficient

Another input to our classifier that comes from the full contact graph is the local clustering coefficient [29]. The local clustering coefficient is the ratio of the number of connections in the neighborhood of a user to the number of connections in a fully connected neighborhood. Intuitively, it is a measure of how tightly the neighborhood of the user is connected. In terms of the Skype network, each user's

Table 2: Confusion matrix

Actual	Predicted Fraudulent	Regular
Fraudulent	True Positive (TP)	False Negative (FN)
Regular	False Positive (FP)	True Negative (TN)

contacts constitute its neighborhood. If the neighborhood is fully connected, the clustering coefficient is 1; on the other hand, if the clustering coefficient is close to 0, there are no connections between contacts of the user.

More formally, let k_i be the size of the neighborhood of user u_i. Let n_i be the number of directed links between those k_i users. Then, the clustering coefficient $cc(u_i)$ of user u_i is defined as:

$$cc(u_i) = \begin{cases} 0 & \text{if } k_i < 2 \\ n_i/(k_i(k_i-1)) & \text{otherwise} \end{cases}$$

g coefficient is greater in social networks than in a random network [29]. As fraudulent users often seem to add unrelated users to their contact lists, their clustering coefficients are often lower than those of normal users.

5. EXPERIMENTAL RESULTS

For the experimental evaluation of our techniques, we consider a sample of Skype users that consists of $100,000$ randomly chosen users labeled as fraudulent by Skype, and the same number of randomly chosen users not so labeled. From this sample, we include a user u in our study if u is not blocked within 4 months since its account creation. We end up with $34,000$ such users. In this set, the ratio of users labeled as fraudulent to other users is 1:6.

Our models are trained using cross-validation with 5 repeated splits of $50\% - 50\%$. We use 4 months as an observational period to collect activity logs $\ell(u_i, a_j)$. We selected the period of 4 months as a compromise: longer periods may result in more information, but our data pertains to a limited time window, and in addition we expect that relatively few fraudulent users escape detection for many months.

5.1 Metrics of success

We base the evaluation of the performance of the classifier on the standard notion of a *confusion matrix* (Table 2). We let

$$\text{TPR (True Positive Rate)} = TP/(TP+FN)$$

and

$$\text{FPR (False Positive Rate)} = FP/(FP+TN)$$

where TPR shows the ratio of correctly classified fraudulent users and FPR shows the ratio of misclassified normal users.

We visualize the tradeoff between TPR and FPR using the *receiver operating characteristic curve (ROC)* [2]. To quantify the overall ability of a model, we compute the *Area Under the Curve* (AUC) of the ROC. The AUC represents the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. Also, in addition to comparing models using their AUC, we fix the FPR to 5% and compare the TPR between the models. The models differ on the inputs used, so comparing them establishes the value of the inputs for the purpose of capturing the patterns of interest.

5.2 Results

Using our approach, we achieve a TPR of 68% with a FPR of 5%. This TPR is especially significant as the fraudulent users that we are detecting were able to overcome the first line of defenses in the existing, effective detection system. Similarly, the FPR of 5%, which may appear as high for a stand-alone system, may be reasonable in the context of other defenses. In addition, with our approach, the number of the fraudulent users that elude detection for more than 10 months since their account creation decreases by a factor of 2.3.

Figure 2 plots the distributions of the account lifetimes for fraudulent users before and after the application of our methods. Each account lifetime is calculated as the interval from the account creation to the final detection (closure) of the account by Skype. The x axis is in months. The figure labeled "Before" depicts the lifetime of fraudsters that escape Skype's current defenses. The figure labeled "After: missed" shows the lifetime of fraudsters that would have escaped detection with our approach, and the one labeled "After:detected" shows those fraudsters that would have been detected after the first 4 months of activity by our methods. The reduction in volume is apparent.

Note that our method is most effective in detecting fraudsters that would have 10 or less months of activity (after the initial 4 months), while missing most of those that stay active for over 30 months. Preliminary investigations indicate that an initial period of observation longer than 4 months would improve the detection of such long-term fraud. Perhaps a cascading set of classifiers, each with a different initial period of observation, would be helpful. We also conjecture that a large number of these fraudsters took over the accounts of normal users. Other techniques, such as change-point detection, may be fruitful in this context.

Figure 3 shows the overall performance of classifiers built using only one feature at a time (left-hand side, labeled "separate models") and classifiers built by adding one feature at a time (right-hand side, labeled "nested models"). For the nested models we begin from the simplest classifier that uses only the profile features. The next model combines profile features and product usage features. We continue to increase the complexity of the classifier by adding the feature sets one-by-one. In this study we use only a particular order for adding the feature sets. The order corresponds to the complexity and computational effort in pre-processing the features as discussed previously (see Section 3).

As depicted in the graph for nested models, the improvement in performance as we add features is monotonic, confirming that all the features contribute to detection.

Table 3 represents these results quantitatively. As can be observed, a model based on only the local social activity information has the best TPR, and the user profiles yield the best overall AUC score. Also, as mentioned above, the best model is one where all the features are used.

Finally, we report on the statistics per type of fraudulent user. As can be seen from Figure 4, we are able to detect most of the type II fraudsters, but we are less successful in detecting type III and type IV fraudsters. In further work, it may be attractive to investigate various feature sets for

Figure 2: Distribution of fraudulent users by their lifetime (of undetected activity) before using our approach and after eliminating those fraudsters caught by our approach

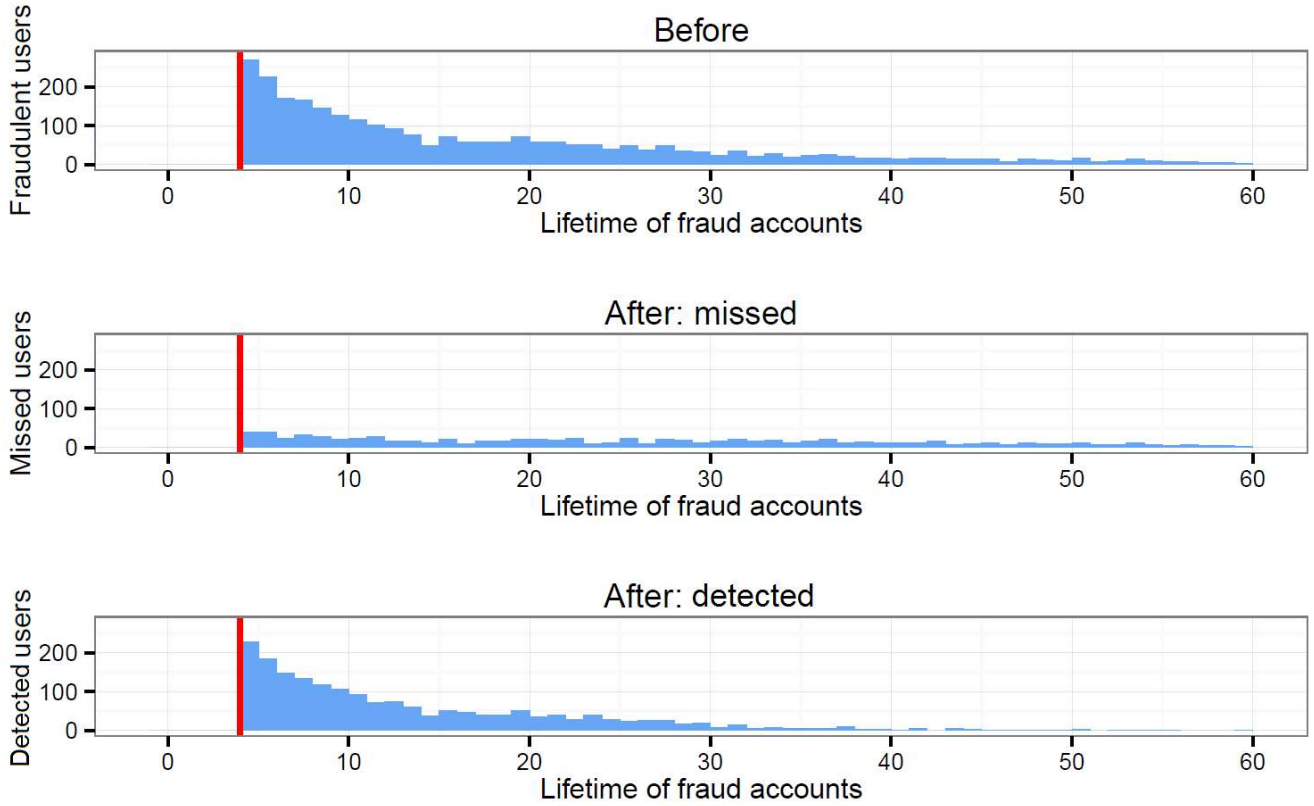


Table 3: Table of TPR under an FPR of 5% with corresponding confidence intervals, and AUC for different models when the features are used in isolation (*isolation*) and added one at a time (*nested*)

Features	**TPR (FPR = 0.05)**	**95% C.I.**	**AUC isolation**	**AUC nested**
Profile set	0.50	(0.48;0.52)	0.79	0.79
Skype product usage	0.25	(0.23;0.27)	0.65	0.84
Local social activity	0.54	(0.52;0.56)	0.74	0.86
Global social activity	0.37	(0.35;0.39)	0.68	0.87
All	0.68	(0.66;0.70)	0.87	

Figure 3: ROC curves for models built on one feature (left-hand side and labeled "separate models") and models built by adding one feature at a time (right-hand side and labeled "nested models")

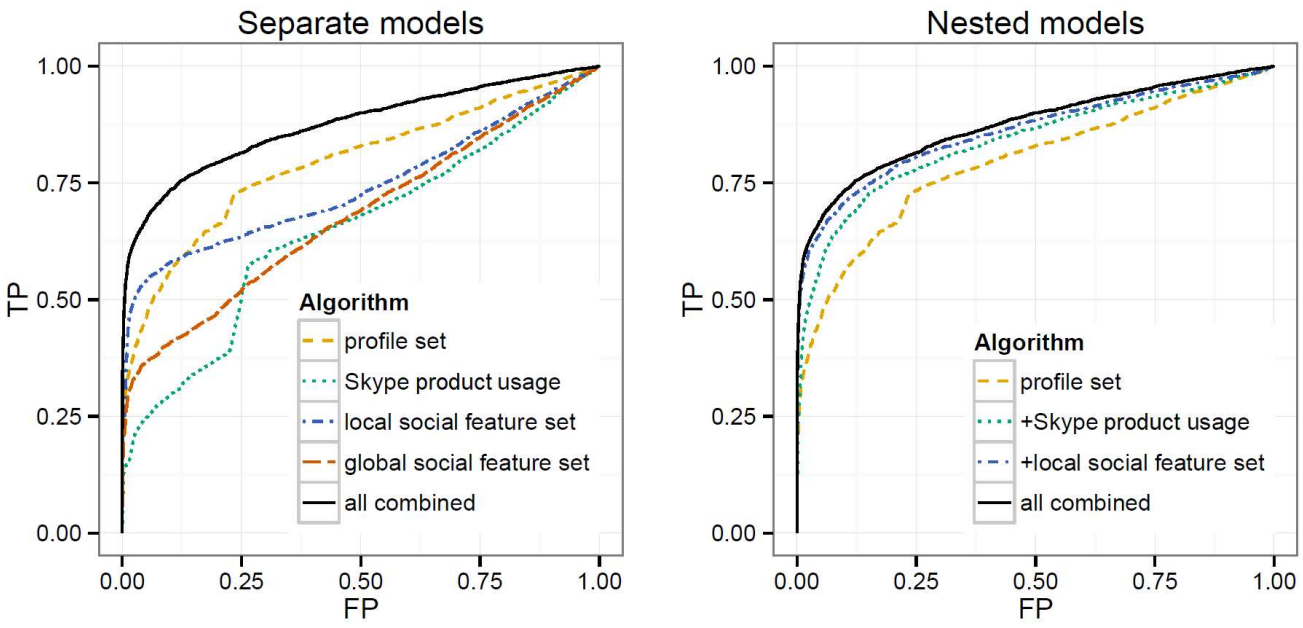
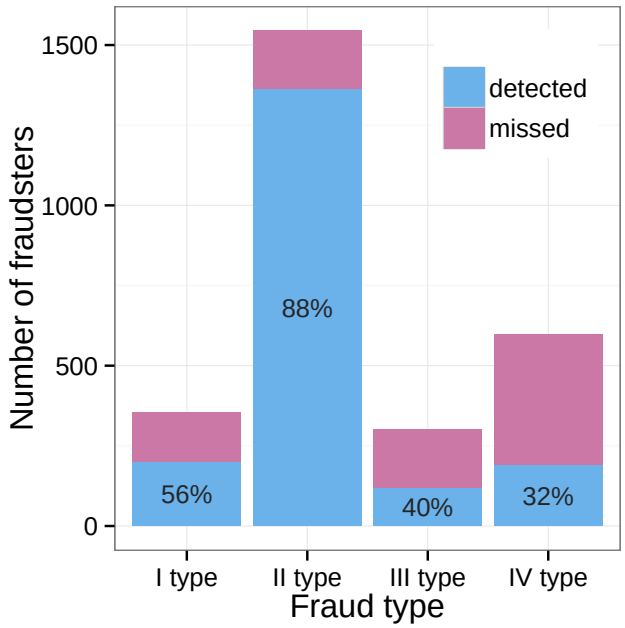


Figure 4: Distribution of fraudulent users by types and proportion of detected among them



different types of fraud, tailoring classifiers to the patterns of behavior that correspond to each type of fraud.

6. CONCLUSIONS

In this paper we present and evaluate an approach to detecting fraudulent users based on supervised machine learning. The approach combines information from diverse sources such as user profiles, user activities, and social connections. As our results demonstrate, fraud classification improves as we add each one of these sources (see Figure 3).

The concrete goal of our work was to detect stealthy fraudulent users. Specifically, we identified 68% of these users within the first 4 months of activity with a 5% false positive rate, and reduced the number of undetected fraudulent users active for over 10 months by a factor of 2.3. We consider that these quantitative results are encouraging and positive.

A central contribution of this paper is a set of methods for transforming raw data into features suitable for consumption by classifiers. In particular, we use HMMs in order to build models from the time-series data, thus producing inputs for our classifier. The applications of this approach go well beyond fraud detection (as suggested by work on failure prediction [12]). Classifiers are some of the most successful technologies to come out of machine learning, and time series are a natural way to represent data.

Our experiments also suggest several directions for further investigation. The different detection rates for various types of fraud indicate that each source of information may correlate differently with those types of fraud; hence, more elaborate ways of combining and cascading classifiers may lead to enhanced fraud detection for particular types of fraud. It should also be interesting to perform experiments with longer time series, attempting in particular to detect points in time at which users change behavior. Those changes in behavior sometimes result from account hijacking, a difficult, important problem that machine learning may help address.

7. ACKNOWLEDGMENTS

The authors gratefully acknowledge the help of Jeff McClelland, Ando Saabas, and Taavi Tamkivi from Skype; and the travel support for Anna from the Software Technology and Applications Competence Center (STACC).

8. REFERENCES

[1] 2013 Online Fraud Report, Online Payment Fraud Trends, Merchant Practices, and Benchmarks. Technical report, CyberSource Corporation, 2013.

[2] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, July 1997.

[3] L. Breiman. Random Forests. In *Machine Learning*, volume 45, pages 1–33, 2001.

[4] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *WWW*, 1998.

[5] J. Cao, M. Ahmadi, and M. Shridhar. Recognition of handwritten numerals with multiple feature and multistage classifier. *Pattern Recognition*, 28(2):153–160, Feb. 1995.

[6] P.-A. Chirita, J. Diederich, and W. Nejdl. Mailrank: using ranking for spam detection. In *CIKM*, pages 373–380, 2005.

[7] C. Cortes, D. Pregibon, and C. Volinsky. Communities of interest. In *Proceedings of the Fourth International Conference on Advances in Intelligent Data Analysis*, pages 105–114, 2001.

[8] J. D.Ratley. ACFE: Report to Members. *Association of Certified Fraud Examiners*, 2012.

[9] H. Farvaresh and M. M. Sepehri. A data mining framework for detecting subscription fraud in telecommunication. *Engineering Applications of Artificial Intelligence*, 24(1):182–194, Feb. 2011.

[10] T. Fawcett and F. Provost. Adaptive fraud detection. *Data mining and knowledge discovery*, 316:291–316, 1997.

[11] D. Fisher. Using egocentric networks to understand communication. *Internet Computing, IEEE*, (October):20–28, 2005.

[12] M. Goldszmidt. Finding soon-to-fail disks in a haystack. *Proceedings of the 4th USENIX conference on Hot Topics in Storage and File Systems*, 2012.

[13] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[14] C. S. Hilas. Designing an expert system for fraud detection in private telecommunications networks. *Expert Systems with Applications*, 36(9):11559–11569, Nov. 2009.

[15] C. S. Hilas and P. A. Mastorocostas. An application of supervised and unsupervised learning approaches to telecommunications fraud detection. *Knowledge-Based Systems*, 21(7):721–726, Oct. 2008.

[16] T. Ho, J. Hull, and S. Srihari. Decision combination in multiple classifier systems. *Pattern Analysis and Machine*, 16(I), 1994.

[17] J. Huang, Y. Xie, F. Yu, Q. Ke, M. Abadi, E. Gillum, and Z. M. Mao. Socialwatch: detection of online service abuse via large-scale social graphs. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, ASIA CCS '13, 2013.

[18] L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar. Adversarial machine learning. In *Proceedings of the Fourth ACM Workshop on Artificial Intelligence and Security (AISec)*. ACM, New York, NY, USA, 20011.

[19] J. Kittler and M. Hatef. On combining classifiers. *IEEE Transactions on*, 20(3):226–239, 1998.

[20] Y. Ku, Y. Chen, and C. Chiu. A Proposed Data Mining Approach for Internet Auction Fraud Detection. pages 238–243, 2007.

[21] M. McGlohon, S. Bay, M. G. Anderle, D. M. Steier, and C. Faloutsos. Snare: a link analytic system for graph labeling and risk detection. In *KDD*, pages 1265–1274, 2009.

[22] B. Nelson, B. I. P. Rubinstein, L. Huang, A. D. Joseph, and J. D. Tygar. Classifier evasion: Models and open problems. In C. Dimitrakakis, A. Gkoulalas-Divanis, A. Mitrokotsa, V. Verykios, and Y. Saygin, editors, *Privacy and Security Issues in Data Mining and Machine Learning*, volume 6549 of *Lecture Notes in Computer Science*, pages 92–98. Springer Berlin / Heidelberg, 2011.

[23] C. Phua, V. C. S. Lee, K. Smith-Miles, and R. W. Gayler. A comprehensive survey of data mining-based fraud detection research. *CoRR*, abs/1009.6119, 2010.

[24] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[25] S. Rosset, U. Murad, E. Neumann, Y. Idan, and G. Pinkas. Discovery of fraud rules for telecommunications-challenges and solutions. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 409–413. ACM Press, 1999.

[26] K. A. Roth. 2012 AFP Payments Fraud and Control Survey. Technical report, The Association for Financial Professionals, 2012.

[27] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 2003.

[28] J.-C. Wang and C.-C. Chiu. Recommending trusted online auction sellers using social network analysis. *Expert Systems with Applications*, 34(3):1666–1679, Apr. 2008.

[29] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–2, June 1998.