Thwarting Fake OSN Accounts by Predicting their Victims

Yazan Boshmaf
University of British Columbia
Vancouver, Canada

Matei Ripeanu
University of British Columbia
Vancouver, Canada

Konstantin Beznosov
University of British Columbia
Vancouver, Canada

ABSTRACT

Traditional defense mechanisms for fighting against automated fake accounts in online social networks are victim-agnostic. Even though victims of fake accounts play an important role in the viability of subsequent attacks, there is no work on utilizing this insight to improve the status quo. In this position paper, we take the first step and propose to incorporate predictions about victims of unknown fakes into the workflows of existing defense mechanisms. In particular, we investigated how such an integration could lead to more robust fake account defense mechanisms. We also used real-world datasets from Facebook and Tuenti to evaluate the feasibility of predicting victims of fake accounts using supervised machine learning.

1. INTRODUCTION

Users are valuable business asset for online social networks (OSNs) like Facebook,[1] Tuenti,[2] and RenRen.[3] For example, Facebook reports its quarterly earnings in terms of user metrics such as monthly active users (MAUs). In the first quarter of 2015 alone, Facebook reported $3.54 billion in revenue from its 1.44 billion MAUs [19]. However, for the same quarter, Facebook estimated that nearly 14 million of its MAUs are in fact "undesirable," representing malicious fake accounts that have been created in violation of the website's terms of service. For such OSNs, it is important that advertisers, developers, and investors trust their user metrics, as otherwise these stakeholders will be less willing to allocate budgets and resources to their platforms [15].

While OSN operators have strong incentives to identify and disable fake accounts, attackers have collateral incentives to create and control them. This is partly attributed to the fact that user accounts are "keys to walled gardens" [47], which means attackers have to either create fake accounts or compromise existing ones in order to exploit the target OSN. To date, this dependency has resulted in a growing underground market for creating, buying, and (re)selling fake accounts with millions of dollars in revenue [49].

From a security perspective, a *fake account* is an active user account that is created and controlled by an attacker for adversarial objectives, which include private data collection [7], social spamming [48], and political astroturfing [39]. To achieve these objectives, however, an attacker has to *infiltrate* the target OSN by connecting the fakes to many real accounts, as isolated fake accounts cannot freely interact with or promote content to most of the users in the OSN [5, 7, 35, 43]. We refer to real accounts that have accepted one or more connection or friend requests sent by malicious fake accounts as *victim accounts*.

The multifaceted threat posed by fake accounts has spawned a line of research with the goal of designing defense mechanisms to thwart their malicious activities (§2). To date, most approaches to identify fakes or control their admission rely on either static or dynamic features extracted from user activities, or on certain topological properties of the underlying social graph [13, 28, 43, 56]. While these techniques are effective against naïve attack strategies (e.g., malicious URL spamming using one fake account), many studies showed that they can be easily evaded in practice and often introduce serious usability issues [5, 8, 25, 36, 38].

This position paper advocates a new but complementary approach to fight against fake accounts in OSNs (§3). In particular, we propose to predict the victims of unknown fakes, and then incorporate these predictions into existing defense mechanisms. The effectiveness of this strategy stems from three main observations. First, as victim accounts are real accounts that are not controlled by attackers, identifying victims is inherently more robust against adversarial attacks than identifying fakes [4, 30, 50]. Second, since victim accounts constitute a small fraction of all real accounts [5, 7], restrictive admission control mechanisms can be applied to only these accounts and their connections, without limiting the experience of others. Third and last, as fakes are directly connected to victims in the social graph, fake account detection mechanisms can leverage identified victims to better delimit the subgraph that contains these fakes, possibly with stronger security guarantees [2, 9, 52, 59].

Our main insight herein is that infiltration campaigns run by fake accounts are similar to outbreaks of infectious diseases. In the early stages, as an attacker uses the information and credibility gained by fakes that have victims in the target OSN, the likelihood of mounting a successful attack increases super-linearly with number of victims. Our position is that defenders can thwart fake accounts both more efficiently and effectively by incorporating information about vulnerable accounts (i.e., potential victims) into existing OSN defenses. In the case of infectious diseases, this could be achieved through vaccination and education of the population at risk. In the case of OSNs, on the other hand, we consider four security mechanisms that are widely-used today, namely, topology-based fake account detection (§3.1), user-facing security advice (§3.2), user edu-

[1] http://facebook.com
[2] http://tuenti.com
[3] http://renren.com

AISec'15, October 16 2015, Denver, CO, USA
© 2015 ACM. ISBN 978-1-4503-3826-4/15/10 $15.00
DOI: http://dx.doi.org/10.1145/2808769.2808772.

cation (§3.3), and OSN honeypots (§3.4). This paper discusses how each mechanism can be improved by incorporating victim account prediction into its workflow.

Finally, to support the claim that one can indeed predict the victims of unknown fakes, more specifically, to predict the likelihood that a user will accept a connection request sent by a fake, we used two real-world dataset from Facebook and Tuenti and trained a victim account classifier using supervised machine learning (§4). The Facebook dataset contained public profiles of 8.8K real users who received friend requests from fake accounts, out of which 32% were victims. The Tuenti dataset contained full profiles of 60K real users who received friend requests from fakes, out of which 50% were victims. In our evaluation, we found that even using few statistically weak features, on OSN can train a victim account classifier with an area under receiver operating characteristics (ROC) curve (AUC) of 0.76, which is 52% better than random.

2. BACKGROUND

In what follows, we first introduce the threat model we assume in this paper (§2.1). After that, we present background on defending OSNs against fake accounts. In particular, we divide fake account defense mechanisms into two categories, based on how new accounts are admitted into the OSN. In the first category, new users are fully admitted without restrictions, but later their account activity is analyzed to detect fakes (§2.2). In the second category, accounts are provisioned and constrained before receiving full access to available services, which is typically based on criteria specific to individual users or the whole OSN (§2.3). We finally conclude and motivate our position with a brief summary (§2.4).

2.1 Threat Model

We focus on social networks that are open to everyone to join and allow users to explicitly form social connections (e.g., friendships). We consider attackers who are capable of creating and automating fake accounts on a large scale [7, 10]. Each fake account, or *socialbot*, can perform social activities similar to those of real users [24]. Such activities include sending friend requests and posting content.

The objectives of attackers include distributing social spam and malware, misinforming the public, and collecting private user data. To achieve these objectives, an attacker has to *infiltrate* the target OSN by using the fakes to befriend as many real accounts as possible. Such an infiltration is required because isolated fake accounts cannot directly interact with or promote content to other users [17]. This observation is also evident by a growing underground market for OSN abuse, where accounts with a large number of connections are sold at a premium. For example, fake Facebook accounts with 1,000 friends are sold for up to $26 [35].

We refer to accounts whose users have accepted friend requests from fake accounts as *victims*. We refer to friendships between victim and fake accounts as *attack edges*. The victims are a subset of *real accounts*, which are accounts created and controlled by benign users who socialize with others in a non-adversarial setting. Moreover, we refer to accounts whose users are more susceptible to social infiltration and are likely to be victims as *potential victims*. We use the terms "account," "profile," and "user" interchangeably but do make the distinction when deemed necessary.

2.2 Detecting Fake Accounts

Recently, fake account detection has received a considerable attention from academia and industry. In this defense category, all users are admitted to the OSN and granted full access to available services without restriction. From the OSN standpoint, giving users "the benefit of the doubt" has the advantage of providing a hassle-free account registration procedure, which might be helpful for attracting new users. The downside, however, is that attackers can have their fakes easily admitted to the OSN and start abusing its services. This is why OSNs, such as Facebook [43], Tuenti [13], and RenRen [58], allocate costly human and computer resources to identify fake accounts and disable them as soon as possible.

From a design perspective, fake account detection systems are either *feature-based* or *topology-based*, depending on whether machine learning or graph analysis techniques are applied. In the following subsections, we discuss each type of detection in detail.

2.2.1 Feature-based Fake Account Detection

Approaches of this type depend on user-level activity and profile details, which are usually logged and stored by the OSN for later use. By identifying discriminating *features* of an account, one can classify each account as fake or real using various machine learning techniques. For example, Facebook employs an "immune system" that performs real-time checks and classification for each read and write action on its database, which are based on features extracted from user accounts and their activities [43].

Yang et al. used ground-truth provided by RenRen to train an SVM classifier in order to detect fake accounts [58]. Using simple features, such as frequency of friend requests and fraction of accepted requests, the authors were able to train a classifier with 99% true-positive rate (TPR) and 0.7% false-positive rate (FPR).

Stringhini et al. utilized honeypot accounts in order to collect data describing a wide range of user activities in OSNs [45]. After analyzing the collected data, they were able to assemble a ground-truth for real and fake accounts, with features similar to those used by Yang et al. [58]. The authors trained two random forests (RF) classifiers to identify fakes in Facebook and Twitter, ending up with 2% FPR and 1% false-negative rate (FNR) for Facebook, and 2.5% FPR and 3% FNR for Twitter.

Wang et al. used a clickstream dataset provided by RenRen to cluster user accounts into "similar" behavioral groups, corresponding to real or fake accounts [54]. The authors extracted both session and click features, including average clicks per session, average session length, the percentage of clicks used to send friend requests, visit photos, and to share content. With these features, the authors were able to train a cluster-based classifier using the METIS clustering algorithm [27] with 3% FPR and 1% FNR.

Cao et al. observed that fake accounts tend to perform loosely synchronized actions in a variety of OSN applications, all from a limited set of IP addresses [14]. The authors extracted simple user action features, such as timestamp, target application, and IP address, in order to cluster user accounts according to the similarity of their actions using a scalable implementation of the single-linkage hierarchical clustering algorithm. Through a large-scale deployment at Facebook, the authors we able to train a cluster-based classifier and detect more than two million fake accounts, which acted similarly at about the same time for a sustained period of time.

The downside of feature-based detection. Even though feature-based detection scales to large OSNs, it is still relatively easy to subvert as it depends on features describing activities of known fakes in order to identify unknown ones. In other words, attackers can evade detection by adversely modifying the content and activity patterns of their fakes, resulting in an arms race [30, 50]. While OSN operators employ skilled human analysts to improve their ground-truth and keep their fake account classifiers accurate, attackers react to detecting their fakes by modifying them and their behaviours in order to evade detection. Additionally, feature-based detection does not provide any formal security guarantees and usually results in a high FPR when deployed in practice [13]. This is partially attributed to the large variety and unpredictability of user behavior in adversarial settings [2, 13].

Our approach. In our proposed approach, feature-based detection is used to identify potential victims in a *non-adversarial setting*. Key to the robustness of this approach is that the dataset used for training the victim classifier includes features of only known real accounts that have accepted or rejected friend requests sent by known fakes. As real accounts are controlled by benign users who are not adversarial, a feature-based victim classifier is harder to circumvent than a similarly-trained fake account classifier.

2.2.2 Topology-based Fake Account Detection

In an attempt to address the lack of formal security guarantees in feature-based detection, topology-based detection approaches model OSNs as *graphs*, with nodes representing user accounts and edges between nodes representing social connections. Assuming that fakes can establish only a small number of attack edges, the subgraph induced by the set of real accounts is loosely connected to fakes. In other words, topology-based approaches rely on a key assumption that the *cut* crossing over attack edges is sparse, and accordingly, they aim to find such a sparse cut with formal guarantees [2, 52, 59]. For example, Tuenti employs SybilRank to rank accounts according to their perceived likelihood of being fake, where the ranking scheme is based on Tuenti's graph topological properties [13].

Yu et al. were among the first to use the social graph for the purpose of identifying fake accounts in computer systems [60, 61]. The authors developed a technique that labels each account as either fake or real based on a number of modified random walks. This *binary classification* was used to split the graph into two subgraphs that are sparsely interconnected via attack edges, separating real accounts from fakes. They also proved that $O(|E_a|\log n)$ fakes can be misclassified in the worst case, where $|E_a|$ is the number of attack edges and n is the number of accounts in the network. Therefore, it is sufficient for attackers to establish $\Omega(n/\log n)$ attack edges in order to evade this detection scheme with 0% TPR.

Viswanath et al. used existing community detection techniques to identify fake accounts in OSNs [51]. In general, community detection decomposes a given graph into a number of tightly-knit subgraphs, each called *community*, that are loosely connected to each other [21, 29]. By gradually expanding a community from known real accounts [33], the authors were able to identify the subgraph which contains mostly real accounts. Recently, however, Alvisi et al. showed that such a local community detection technique can be easily circumvented, if fake accounts establish sparse connectivity among themselves [2].

As binary classification often results in a high FPR [51], Cao et al. proposed to rank users instead so that fakes are ideally ranked lower than real accounts [13]. The authors designed and deployed a fake account detection system (based on a modified random walk) that assigns each account a rank depending on how likely it is to be fake. They also proved that $O(|E_a|\log n)$ fake accounts can outrank real accounts in the worst case, given that fakes establish $|E_a|$ attack edges with victims at random.

The downside of topology-based detection. While topology-based detection provides formal security guarantees, real-world social graphs do not conform with its key assumption. Specifically, several studies showed that attackers can infiltrate OSNs by deceiving users into befriending their fakes [5, 7, 18, 53]. This also means that fakes can create many attack edges such that they become densely connected to real accounts, which renders topology-based fake account detection ineffective in practice [9].

Our approach. In our approach, victim prediction can be leveraged to artificially prune attack edges so that the cut between fake and real accounts becomes sparse and easier to detect. As discussed in §3.1, one way to achieve this is by reassigning edge weights in the graph, such that edges incident to potential victims have lower weights than others. This means that one can identify attack edges and bound the security guarantee by the aggregate weight on attack edges, called their *volume* $vol(E_a)$, rather than their number $|E_a|$.

2.3 Controlling Account Admission

As presented in §2.2, fake account detection represents a *reactive* defense strategy, in which new accounts are first admitted to the OSN and then classified. On the other hand, admission control represents a *proactive* defense strategy, in which accounts are provisioned and controlled before being fully admitted to the OSN or given full access to its services.

Xie et al. [56] proposed an admission process, where already admitted accounts implicitly *vouch* for newly created accounts by, for example, sending them personal messages. By carefully monitoring implicit vouching via social community structures, they were able to admit 85% of real accounts while reducing the percentage of admitted fake accounts from 44.4% to 2.4%, using Microsoft Hotmail and Twitter datasets.

Mislove et al. were among the first to dynamically limit available OSN services provided to unknown fakes by modelling lateral trust relationships among users as a *credit network* [32]. The authors developed a technique that assigns credit values to friendships, such that an account is able to send a friend request only if there is a path with available credit from the sender to the receiver. Mondal et al. utilized this approach to limit large-scale crawls in OSNs [34].

Kim et al. proposed to visualize the trust between users in order to help them better authenticate those who request their friendship in OSNs [28]. Inspired by social science research demonstrating that the *social tie strength* is a strong indicator of trust, the authors developed a tool to visualize the social tie strength between the receiver and the sender of a friend request, based on features of their mutual friends, such as their interaction frequency, communication reciprocity, recency, and length. To evaluate their approach, the authors conducted a survey with 93 participants who used their visualization tool. The participants found that the tool helped them make better befriending decisions, especially when they received friends requests from fake accounts posing as strangers.

Wang et al. utilized known concepts from behavioral decision research and soft paternalism in order to design mechanisms that "nudge" users to reconsider the content and context of their online disclosures before committing them [55]. The authors evaluated the effectiveness of their approach with 21 Facebook users in a three week exploratory field study, followed up with 13 interviews. The results suggest that *privacy nudges* can be a promising way to prevent unintended disclosures when, for example, one befriends a fake account or shares a post with the public.

The downside of account admission control. Even though this type of approaches may lead to more resilient defense mechanisms, it is based on the view that "users are guilty until proven innocent." If acted upon, this can significantly degrade user experience, introduce friction, and limit network growth [3, 42]. It is not surprising that large, fast-growing OSNs, such as Facebook, avoid deploying stringent admission control and instead advocate user feedback after admission [43].

Our approach. In §3.2, we argue that user-controlled account admission techniques, especially those facilitated by a user-facing security advice, can be improved by identifying potential victims. One way to achieve this is by focusing available resources on helping potential victims make better befriending decisions through education or personalized advice; approaches that would be too costly or degrade user experience if they were applied indiscriminately to the entire user population.

2.4 Summary

Defending against fake accounts in OSNs can be done reactively by admitting all accounts and then classifying them, or proactively by applying various forms of admission control. Approaches based on reactive detection rely on distinct patterns of fakes in order to identify them. These patterns capture account activities at both the content and structure levels, but are subject to the complete control by attackers. As a result, this leads to an arms race, where the party with more resources eventually wins. Proactive admission control, while promising, degrades the experience of newly registered legitimate users, and goes against the business model of today's OSNs which compete on the number of active users.

Overall, all existing approaches to thwarting automated fake accounts in OSNs are victim-agnostic and attacker-centric. Even though victims of fakes play an important role in the viability of subsequent attacks, there is no work on using this insight to improve the status quo. We herein take the first step towards bridging this gap.

3. LEVERAGING VICTIM PREDICTION

We now consider four defense mechanisms and investigate how incorporating predictions of victims can improve their performance.

3.1 Topology-based Fake Account Detection

As victims are located at the borderline between two sub-graphs separating fakes from real accounts, one could limit the weight of the edges crossing this boundary by incorporating victim prediction into the detection process. We propose to achieve this improvement by assigning lower weights to edges incident to potential victims. Now, consider a short random walk that starts from a known, real account that is not a victim. It is unlikely for this walk to traverse low-weight edges and land on a node representing a fake account, as the walk picks the next node with a probability proportional to the weight of the corresponding edge. By ranking accounts based on their landing probability, one expects most of the fakes to have a strictly lower rank than real accounts, even if the fakes befriend many victim accounts.

We have recently validated this approach with Íntegro [11], which is a scalable, infiltration-resilient defense system that helps OSNs detect automated fake accounts using a user ranking scheme. The system delivers $O(|E_a|/vol(E_a))$ improvement over SybilRank [13], the state-of-the-art in topology-based fake account detection, and is currently deployed at Tuenti with up to an order of magnitude better precision in fake account detection.

3.2 User-Facing Security Advice

User advice represents the first line of defense against increasingly sophisticated social engineering attacks [26, 44]. While many studies showed that users tend to reject or ignore security advice because of low motivation and poor understanding of the involved threats [1, 31], others asserted that users do so because it is entirely rational from an economic standpoint [20, 23], as a randomly chosen user has a low chance to become a victim of a particular attack.

A security advice aims to protect users from the direct costs of an attack, but burdens them with increased indirect costs in the form of additional effort (e.g., additional steps to accept friend requests). When the security advice is applied to all users, it becomes a daily burden equally taken by the entire OSN population, whose benefit is the potential saving of direct costs for the actual victims of attacks. When this fraction is small, designing a good security advice that works well in practice becomes difficult. For example, it is not feasible to burden the 1.44 billion Facebook users with a daily task in order to spare, say, 1% of them from becoming victims.

One way to increase the benefit of a security advice is to improve its usability, which in effect reduces its indirect costs to users. This has been the focus of a growing community of usable security researchers who consider user education essential to securing socio-technical systems such as OSNs [16].

Another, complementary, way to reduce indirect costs is to focus the security advice on the fraction of users that would directly benefit from it; the potential victims. We propose to achieve this reduction by providing the security advice in an informed and targeted manner. In particular, victim prediction provides OSNs with a robust method to quantitatively estimate how vulnerable each real account is, as some users are more likely to be victims than others. So, an OSN can use this information to focus only on the most vulnerable user population and, accordingly, influence their decision making through a personalized security advice while relieving the rest of the population from the indirect costs.

3.3 User Education

Similar to targeted advertising, potential victims can be the target of user education campaigns. Compared to a security advice, user education is more generic as it does not assume a specific context, and is generally focused towards increasing risk awareness instead of informing context-specific decision making.

We hypothesize that a highly efficient education method is to operate a network of benign socialbots to send friend requests to potential victims. In case a user accepts such requests, the bots would educate the user about the involved risks (e.g., private data breaches), and provide pointers for better decision making (e.g., to opt-in for a personalized security advice).

3.4 Honeypots and Activity Sampling

Sampling is used to maintain an up-to-date ground-truth for fake and benign account characterization. One way to sample an OSN is to use *honeypots*, which are user accounts that are created to log activities of other users, in particular, those who contact honeypots by sending them friend requests or by sharing content [45]. While honeypots are often used by third parties, OSNs perform a similar sampling albeit with direct access to user data and activities [43]. If the goal is to collect fake activity, such a sampling technique is often inefficient, as it is opportunistic if not completely random. For example, Stringhini et al. used 300 honeypot accounts in Facebook to record user activities over 12 months [45]. The collected dataset, however, was small relative to the sampling period, with only 3,831 friend requests (4.5% fake) and 72,431 messages (5.4% fake).

Assuming that the percentage of fakes in the OSN is small, sampling users at random will collect mostly benign content originating from real accounts. The problem, however, is when one samples for *abusive content*, as the sampling has to be biased towards unknown fake accounts. For example, Facebook has more than 600 million daily active users and they perform billions of actions per day [37]. In contrast, the number of fakes involved in an attack is often on the order of thousands [14]. Without a reliable way to inform the sampling method, an OSN will be looking for a needle in a haystack.

As victims are directly connected to abusive fakes, we propose to identifying potential victims of fakes and then sample the activities of their friends. Along with typical uniform random sampling [40], an OSN can now achieve a desirable benign-to-abusive content ratio (i.e., class distribution), which, depending on the used algorithms, is important for effective feature-based abuse detection using machine learning techniques [22].

4. IDENTIFYING POTENTIAL VICTIMS

The feasibility of the mechanisms discussed in §3 depends on the assumption that an OSN can accurately predict the victims of fake accounts, that is, the likelihood an account will accept a connection request sent by a fake.

We next describe how we used supervised machine learning and real-world datasets to validate this assumption. We employed only low-cost features extracted from readily-available user profiles, and thus, our experiments provide what is likely a lower estimate for the victim classification performance. We believe that it is possible to achieve better classification performance, at relatively higher cost, by using richer profile features and advanced learning algorithms that incorporate temporal activity [22].

4.1 Robustness

What differentiates a feature-based victim classifier from a similarly-trained fake account classifier—other than classification labels—is that it is relatively harder to circumvent [4]. This robustness stems from the fact that the features used to train the victim classifier are extracted from known real accounts *that are not controlled by attackers*, which is not the case when classifying fake accounts.

For example, in the "boiling-frog" attack [41, 50], fake accounts can force a classifier to tolerate abusive activities by slowly introducing similar activities to the OSN. Because the OSN operator has to retrain all deployed classifiers in order to capture new behaviors, a fake account classifier will learn to tolerate more and more abusive activities, up until the attacker can launch a full-scale attack without detection [7]. When identifying potential victims, however, this is only possible if the real accounts used to train the victim classifier have been compromised. This situation can be avoided by verifying the accounts, as described in §4.2.3.

4.2 Datasets

We used real-world datasets of different OSNs. The first dataset was collected from Facebook as part of a 2011 study [7], and contained public user profiles. As for the second dataset, we collaborated with Tuenti to access a day's worth of aggregated, anonymized, and server-cached user profiles, with the whole process being mediated by Tuenti's Site Integrity team.

4.2.1 Facebook

The dataset contained public profiles of 9,646 real users who received friend requests from fake accounts. Since the dataset was collected in 2011, we wanted to verify whether these users are still active on Facebook. Accordingly, we revisited their public profiles in June 2013 . We found that 7.9% of these accounts were disabled by Facebook or deactivated by the users themselves. We thus excluded these accounts, ending up with 8,888 accounts, out of which 32.4% were victims who accepted a single friend request sent by an automated fake account posing as a stranger. A comprehensive description and analysis of this dataset can be found in [6].

4.2.2 Tuenti

The dataset contained full user profiles of 60,000 real accounts who received friend requests from fake accounts, out of which 50% were victims. The dataset was collected on Feb 10, 2014 by Tuenti from live production servers, where the data resided in memory. From a practical perspective, collecting the dataset was a relatively low-cost process, because it involved reading cached profiles of users who logged in to Tuenti on that particular day (i.e., the daily active users). Accordingly, there was no need for more expensive queries to the backend infrastructure.

4.2.3 Ground-truth

For the Facebook dataset, we started with the ground-truth from the original study, and then re-validated it in mid 2013, as described above. For the Tuenti dataset, all accounts were manually inspected and labeled by its account analysts. The inspection included matching profile photos to the user's declared age and address, understanding natural language used in user posts and messages, examining the user's friends, and analyzing related IP address and HTTP information, including requests and cookies.

4.3 Features

As summarized in Table 1, we extracted features from both datasets to generate feature vectors. The selection requirement was to have each feature value available for all users in the dataset, so that the resulting feature vectors are complete. For the Facebook dataset, we were able to extract 18 features from public user profiles. For the Tuenti dataset, we were able to extract only 14 features.

In Table 1, the RI score of a particular feature stands for its *relative importance* when compared to all other features. An "N/A" means the feature was not available for the corresponding dataset. A *k-categorical* feature means the feature can have one value out of *k* unique categories. For example, boolean features are 2-categorical.

4.4 Classifier Tuning

We used random forests (RF) to train a victim classifier. The RF algorithm is an *ensemble* algorithm, where a set of decision trees are constructed at training time. When evaluating the classifier on new data (i.e., unlabeled feature vectors), the decisions from all trees are combined using a majority voting aggregator [12]. Each decision tree in the forest uses a small random subset of available features in order to decrease the *generalization error*, which measures how well the trained classifier generalizes to unseen data [22]. As shown in Figure 1, we performed parameter tuning to calibrate the RF classifier. In particular, we used the out-of-bag error estimates freely computed by the RF algorithm to numerically find the best number of decision trees and the number of features for each tree, so that the prediction variance and bias are controlled across the trees. For the Facebook dataset, we used 450 decision trees, where each tree had 3 features picked at random out of 18 features. For the Tuenti dataset, we used 500 decision trees, where each tree had 7 features picked at random out of 14 features.

4.5 Validation Method

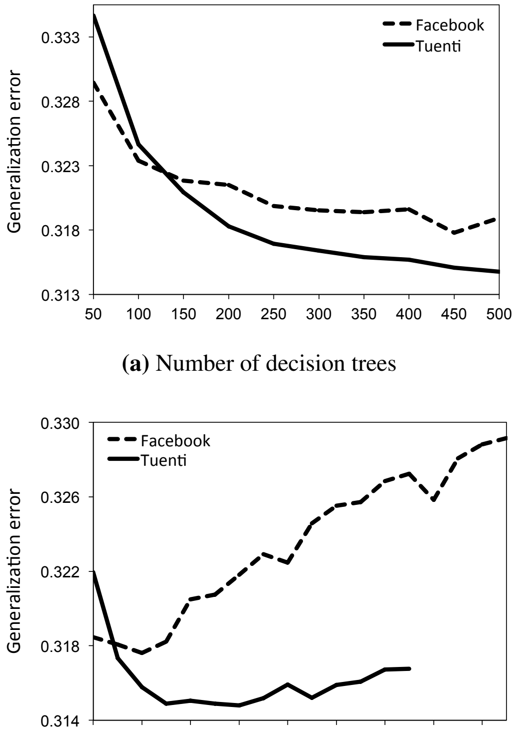
To evaluate the accuracy of the victim classifier, we performed a 10-fold, *stratified cross-validation* method [22] using the RF learning algorithm after parameter tuning. First, we randomly partitioned the dataset into 10 equally-sized sets, with each set having the same percentage of victims as the complete dataset. We next trained an RF classifier using 9 sets and tested it using the remaining set. We repeated this procedure 10 times (i.e., folds), with each of the sets being used once for testing. Finally, we combined the results of the folds by computing the mean of their true-positive rate (TPR) and false-positive rate (FPR).

4.6 Performance Metrics

The output of a victim classifier depends on its *operating threshold*, which is a cutoff value in the prediction probability after which the classifier identifies a user as a potential victim. In order to capture the trade-off between TPR and FPR in single curve, we repeated the cross-validation method under different threshold values using a procedure known as *receiver operating characteristics* (ROC) analysis. In ROC analysis, the closer the curve is to the top-left corner at point $(0,1)$ the better the classification performance

Feature	Brief description	Type	RI Score (%)	
			Facebook	Tuenti
User activity:				
Friends	Number of friends the user had	Numeric	100.0	84.5
Photos	Number of photos the user shared	Numeric	93.7	57.4
Feed	Number of news feed items the user had	Numeric	70.6	60.8
Groups	Number of groups the user was member of	Numeric	41.8	N/A
Likes	Number of likes the users made	Numeric	30.6	N/A
Games	Number of games the user played	Numeric	20.1	N/A
Movies	Number of movies the user watched	Numeric	16.2	N/A
Music	Number of albums or songs the user listened to	Numeric	15.5	N/A
TV	Number of TV shows the user watched	Numeric	14.2	N/A
Books	Number of books the user read	Numeric	7.5	N/A
Personal messaging:				
Sent	Number of messages sent by the user	Numeric	N/A	53.3
Inbox	Number of messages in the user's inbox	Numeric	N/A	52.9
Privacy	Privacy level for receiving messages	5-categorical	N/A	9.6
Blocking actions:				
Users	Number of users blocked by the user	Numeric	N/A	23.9
Graphics	Number of graphics (photos) blocked by the user	Numeric	N/A	19.7
Account information:				
Last updated	Number of days since the user updated the profile	Numeric	90.77	32.5
Highlights	Number of years highlighted in the user's time-line	Numeric	36.3	N/A
Membership	Number of days since the user joined the OSN	Numeric	31.7	100
Gender	User is male or female	2-categorical	13.8	7.9
Cover picture	User has a cover picture	2-categorical	10.5	< 0.1
Profile picture	User has a profile picture	2-categorical	4.3	< 0.1
Pre-highlights	Number of years highlighted before 2004	Numeric	3.9	N/A
Platform	User disabled third-party API integration	2-categorical	1.6	< 0.1

Table 1: Low-cost features extracted from Facebook and Tuenti.



(a) Number of decision trees

(b) Number of features in each tree

Figure 1: Victim classifier tuning using a grid search.

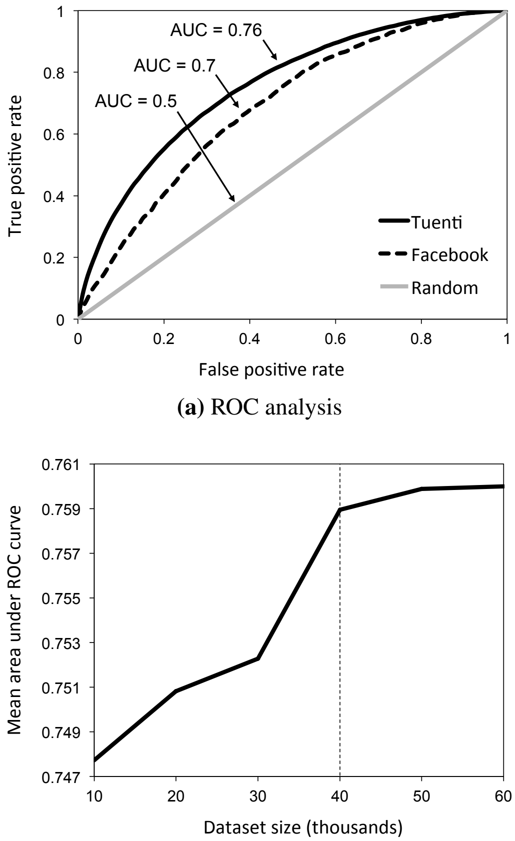
is. The quality of the classifier can be quantified with a single value by calculating the *area under its ROC curve* (AUC) [22]. An AUC of 1 means a perfect classifier, while an AUC of 0.5 means a random classifier. The victim classifier has to be better than random, i.e., with the AUC > 0.5.

We also recorded the *relative importance* (RI) of features used for the classification. The RI score is computed by the RF learning algorithm, and it describes the relative contribution of each feature to the predictability of the label—being a victim or a non-victim—when compared to all other features [12].

4.7 Results

For both datasets, the victim classifier resulted in an AUC greater than 0.5, as depicted in Figure 2a. In particular, for the Facebook dataset, the classifier delivered an AUC of 0.7, which is 40% better than random. For the Tuenti dataset, on the other hand, the classifier delivered an AUC of 0.76, which is 52% better than random. Also, increasing the dataset size to more than 40K feature vectors did not significantly improve the AUC in cross-validation, as shown in Figure 2b. This means an OSN can train a victim classifier using a relatively small dataset and fewer accounts need to be manually verified in order to maintain a low-noise ground truth.

We also experimented with two other widely-used learning algorithms: Naïve Bayes (NB) and SVM [22]. Both of these algorithms resulted in lower AUCs on both datasets. In particular, on the Facebook dataset, the NB classifier achieved an AUC of 0.63 and the SVM classifier achieved an AUC of 0.57. Similarly, on the Tuenti dataset, the NB classifier achieved an AUC of 0.64 and the SVM classifier achieved an AUC of 0.59. This, however, is not surprising, as ensemble learning algorithms, such as RF, achieve better predictive performance in case individual classifiers are "statisti-



(a) ROC analysis

(b) Sensitivity to dataset size

Figure 2: Victim classification using the RF algorithm

cally weak," meaning they have low AUCs but are still better than random [22]. We note that an in-depth analysis and a comprehensive sensitivity analysis is typically required if one aims to improve the AUC of a classifier.

5. RELATED WORK

While we are the first to leverage potential victims to thwart fake accounts, others have analyzed victim accounts in order to understand the larger cyber criminal ecosystem for OSN abuse [46].

Wagner et al. developed predictive models to identify users who are more susceptible to social infiltration in Twitter [53]. They found that *susceptible users*, or potential victims, tend to use Twitter for conversational purposes, are more open and social since they communicate with many different users, use more socially welcoming words, and show higher affection than non-susceptible users.

Yang el al. studied the cyber criminal ecosystem on Twitter [57]. They found that victims fall into one of three categories. The first are *social butterflies* who have large numbers of followers and followings, and establish social relationships with other accounts without careful examination. The second are *social promoters* who have large following-follower ratios, larger following numbers, and a relatively high URL ratios in their tweets. These victims use Twitter to promote themselves or their business by actively following other accounts without consideration. The last are *dummies* who post few tweets but have many followers. In fact, these victims are dormant fake accounts at an early stage of their abuse.

6. CONCLUSION

"There can be no evil without good." — Augustine

In this position paper, we proposed a new approach to thwart fake accounts in OSNs. The approach incorporates prediction of victims of unknown fakes into the existing defense mechanisms. We discussed how this approach could improve the status quo, focusing on known defense mechanisms such as topology-based fake account detection and user-facing security advice. Finally, we proposed and evaluated a technique for predicting victims of fake accounts using supervised machine learning. In particular, we showed that one can train a victim account classifier that is 52% better than random, using strictly low-cost features.

7. ACKNOWLEDGEMENTS

We would like to thank Dionysios Logothetis (Facebook), Eduardo Martinez-Esparzaand (Tuenti), Elizeu Santos-Neto (Google), Georgos Siganos (QCRI), and members of NetSysLab and LERSSE for their invaluable feedback on an early version of this paper. The first author would like to thank the University of British Columbia for a generous doctoral fellowship.

References

[1] A. Adams and M. A. Sasse. Users are not the enemy. *Communications of the ACM*, 42(12):40–46, 1999.

[2] L. Alvisi, A. Clement, A. Epasto, U. Sapienza, S. Lattanzi, and A. Panconesi. SoK: The evolution of sybil defense via social networks. *In Proceedings of the IEEE Symposium on Security and Privacy*, 2013.

[3] D. Balfanz, G. Durfee, D. K. Smetters, and R. Grinter. In search of usable security: Five lessons from the field. *IEEE Security & Privacy*, 2004.

[4] M. Barreno, B. Nelson, A. D. Joseph, and J. Tygar. The security of machine learning. *Machine Learning*, 81(2): 121–148, 2010.

[5] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international Conference on World Wide Web*, pages 551–560. ACM, 2009.

[6] Y. Boshmaf. *Security analysis of malicious socialbots on the web*. PhD thesis, University of British Columbia, 2015.

[7] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. The socialbot network: when bots socialize for fame and money. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 93–102. ACM, 2011.

[8] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. Key challenges in defending against malicious socialbots. In *Proceedings of the 5th USENIX Workshop on Large-scale Exploits and Emergent Threats*, volume 12, 2012.

[9] Y. Boshmaf, K. Beznosov, and M. Ripeanu. Graph-based sybil detection in social and information systems. In *Proceedings of 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 2013.

[10] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. Design and analysis of a social botnet. *Computer Networks*, 57(2):556–578, 2013.

[11] Y. Boshmaf, D. Logothetis, G. Siganos, J. Lería, J. Lorenzo, M. Ripeanu, and K. Beznosov. Íntegro: Leveraging victim prediction for robust fake account detection in OSNs. In *In proceedings of ISOC Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2015.

[12] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[13] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *USENIX conference on Networked Systems Design and Implementation*, pages 15–15. USENIX Association, 2012.

[14] Q. Cao, X. Yang, J. Yu, and C. Palow. Uncovering large groups of active malicious accounts in online social networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS'14, pages 477–488. ACM, 2014.

[15] CBC. Facebook shares drop on news of fake accounts, Aug 2012. URL http://goo.gl/6s5FKL.

[16] L. F. Cranor. *Security and usability: designing secure systems that people can use*. O'Reilly Media, Inc., 2005.

[17] N. B. Ellison et al. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, 2007.

[18] A. Elyashar, M. Fire, D. Kagan, and Y. Elovici. Homing socialbots: intrusion on a specific organization's employee using socialbots. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 1358–1365. ACM, 2013.

[19] Facebook. Quarterly earning reports, May 2015. URL http://goo.gl/YujtO.

[20] D. Florêncio and C. Herley. Where do security policies come from? In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, page 10. ACM, 2010.

[21] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[22] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: Data mining, inference, and prediction, second edition*. Springer, 2009.

[23] C. Herley. So long, and no thanks for the externalities: the rational rejection of security advice by users. In *Proceedings of the 2009 workshop on New security paradigms workshop*, pages 133–144. ACM, 2009.

[24] T. Hwang, I. Pearce, and M. Nanis. Socialbots: Voices from the fronts. *interactions*, 19(2):38–45, 2012.

[25] D. Irani, M. Balduzzi, D. Balzarotti, E. Kirda, and C. Pu. Reverse social engineering attacks in online social networks. In *Detection of intrusions and malware, and vulnerability assessment*, pages 55–74. Springer, 2011.

[26] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Communications of the ACM*, 50(10): 94–100, 2007.

[27] G. Karypis and V. Kumar. Multilevel *k*-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed computing*, 48(1):96–129, 1998.

[28] T. H.-J. Kim, A. Yamada, V. Gligor, J. Hong, and A. Perrig. Relationgram: Tie-strength visualization for user-controlled online identity authentication. In *In Proceedings of Financial Cryptography and Data Security Conference*, pages 69–77. Springer, 2013.

[29] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[30] D. Lowd and C. Meek. Adversarial learning. In *Proceedings of the 11th ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647. ACM, 2005.

[31] M. Mannan and P. C. van Oorschot. Security and usability: the gap in real-world online banking. In *Proceedings of the 2007 Workshop on New Security Paradigms*, pages 1–14. ACM, 2008.

[32] A. Mislove, A. Post, P. Druschel, and P. K. Gummadi. Ostra: Leveraging trust to thwart unwanted communication. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, pages 15–30. USENIX Association, 2008.

[33] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You are who you know: inferring user profiles in online social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 251–260. ACM, 2010.

[34] M. Mondal, B. Viswanath, A. Clement, P. Druschel, K. P. Gummadi, A. Mislove, and A. Post. Limiting large-scale crawls of social networking sites. *ACM SIGCOMM Computer Communication Review*, 41(4):398–399, 2011.

[35] M. Motoyama, D. McCoy, K. Levchenko, S. Savage, and G. M. Voelker. Dirty jobs: The role of freelance labor in web service abuse. In *Proceedings of the 20th USENIX Security Symposium*, pages 14–14. USENIX Association, 2011.

[36] F. Nagle and L. Singh. Can friends be trusted? exploring privacy in online social networks. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Network Analysis and Mining*, pages 312–315. IEEE, 2009.

[37] R. Nishtala, H. Fugal, S. Grimm, M. Kwiatkowski, H. Lee, H. C. Li, R. McElroy, M. Paleczny, D. Peek, P. Saab, D. Stafford, T. Tung, and V. Venkataramani. Scaling memcache at Facebook. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, NSDI'13, pages 385–398. USENIX Association, 2013.

[38] R. Potharaju, B. Carbunar, and C. Nita-Rotaru. iFriendU: Leveraging 3-cliques to enhance infiltration attacks in online social networks. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 723–725. ACM, 2010.

[39] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, S. Patil, A. Flammini, and F. Menczer. Truthy: mapping the spread of astroturf in microblog streams. In *Proceedings of the 20th International Conference Companion on World Wide Web*, pages 249–252. ACM, 2011.

[40] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, 2005.

[41] B. I. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.-h. Lau, S. Rao, N. Taft, and J. Tygar. Antidote: understanding and defending against poisoning of anomaly detectors. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 1–14. ACM, 2009.

[42] D. K. Smetters and R. E. Grinter. Moving from the design of usable security technologies to the design of useful secure applications. In *Proceedings of the 2002 workshop on New security paradigms*, pages 82–89. ACM, 2002.

[43] T. Stein, E. Chen, and K. Mangla. Facebook immune system. In *Proceedings of the 4th Workshop on Social Network Systems*, pages 8–14. ACM, 2011.

[44] K. Strater and H. R. Lipford. Strategies and struggles with privacy in an online social networking community. In *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction-Volume 1*, pages 111–119. British Computer Society, 2008.

[45] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 1–9. ACM, 2010.

[46] G. Stringhini, G. Wang, M. Egele, C. Kruegel, G. Vigna, H. Zheng, and B. Y. Zhao. Follow the green: growth and dynamics in twitter follower markets. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 163–176. ACM, 2013.

[47] S.-T. Sun, Y. Boshmaf, K. Hawkey, and K. Beznosov. A billion keys, but few locks: the crisis of web single sign-on. In *Proceedings of the 2010 workshop on New security paradigms*, pages 61–72. ACM, 2010.

[48] K. Thomas, C. Grier, D. Song, and V. Paxson. Suspended accounts in retrospect: an analysis of Twitter spam. In *Proceedings of the 2011 ACM SIGCOMM Internet Measurement Conference*, pages 243–258. ACM, 2011.

[49] K. Thomas, D. McCoy, and C. Grier. Trafficking fraudulent accounts: the role of the underground market in twitter spam and abuse. In *Proceedings of the 22nd USENIX Security Symposium*, pages 195–210. USENIX Association, 2013.

[50] J. Tygar. Adversarial machine learning. *IEEE Internet Computing*, 15(5), 2011.

[51] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove. An analysis of social network-based sybil defenses. In *Proceedings of ACM SIGCOMM Computer Communication Review*, pages 363–374. ACM, 2010.

[52] B. Viswanath, M. Mondal, A. Clement, P. Druschel, K. P. Gummadi, A. Mislove, and A. Post. Exploring the design space of social network-based sybil defenses. In *In Proceedings of the 4th International Conference on Communication Systems and Networks*, pages 1–8. IEEE, 2012.

[53] C. Wagner, S. Mitter, C. Körner, and M. Strohmaier. When social bots attack: Modeling susceptibility of users in online social networks. In *Proceedings of the WWW*, volume 12, 2012.

[54] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao. You are how you click: Clickstream analysis for sybil detection. In *Proceedings of the 22nd USENIX Security Symposium*, pages 1–8. USENIX Association, 2013.

[55] Y. Wang, P. G. Leon, K. Scott, X. Chen, A. Acquisti, and L. F. Cranor. Privacy nudges for social media: an exploratory facebook study. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 763–770. International World Wide Web Conferences Steering Committee, 2013.

[56] Y. Xie, F. Yu, Q. Ke, M. Abadi, E. Gillum, K. Vitaldevaria, J. Walter, J. Huang, and Z. M. Mao. Innocent by association: early recognition of legitimate users. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 353–364. ACM, 2012.

[57] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu. Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter. In *Proceedings of the 21st international conference on World Wide Web*, pages 71–80. ACM, 2012.

[58] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai. Uncovering social network sybils in the wild. In *Proceedings of the 2011 ACM SIGCOMM Internet Measurement Csonference*, pages 259–268. ACM, 2011.

[59] H. Yu. Sybil defenses via social networks: a tutorial and survey. *ACM SIGACT News*, 42(3):80–101, 2011.

[60] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. *ACM SIGCOMM Computer Communication Review*, 36(4):267–278, 2006.

[61] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 3–17. IEEE, 2008.