Towards Detecting Anomalous User Behavior in Online Social Networks

Bimal Viswanath
MPI-SWS

M. Ahmad Bashir
MPI-SWS

Mark Crovella
Boston University

Saikat Guha
MSR India

Krishna P. Gummadi
MPI-SWS

Balachander Krishnamurthy
AT&T Labs–Research

Alan Mislove
Northeastern University

Abstract

Users increasingly rely on crowdsourced information, such as reviews on Yelp and Amazon, and liked posts and ads on Facebook. This has led to a market for black-hat promotion techniques via fake (e.g., Sybil) and compromised accounts, and collusion networks. Existing approaches to detect such behavior relies mostly on supervised (or semi-supervised) learning over known (or hypothesized) attacks. They are unable to detect attacks missed by the operator while labeling, or when the attacker changes strategy.

We propose using unsupervised anomaly detection techniques over user behavior to distinguish potentially bad behavior from normal behavior. We present a technique based on Principal Component Analysis (PCA) that models the behavior of normal users accurately and identifies significant deviations from it as anomalous. We experimentally validate that normal user behavior (e.g., categories of Facebook pages liked by a user, rate of like activity, etc.) is contained within a low-dimensional subspace amenable to the PCA technique. We demonstrate the practicality and effectiveness of our approach using extensive ground-truth data from Facebook: we successfully detect diverse attacker strategies—fake, compromised, and colluding Facebook identities—with no *a priori* labeling while maintaining low false-positive rates. Finally, we apply our approach to detect click-spam in Facebook ads and find that a surprisingly large fraction of clicks are from anomalous users.

1 Introduction

The black-market economy for purchasing Facebook likes,[1] Twitter followers, and Yelp and Amazon reviews has been widely acknowledged in both industry and academia [6, 27, 37, 58, 59]. Customers of these black-market services seek to influence the otherwise "organic" user interactions on the service. They do so through a variety of constantly-evolving strategies including fake (e.g., Sybil) accounts, compromised accounts where malware on an unsuspecting user's computer clicks likes or posts reviews without the user's knowledge [35], and incentivized collusion networks where users are paid to post content through their account [7, 8].

When (if) an attack is detected, the affected service usually takes corrective action which may include suspending the identities involved in the attack or nullifying the impact of their attack by removing their activity in the service. One approach for defense used today by sites like Facebook is to raise the barrier for creating fake accounts (by using CAPTCHAs or requiring phone verification). However, attackers try to evade these schemes by using malicious crowdsourcing services that exploit the differences in the value of human time in different countries. Another approach used widely today is to detect misbehaving users after they join the service by analyzing their behavior. Techniques used to address this problem to date have focused primarily on detecting specific attack strategies, for example, detecting Sybil accounts [10, 65, 67], or detecting coordinated posting of content [36]. These methods operate by assuming a particular attacker model (e.g., the attacker is unable to form many social links with normal users) or else they train on known examples of attack traffic, and find other instances of the same attack. Unfortunately, these approaches are not effective against an adaptive attacker. It is known that attackers evolve by changing their strategy, e.g., using compromised accounts with legitimate social links instead of fake accounts [14, 15, 35], to avoid detection.

In this paper we investigate a different approach: detecting *anomalous* user behavior that deviates significantly from that of normal users. Our key insight, which we validate empirically, is that normal user behavior in online social networks can be modeled using

[1] When printed in this font, likes refer to Facebook "Like"s (i.e., the action of clicking on a Like button in Facebook).

only a small number of suitably chosen latent features. Principal Component Analysis (PCA), a technique with well-known applications in uncovering network traffic anomalies [44], can be used to uncover anomalous behavior. Such anomalous behavior may then be subjected to stricter requirements or manual investigations.

We make the following three contributions: First, we introduce the idea of using PCA-based anomaly detection of user behavior in online social networks. PCA-based anomaly detection requires that user behavior be captured in a small number of dimensions. As discussed in more detail in Section 4, using over two years of complete user behavior data from nearly 14K Facebook users, 92K Yelp users, and 100K Twitter users (all sampled uniformly at random), we find that the behavior of normal users on these social networks can be captured in the top three to five principal components. Anomalous behavior, then, is user behavior that cannot be adequately captured by these components. Note that unlike prior proposals, *we do not require labeled data in training the detector*. We train our anomaly detector on a (uniformly) random sampling of Facebook users which contains some (initially unknown) fraction of users with anomalous behavior. Using PCA we are able to distill a detector from this unlabeled data as long as a predominant fraction of users exhibit normal behavior, a property which is known to hold for Facebook.

Second, we evaluate the accuracy of our PCA-based anomaly detection technique on ground-truth data for a diverse set of normal and anomalous user behavior on Facebook. To do so, we acquired traffic from multiple black-market services, identified compromised users, and obtained users who are part of incentivized collusion networks. Our approach detects over 66% of these misbehaving users at less than 0.3% false positive rate. In fact, the detected misbehaving users account for a large fraction, 94% of total misbehavior (number of likes). Section 6 reports on the detailed evaluation.

Lastly, in Section 7 we apply our technique to detect anomalous ad clicks on the Facebook ad platform. Where only 3% of randomly sampled Facebook users had behavior flagged by us as anomalous (consistent with Facebook's claims [32]), a significantly higher fraction of users liking our Facebook ads had behavior flagged as anomalous. Upon further investigation we find that the like activity behavior of these users is indistinguishable from the behavior of black-market users and compromised users we acquired in the earlier experiment. Our data thus suggests that while the fraction of fake, compromised or otherwise suspicious users on Facebook may be low, they may account for a disproportionately high fraction of ad clicks.

2 Overview

Our goal is to detect anomalous user behavior without *a priori* knowledge of the attacker strategy. Our central premise is that attacker behavior should appear anomalous relative to normal user behavior along some (unknown) latent features. Principal Component Analysis (PCA) is a statistical technique to find these latent features. Section 3 describes PCA and our anomaly-detection technique in detail. In this section we first build intuition on why attacker behavior may appear anomalous relative to normal user behavior (regardless of the specific attacker strategy), and overview our approach.

2.1 Illustrative Example and Intuition

Consider a black-market service that has sold a large number of Facebook likes in some time frame to a customer (e.g., the customer's page will receive 10K likes within a week). Since a Facebook user can contribute at most one like to a given page, the black-market service needs to orchestrate likes from a large number of accounts. Given the overhead in acquiring an account—maintaining a fake account or compromising a real account—the service can amortize this overhead by selling to a large number of customers and leveraging each account multiple times, once for each customer. Such behavior may manifest along one of two axes: temporal or spatial (or both). By *temporal* we mean that the timing of the like may be anomalous (e.g., the inter-like delay may be shorter than that of normal users, or the weekday-weekend distribution may differ from normal). By *spatial* anomaly we mean other (non-temporal) characteristics of the like may be anomalous (e.g., the distribution of page categories liked may be different, or combinations of page categories rarely liked together by normal users may be disproportionately more frequent).

A smart attacker would attempt to appear normal along as many features as possible. However, each feature along which he must constrain his behavior reduces the amortization effect, thus limiting the scale at which he can operate. We show in Section 6 that black-market users we purchased have nearly an order of magnitude larger number of likes than normal users, and four times larger number of categories liked. If the attacker constrained himself to match normal users, he would require significantly more accounts to maintain the same level of service, adversely affecting profitability.

In the above illustrative example, it is not clear that the number of likes and categories liked are the best features to use (in fact, in section 6.4 we show that such simple approaches are not very effective in practice). Some other feature (or combination of features) that is even more discriminating between normal and anomalous behavior and more constraining for the attacker may be better. Assuming we find such a feature, hard-coding that feature into the anomaly detection algorithm is undesirable in case "normal" user behavior changes. Thus, our approach must automatically find the most discriminating features to use from unlabeled data.

2.2 Approach

At a high level, we build a model for normal user behavior; any users that do not fit the model are flagged as anomalous. We do not make any assumptions about attacker strategy. We use PCA to identify features (dimensions) that best explain the predominant normal user behavior. PCA does so by projecting high-dimensional data into a low-dimensional subspace (called the *normal subspace*) of the top-N principal components that accounts for as much variability in the data as possible. The projection onto the remaining components (called the *residual subspace*) captures anomalies and noise in the data.

To distinguish between anomalies and noise, we compute bounds on the L^2 norm [43] in the residual subspace such that an operator-specified fraction of the *unlabeled* training data (containing predominantly normal user behavior) is within the bound. Note that the normal users do not need to be explicitly identified in the input dataset. When testing for anomalies, any data point whose L^2 norm in the residual subspace exceeds the bound is flagged as anomalous.

2.3 Features

We now discuss the input features to PCA that we use to capture user behavior in online social networks. We focus on modeling Facebook like activity behavior and describe suitable features that capture this behavior.

Temporal Features: We define a temporal feature as a *time-series of observed values*. The granularity of the time-series, and the nature of the observed value, depends on the application. In this paper, we use the number of likes at a per-day granularity. In general, however, the observed value may be the time-series of number of posts, comments, chat messages, or other user behavior that misbehaving users are suspected of engaging in.

Each time-bucket is a separate dimension. Thus, for a month-long trace, the user's like behavior is described by a $\sim$30-dimensional vector. The principal components chosen by PCA from this input set can model inter-like delay (i.e., periods with no likes), weekday-weekend patterns, the rate of change of like activity, and other latent features that are linear combinations of the input features, without us having to explicitly identify them.

Spatial Features: We define a spatial feature as a *histogram of observed values*. The histogram buckets depend on the application. In this paper, we use the category of Facebook pages (e.g., sports, politics, education) as buckets, and number of likes in each category as the observed value. In general, one might define histogram buckets for any attribute (e.g., the number of words in comments, the number of friends tagged in photos posted, page-rank of websites shared in posts, etc).

As with temporal features, each spatial histogram bucket is a separate dimension. We use the page categories specified by Facebook[2] to build the spatial feature vector describing the user's like behavior, which PCA then reduces into a low-dimensional representation.

Spatio-Temporal Features: Spatio-temporal features combine the above two features into a single feature, which captures the *evolution of the spatial distribution of observed values* over time. In essence, it is a time-series of values, where the value in each time bucket summarizes the spatial distribution of observed values at that time. In this paper, we use *entropy* to summarize the distribution of like categories. Entropy is a measure of information content, computed as $-\sum_i p_i \log_2 p_i$, where bucket i has probability p_i. In general, one might use other metrics depending on the application.

Multiple Features: Finally, we note that temporal, spatial, and spatio-temporal features over multiple kinds of user behavior can be combined by simply adding them as extra dimensions. For instance, like activity described using l_T temporal dimensions, l_S spatial dimensions, and l_{ST} spatio-temporal dimensions, and wall posting activity described similarly (p_T, p_S, p_{ST}), can be aggregated into a vector with $\sum_x l_x + \sum_x p_x$ dimensions passed as input into PCA.

3 Principal Component Analysis (PCA)

Principal component analysis is a tool for finding patterns in high-dimensional data. For a set of m users and n dimensions, we arrange our data in an $m \times n$ matrix $\mathbf{X}$, whose rows correspond to users and whose columns correspond to user behavior features discussed above. PCA then extracts common patterns from the rows of $\mathbf{X}$ in an optimal manner. These common patterns are called *principal components*, and their optimality property is as follows: over the set of all unit vectors having n elements, the first principal component is the one that captures the *maximum variation* contained in the rows of $\mathbf{X}$. More formally, the first principal component v_1 is given by:

$$v_1 = \arg \max_{||v||=1} ||\mathbf{X}v||.$$

The expression $\mathbf{X}v$ yields the inner product (here, equivalent to the correlation) of v with each row of $\mathbf{X}$; so v_1

[2] Facebook associates a topic category to each Facebook page which serves as the category of the like.

maximizes the sum of the squared correlations. Loosely, v_1 can be interpreted as the n-dimensional pattern that is most prevalent in the data. In analogous fashion, for each k, the k^{th} principal component captures the maximum amount of correlation beyond what is captured by the previous $k-1$ principal components.

The principal components $v_1, \ldots, v_n$ are constructed to form a *basis* for the rows of $\mathbf{X}$. That is, each row of $\mathbf{X}$ can be expressed as a linear combination of the set of principal components. For any principal component v_k, the amount of variation in the data it captures is given by the corresponding *singular value* σ_k.

A key property often present in matrices that represent measurement data is that only a small subset of principal components suffice to capture most of the variation in the rows of $\mathbf{X}$. If a small subset of singular values are much larger than the rest, we say that the matrix has *low effective dimension.* Consider the case where r singular values $\sigma_1, \ldots, \sigma_r$ are significantly larger than the rest. Then we know that each row of $\mathbf{X}$ can be approximated as a linear combination of the first r principal components $v_1, \ldots, v_r$; that is, $\mathbf{X}$ has *effective dimension* r.

Low effective dimension frequently occurs in measurement data. It corresponds to the observation that the number of factors that determine or describe measured data is not extremely large. For example, in the case of human-generated data, although data items (users) may be described as points in high-dimensional space (corresponding to the number of time bins or categories), in reality, the set of factors that determine typical human behavior is not nearly so large. A typical example is the user-movie ranking data used in the Netflix prize; while the data matrix of rankings is of size about 550K users $\times$ 18K movies, reasonable results were obtained by treating the matrix as having an effective rank of 20 [41]. In the next section, we demonstrate that this property also holds for user behavior in online social networks.

4 Dimensioning OSN User Behavior

To understand dimensionality of user behavior in online social networks, we analyze a large random sampling of users from three sources: Facebook, Yelp, and Twitter. The Facebook data is new in this study, while the Yelp and Twitter datasets were repurposed for this study from [50] and [4] respectively. We find low-effective dimension in each dataset as discussed below.

4.1 User Behavior Datasets

We use Facebook's people directory [25] to sample Facebook users uniformly at random.[3] The directory summarizes the number of people whose names start with a given character `x`, and allows direct access to the $\mathtt{y}^{th}$ user with name starting with `x` at `https://www.facebook.com/directory/people/x-y`. We sample uniformly at random from all possible (1.14B) `x-y` pairs, and follow a series of links to the corresponding user's profile.

We collected the publicly visible like and Timeline [34] activity of 13,991 users over the 26 month period ending in August 2013. For each user, we record three types of features: (i) *temporal*, a time-series of the number of likes at day granularity resulting in 181 dimensions for a 6-month window, (ii) *spatial*, a histogram of the number of likes in the 224 categories defined by Facebook, and (iii) *spatio-temporal*, a time-series of entropy of like categories at day granularity (181 dimensions for 6 months). We compute the entropy H_t on day t as follows: for a user who performs n_t^i likes in category i on day t, and n_t likes in total on day t, we compute $H_t = -\sum_i \frac{n_t^i}{n_t} \log_2 \frac{n_t^i}{n_t}$.

The Yelp dataset consists of all 92,725 Yelp reviewers in the San Francisco area [50] who joined before January 2010 and were active (wrote at least one review) between January 2010 and January 2012. The spatial features are constructed by a histogram of number of reviews posted by the user across 445 random groupings of 22,250 businesses[4] and 8 additional features (related to user reputation provided by Yelp[5]). The dataset also contains temporal features, the time-series of the number of reviews posted by a user at day granularity resulting in 731 dimensions covering the two year period.

The Twitter dataset consists of a random sample of 100K out of the 19M Twitter users who joined before August 2009 [4]. Previous work [4] identified topical experts in Twitter and the topics of interests of users were inferred (e.g., technology, fashion, health, etc) by analyzing the profile of topical experts *followed* by users. In this dataset, each expert's profile is associated with a set of topics of expertise. We construct a spatial histogram by randomly grouping multiple topics (34,334 of them) into 687 topic-groups and counting the number of experts a user is following in a given topic-group. The Twitter dataset does not have temporal features.

4.2 Low-Dimensionality of User Behavior

A key observation in our results from all three online social networks (Facebook, Yelp, Twitter) across the three user behaviors (temporal, spatial, and spatio-temporal)

[3] Users may opt-out of this directory listing. However, our analysis found 1.14 billion users listed in the directory as of April 2013, while Facebook reported a user count of 1.23 billion in December 2013 [31]. We therefore believe the directory to be substantially complete and representative.

[4] Randomly grouping the feature space helps compress the matrix without affecting the dimensionality of the data [13].

[5] Examples of reputation features include features such as number of review endorsements and number of fans.

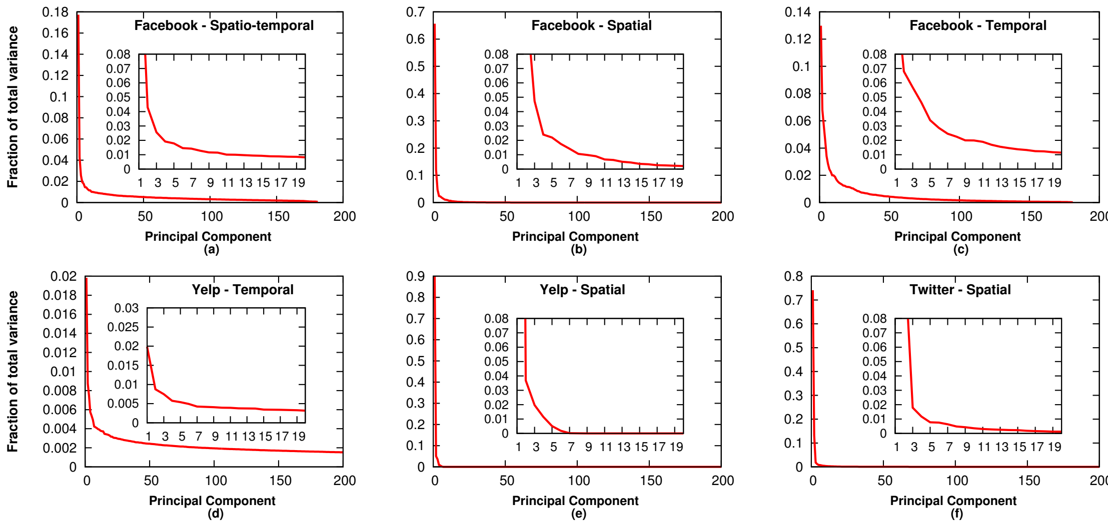


Figure 1: **Scree plots showing low-dimensionality of normal user behavior. A significant part of variations can be captured using the top three to five principal components (the "knee" of the curves).**

is that they all have low effective dimension. Figure 1 presents *scree plots* that show how much each principal component contributes when used to approximate the user behavior matrix $\mathbf{X}$, and so gives an indication of the effective dimension of $\mathbf{X}$. The effective dimension is the x-value at the "knee" of the curve (more clearly visible in the inset plot that zooms into the lower dimensions), and the fraction of the area under the curve left of the knee is the total variance of the data accounted for. In other words, the important components are the ones where the slope of the line is very steep, and the components are less important when the slope becomes flat. This method of visually inspecting the scree plot to infer the effective dimension is known as Cattell's Scree test in the statistics literature [5].

For Facebook like behavior (Figure 1(a)–(c)), the knee is around five principal components. In fact, for spatial features in Facebook like activity (Figure 1(b)), these top five components account for more than 85% of the variance in the data. We perform a parameter sweep in Section 6 and find that our anomaly detector is not overly sensitive (detection rate and false positives do not change drastically) to minor variations in the choice of number of principal components [54]. Yelp and Twitter (Figure 1(d)–(f)) show a knee between three and five dimensions as well. Overall, across all these datasets where the input dimensionality for user behavior were between 181 and 687, we find that the effective dimensionality is around three to five dimensions.

5 Detecting Anomalous User Behavior

In this section, we elaborate on the normal subspace and residual subspace discussed in Section 2, and describe how an operator can use them to detect anomalous behavior.

The operation of separating a user's behavior into principal components can be expressed as a *projection* . Recall that the space spanned by the top k principal components $v_1, \ldots, v_k$ is called the *normal subspace.* The span of the remaining dimensions is referred to as the *residual subspace.* To separate a user's behavior, we project it onto each of these subspaces. Formulating the projection operation computationally is particularly simple since the principal components are unit-norm vectors. We construct the $n \times k$ matrix $\mathbf{P}$ consisting of the (column) vectors $v_1, \ldots, v_k$. For a particular user's behavior vector x, the normal portion is given by $x_n = \mathbf{P}\mathbf{P}^T x$ and the residual portion is given by $x_r = x - x_n$.

The intuition behind the *residual subspace detection* method for detecting anomalies is that if a user's behavior has a large component that cannot be described in terms of *most* user's behavior, it is anomalous. Specifically, if $\|x_r\|_2$ is unusually large where $\|\cdot\|_2$ represents the L^2 norm, then x is likely anomalous. This requires setting thresholds for $\|x_r\|_2^2$ known as the squared prediction error or SPE [44]. We discuss how we choose a threshold in Section 6.

5.1 Deployment

In practice, we envision our scheme being deployed by the social network operator (e.g., Facebook), who has access to all historical user behavior information. The provider first selects a time window in the past (e.g., $T =$ 6 months) and a large random sample of users active during that time (e.g., 1M) whose behavior will be used to train the detector. As described earlier, training involves extracting the top k principal components that define the normal and residual subspace for these users. This training is repeated periodically (e.g., every six months) to account for changes in normal user behavior.

The service provider detects anomalous users periodically (e.g., daily or weekly) by constructing the vector of user behavior over the previous T months, projecting it onto the residual subspace from the (latest) training phase, and analyzing the L^2 norm as discussed earlier. Since each user is classified independently, classification can be trivially parallelized.

6 Evaluation

We now evaluate the effectiveness of our anomaly detection technique using real-world ground-truth data about normal and anomalous user behavior on Facebook. Our goal with anomaly detection in this section is to detect Facebook like spammers.

6.1 Anomalous User Ground Truth

We collected data for three types of anomalous behaviors: fake (Sybil) accounts that do not have any normal user activity, compromised accounts where the attacker's anomalous activity interleaves with the user's normal activity, and collusion networks where users collectively engage in undesirable behavior. We used the methods described below to collect data for over 6.8K users. We then used Selenium to crawl the publicly visible data for these users, covering 2.16M publicly-visible likes and an additional 1.19M publicly-visible Timeline posts including messages, URLs, and photos. We acquired all activity data for these users from their join date until end of August 2013.

Black-Market Services: We searched on Google for websites offering paid Facebook likes (query: "buy facebook likes"). We signed up with six services among the top search results and purchased the (standard) package for 1,000 likes; we paid on average \$27 to each service. We created a separate Facebook page for each service to like so we could track their performance. Four of the services [18–21] delivered on their promise (3,437 total users), while the other two [22, 23] did not result in any likes despite successful payment.

As mentioned, we crawled the publicly-visible user behavior of the black-market users who liked our pages. We discovered 1,555,534 likes (with timestamps at day granularity) by these users. We further crawled the users' publicly visible Timeline for public posts yielding an additional 89,452 Timeline posts.

Collusion Networks: We discovered collaborative services [7, 8] where users can collaborate (or collude) to boost each other's likes. Users on these services earn virtual credits for liking Facebook pages posted by other users. Users can then "encash" these credits for likes on their own Facebook page. Users can also buy credits (using real money) which they can then encash for likes on their page. We obtained 2,259 likes on three Facebook pages we created, obtaining a set of 2,210 users, at an average cost of around \$25 for 1,000 likes. The price for each like (in virtual credits) is set by the user requesting likes; the higher the price, the more likely it is that other users will accept the offer. We started getting likes within one minute of posting (as compared to more than a day for black-market services).

As with black-market users, we crawled the user activity of the users we found through collusion networks. We collected 359,848 likes and 186,474 Timeline posts.

Compromised Accounts: We leveraged the browser malware Febipos.A [35] that infects the user's browser and (silently) performs actions on Facebook and Twitter using the credentials/cookies stored in the browser. The malware consists of a browser plugin, written in (obfuscated) Javascript, for all three major browsers: Chrome, Firefox and Internet Explorer [28, 29].

We installed the malware in a sandbox and deobfuscated and analyzed the code. The malware periodically contacts a CnC (command-and-control) server for commands, and executes them. We identified 9 commands supported by the version of the malware we analyzed: (1) like a Facebook page, (2) add comments to a Facebook post, (3) share a wall post or photo album, (4) join a Facebook event or Facebook group, (5) post to the user's wall, (6) add comments to photos, (7) send Facebook chat messages, (8) follow a Twitter user, and (9) inject third-party ads into the user's Facebook page.

We reverse-engineered the application-level protocol between the browser component and the CnC server, which uses HTTP as a transport. We then used `curl` to periodically contact the CnC to fetch the commands the CnC would have sent, logging the commands every 5 minutes. In so doing, we believe we were able to monitor the entire activity of the malware for the time we measured it (August 21–30, 2013).

Identifying which other Facebook users are compromised by Febipos.A requires additional data. Unlike in the black-market services and collusion networks—where we were able to create Facebook pages and give to the service to like— we can only passively monitor the malware and cannot inject our page for the other infected users to like (since we do not control the CnC server).

To identify other Facebook users compromised by

Febipos.A, we identified two commands issued during the week we monitored the malware: one which instructed the malware to like a specific Facebook page, and second, to join a specific Facebook event. We use Facebook's graph search [26] to find other users that liked the specific page and accepted the specific event directed by the CnC. From this list we sampled a total of 4,596 users. Note, however, that simply because a user matched the two filters does not necessarily mean they are compromised by Febipos.A.

To improve our confidence in compromised users, we clustered the posts (based on content similarity) made to these users' walls and manually inspected the top 20 most common posts. Among these 20 posts, two posts looked suspicious. Upon further investigation, we found out that one of the post was also found on pages the malware was directed to like. The other post was present in the CnC logs we collected. The first was posted by 1,173 users while the second was posted by 135 users. We considered users from both these clusters and obtained a set of 1,193 unique users.[6] We collected 247,589 likes and 916,613 Timeline posts from their profile.

6.2 Ethics

We note that all money we paid to acquire anomalous likes were exclusively for pages both controlled by us and setup for the sole purpose of conducting the experiments in this paper. For the malware analysis, we ensured that our sandbox prevented the malware from executing the CnC's instructions. We did not seek or receive any account credentials of any Facebook user. Overall, we ensured that no other Facebook page or user was harmed or benefited as a result of this research experiment.

6.3 Normal User Ground Truth

We collected three datasets to capture normal user behavior. The first dataset is the 719 users that are part of the SIGCOMM [33] and COSN [24] Facebook groups. We picked these technically savvy users, despite the obvious bias, because we presume that these users are less likely to be infected by browser or other malware which we have found to be stealthy enough to avoid detection by non-technically-savvy users. An anomaly detector that has low false-positives on both this dataset as well as a more representative Facebook dataset is more likely to have a range that spans the spectrum of user behavior on Facebook.

[6]The friendship network formed by these users has a very low edge density of 0.00023. Thus, even though they had similar posts on their Timeline, very few of them were friends with each other (further suggesting suspicious behavior).

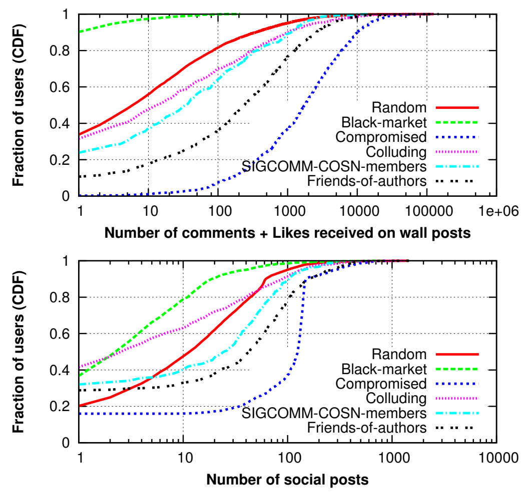


Figure 2: **Characterizing social activity of normal and anomalous users considered in our study based on activity on their Timeline.**

For our second dataset, we use the random sampling of Facebook users described in Section 4.1. Note that this dataset may be biased in the opposite direction: while it is representative of Facebook users, an unknown fraction of them are fake, compromised, or colluding. Public estimates lower-bound the number of fake users at 3% [32], thus we expect some anomalies in this dataset.

A compromise between the two extremes is our third dataset: a 1-hop crawl of the social-neighborhood of the authors (a total of 1,889 users). This dataset is somewhat more representative of Facebook than the first dataset, and somewhat less likely to be fake, compromised, or colluding than the second dataset. Users in these three datasets in total had 932,704 likes and 2,456,864 Timeline posts putting their level of activity somewhere between the black-market service on the low end, and compromised users on the high end. This fact demonstrates the challenges facing anomaly detectors based on simplistic activity thresholds.

For the rest of the analysis in this paper, we use the random sampling dataset for training our anomaly detector, and the other two datasets for testing normal users.

Figure 2 plots the cumulative distribution (CDF) of likes and comments received on wall posts and the number of *social*[7] posts for all of our six datasets. The top figure plots the CDF of likes and comments on a logarithmic x-axis ranging from 1 to 1M, and the bottom figure plots the CDF of social posts (messages, URLs, photos). As is evident from the figure, black-market users are the least active, compromised users are the most active, and all three normal user datasets—as well as the collusion network users—fall in the middle and are hard to distin-

[7]Posts that involve interaction with other users, e.g., photo tagging.

	Random	Normal	Black-market	Compromised	Colluding
#Users (#likes)	11,851 (561,559)	1,274 (73,388)	3,254 (1,544,107)	1,040 (209,591)	902 (277,600)

Table 1: **Statistics of different types of users whose like activity (from June 2011 to August 2013) we analyze.**

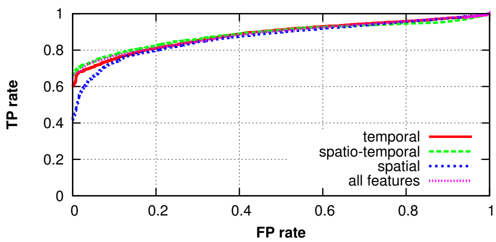


Figure 3: **ROC curve showing the performance of our anomaly detector in distinguishing between normal and misbehaving users.**

guish visually (especially for social post activity).

6.4 Detection Accuracy

Methodology: We analyze Facebook like activity from June 2011 to August 2013. We need to pay special attention to users that joined Facebook in the middle of our analysis period (or stopped being active) to avoid the degenerate case where the anomaly detection flags their lack of activity. We avoid this by considering a six-month sliding window that advances by one month. In each window, we consider users that joined before that window and had at least one like during the window. Unless otherwise mentioned, for the rest of the analysis in the paper, we consider only these users and their likes that fall within our period of analysis—data statistics are shown in Table 1. A user's behavior is flagged as anomalous if they are flagged in any one of the sliding time windows. They are flagged as anomalous in a window if the squared prediction error (SPE) exceeds the threshold parameter.

We set the detection threshold (conservatively) based on Facebook's estimate (from their SEC filings [32]) of users that violate terms of service. Facebook estimates around 3.3% users in 2013 to be undesirable (spam or duplicates). Recall that we train our anomaly detector on the like behavior of random Facebook users during much of the same period. We conservatively pick a training threshold that flags 3% of random accounts, and adjust our false-positive rate downwards by the same amount and further normalize it to lie in the range 0 to 1. We select the top-five components from our PCA output to build the normal subspace.

Results: Figure 3 plots the *receiver operating characteristic* (ROC) curve of our detector when evaluated on all datasets for normal and anomalous user behavior (except random, which was used to train the detector) as we perform a parameter-sweep on the detection threshold. The y-axis plots the true-positive rate ($\frac{TP}{TP+FN}$) and the x-axis plots the false-positive rate ($\frac{FP}{FP+TN}$) where TP, TN, FP, FN are true-positive, true-negative, false-positive, and false-negative, respectively. The area under the ROC curve for an ideal classifier is 1, and that for a random classifier is 0.5. For the mix of misbehaviors represented in our ground-truth dataset, the spatio-temporal features performs best, with an area under the curve of 0.887, followed closely by temporal and spatial features at 0.885 and 0.870, respectively.

By combining the set of users flagged by all three features, our detector is able to flag 66% of all misbehaving users at a false-positive rate of 0.3%. If we compare this with a naïve approach of flagging users based on a simple like volume/day (or like categories/day) cut-off (i.e., by flagging users who exceed a certain number of likes per day or topic categories per day) we can only detect 26% (or 49%) of all misbehaving users at the same false-positive rate. This further suggests that our PCA-based approach is more effective than such naïve approaches at capturing complex normal user behavior patterns to correctly flag misbehaving users.

Figure 4 and Table 2 explore how the set of features performed on the three classes of anomalous behavior. Spatio-temporal features alone flagged 98% of all activity for users acquired through the four black-market services. 61% (939K) of black-market activity was flagged as anomalous by all three sets of features. Due to the dominant nature of the spatio-temporal features on the black-market dataset, there is insufficient data outside the spatio-temporal circle to draw inferences about the other features. The three features performed more evenly on the dataset of compromised and colluding users, with 43.9% and 78.7% of the anomalous user behavior respectively being flagged by all three sets of features, and 64% and 91% respectively being flagged by at least one. Except in the black-market case, no class of features dominates, and combined they flag 94.3% of all anomalous user behavior in our dataset.

6.5 Error Analysis

To better understand our false-negative rate, Figure 5 plots the likelihood of detection as a function of the level of activity (number of likes) for each class of anomalous traffic. Unlike black-market users that are easily detected at any level of activity, the anomaly detector does not flag compromised and colluding users with low activity. This

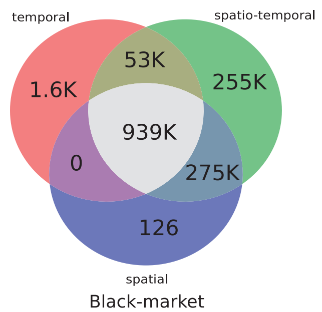


Figure 4: **Venn diagram illustrating performance of different features in detecting different classes of anomalous user behavior. The numbers indicate number of likes flagged.**

Identity type	Identities flagged	Likes flagged			
		Total	Temporal	Spatio-temporal	Spatial
Black-market	2,987/3,254 (91%)	1,526,334/1,544,107 (98%)	994,608 (64%)	1,524,576 (98%)	1,215,396 (78%)
Compromised	171/1,040 (16%)	134,320/209,591 (64%)	104,596 (49%)	123,329 (58%)	116,311 (55%)
Colluding	269/902 (29%)	254,949/277,600 (91%)	246,016 (88%)	232,515 (83%)	237,245 (85%)

Table 2: **Performance of different features in detecting different classes of anomalous user behavior.**

is consistent with compromised and colluding user behavior being a blend of normal user behavior intermixed with attacker behavior. At low levels of activity, the detector lacks data to separate anomalous behavior from noise. However, as the attacker leverages the account for more attacks, the probability of detection increases. It increases faster for colluding users, where the user is choosing to engage in anomalous activity, and more slowly for compromised accounts where the user contributes normal behavior to the blend.

Figure 6 compares anomalous user behavior that was not flagged by our detector to the behavior of normal users. As is evident from the figure, the false-negatives for compromised and colluding users appear indistinguishable from normal user behavior, especially when compared to the behavior of colluding and compromised users that were flagged. Our hypothesis (consistent with the previous paragraph) is that these false-negative users are newly compromised users or users newly recruited to the collusion network, and their overall behavior has not yet diverged significantly enough to be considered an anomaly.

Regarding false-positives, we expect some fraction of users to be flagged, since an unknown fraction of the normal users may be infected by malware. Our false-

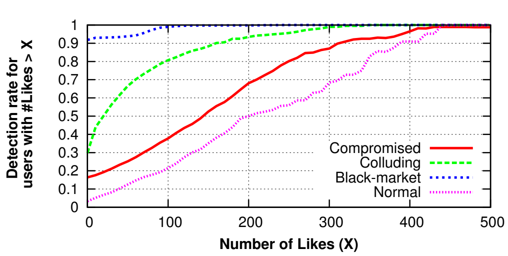


Figure 5: **Higher like activity generally correlates with higher detection rates, however limits for normal user behavior being flagged are 50–100 likes higher than for anomalous user behavior.**

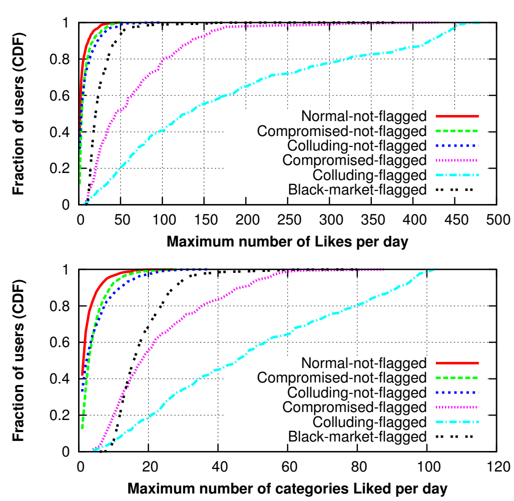


Figure 6: **Characterizing activity of users that are not flagged in the compromised and colluding set and comparing them with normal users who were not flagged.**

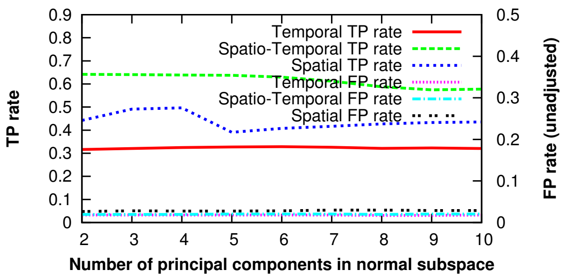


Figure 7: **False-positive rate (unadjusted) and true-positive rate as we vary the number of principal components chosen for the normal subspace. Our detector is stable for small variations in the number of principal components chosen.**

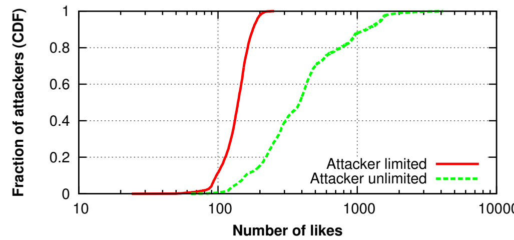


Figure 8: **Distribution of number of anomalous likes before anomalous users are flagged by our approach. For comparison, we show the actual number of anomalous likes we received.**

positive rate is under 3.3%, which when adjusted for the fraction of users Facebook expects to be anomalous [32], suggests a false-positive rate of 0.3%. We specifically note in Figure 5 that the threshold before normal user behavior is flagged is consistently 50–100 likes higher than that for compromised users for the same y-axis value. Thus, our anomaly detection technique accommodates normal users that are naturally prone to clicking on many likes.

6.6 Robustness

Next we evaluate the sensitivity of our detector to small variations in the number of principal components chosen for the normal subspace. Figure 7 plots the true-positive rate and the false-positive rate (unadjusted) as we vary k, the number of principal components used to construct the normal subspace. As is evident from the figure, our detection accuracy does not change appreciably for different choices of k. Thus our detector is quite robust to the number of principal components chosen.

6.7 Adversarial Analysis

In this section, we consider two classes of attackers: first, where the attacker scales back the attack to avoid detection, and second, where the attacker attempts to compromise the training phase.

Scaling Back: Figure 8 explores the scenario where attackers scale back their attacks to avoid detection. Specifically, we simulate the scenario where we subsample likes uniformly at random from our ground-truth attack traffic (black-market, compromised and colluding) until the point a misbehaving user is no longer flagged by the anomaly detector. As users' behavior spans multiple six month time windows, for each user we consider the window in which the user displayed maximum misbehavior (maximum number of likes in this case). In this way, we analyze the extent to which we can constrain attackers during their peak activity period. We find that our current model parameters constrains attackers by a factor of 3 in the median case, and by an order of magnitude at the 95^{th} percentile.

Compromising Training: An attacker that controls a sufficiently large number of users may attempt to compromise the training phase by injecting additional likes, thereby distorting the principal components learned for normal users [39, 55, 56]. The compromised detector would have a higher false-negative rate, since more anomalous behavior would fall within the normal subspace. At a high level, this attack may be mitigated by defense-in-depth, where multiple techniques can be used to filter users selected for the training set.

The first defense-in-depth technique is the attacker's need to control a sufficiently large number of anomalous users. We first note that our training data already contains an estimated 3% anomalous users, and that the trained detector has good performance on the ROC curve. Since users in the training set are sampled uniformly at random from all users, an attacker with equivalent power would need to be in control of over $30M$ users (given Facebook's user base of over $1B$ users). In comparison, one of the largest botnets today is estimated to have fewer than 1 million bots [47]. A related issue is that the quantity of like volume that must be injected to affect the detector depends on the overall volume of likes in the system, which is information that is not likely to be readily available to the attacker.

Assuming the attacker is able to amass this large a number of users, the next defense-in-depth technique is to sanitize training data, where anomalous users discovered in one time window are excluded from being used for training in all subsequent time windows [39]. Thus if an attacker ends up altering like traffic significantly in one time window, it could lead to detection and further removal of those anomalous users from the training set.

Finally, variants of PCA that are more robust to outliers can be used to further harden the training phase from compromise. Croux et al. [9, 39] proposed the robust PCA-GRID algorithm that reduces the effect of outliers in the training data. Using this approach one can compute principal components that maximize a more robust measure of data dispersion – the *median absolute deviation* without under-estimating the underlying variance in the data. Such an algorithm could yield robust estimates for the normal subspace.

6.8 Scalability

As discussed earlier, classifying users can be trivially parallelized once the training phase is complete. Thus our primary focus in this section is on evaluating the scalability of the training phase.

Space: The total space requirement of the training phase is $O(n \times m)$ where n is the number of input dimensions (typically a few hundred), and m is the number of users in the training set (typically a few million). Thus the space needed to store the matrix is at most a few gigabytes, which can easily fit in a typical server's memory.

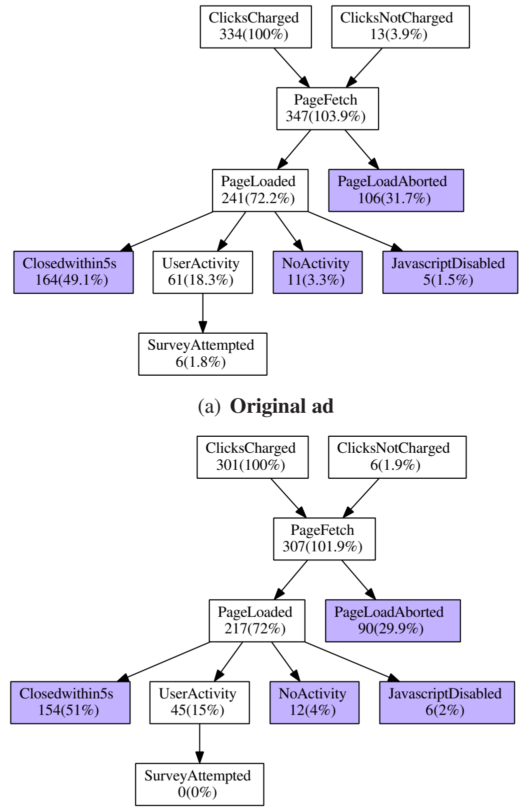
Computation: The primary computation cost in PCA arises from the eigenvalue decomposition of the covariance matrix of the feature vectors, which is a low-order polynomial time algorithm with complexity $O(n^3 + n^2 m)$. Eigenvalue decomposition is at the heart of the PageRank algorithm (used in early search engines) for which efficient systems exist to handle input data several orders of magnitude larger than our need [1]. Furthermore, efficient algorithms for PCA based on approximation and matrix sketching have been designed which have close to $O(mn)$ complexity [46, 57].

7 Detecting Click-Spam on Facebook Ads

So far, we have discussed the performance of our anomaly detector in detecting diverse attack strategies. Next, we demonstrate another real world application of our technique: detecting click-spam on Facebook ads. Click-spam in online ads—where the advertiser is charged for a click that the user did not intend to make (e.g., accidental clicks, clicks by bots or malware)—is a well-known problem in web search [11, 12], and an emerging problem for Facebook ads [2, 16, 17].

7.1 Click-Spam in Facebook

To gain a preliminary understanding of Facebook click-spam, we signed up as an advertiser on Facebook. We set up an ad campaign targeting users in the USA aged between 15 and 30. The campaign advertised a simple user survey page about Facebook's privacy settings. When clicked, the ad leads to our heavily instrumented landing page to capture any user activity such as mouse clicks, mouse movement, or keyboard strokes. Of the 334 original ad clicks Facebook charged us for, only 61 (18.3%) performed any activity on the landing page (e.g., mouse move). Figure 9(a) shows how users proceeded after clicking the ad. Percentages are relative to the number of ad clicks Facebook charged us for. Shaded boxes are undesirable terminal states that suggest click-spam. For instance, 106 users (31.7%) did not even complete the first HTTP transaction to load the page (e.g., closed the tab, or pressed the back button immediately after clicking the ad).



(a) **Original ad**

(b) **Bluff ad**

Figure 9: **Summary of click statistics for real and bluff ad campaigns on Facebook.**

To distinguish between unintentional clicks and intentional clicks followed by lack of interest in our page, we ran Bluff Ads [11,38] that are ads with identical targeting parameters as the original ad, but nonsensical content. Our bluff ad content was empty. Figure 9(b) shows that our bluff ad performed identically to the original ad, both qualitatively and quantitatively; of 301 clicks in roughly the same time-frame as the original ad, almost 30% did not complete first HTTP, etc. From our data it appears that the content of the ad has no effect on clicks on Facebook ads that we were charged for, a strong indicator of click-spam.

7.2 Anomalous Clicks in Facebook Ads

In order to analyze anomalous user behavior, our approach requires information from the user's profile. Due to a change in how Facebook redirects users on ad clicks [42], we were unable to identify the users that clicked on our ad in the experiment above. Fortunately, Facebook offers a different type of ad campaign optimization scheme—maximizing likes—where the destination must be a Facebook page as opposed to an arbitrary website. With such ads, it is possible to identify the users that clicked on such an ad, but not possible to instrument the landing page to get rich telemetry as above. We chose this campaign optimization option for maximizing likes to the advertised page.

We set up 10 ad campaigns, listed in Table 3, targeting the 18+ demographic in 7 countries: USA, UK, Australia, Egypt, Philippines, Malaysia and India. Our 10 campaigns were about generic topics such as humor, dogs, trees, and privacy awareness. Our ad contained a like button, a link to the Facebook page, some text, and an image describing the topic of the ad. We ran these ads at different points in time: Campaigns 1 to 4 were run in February 2014, while campaigns 5 to 10 were run in January 2013. In total, we received 3,766 likes for all our pages. For most of the campaigns targeting India (especially #7), we received 80% of the likes within 10 minutes, which is very anomalous.

We first checked whether we obtained most of these likes via social cascades (i.e., a user liking a page because their friend liked it), or from the Facebook ads directly. To do so, we analyzed the edge density of all friendship networks (graph formed by friendship links between users) formed by users of each ad campaign. We find the edge density of friendship networks for all campaigns to be very low (e.g., the friendship network edge density for users in campaign #8 was only 0.000032). This strongly suggests that the Facebook ads, rather than any social cascades, were responsible for the likes.

Out of 3,766 likes, we were able to crawl the identity of the users clicking like for 3,517 likes.[8] Next, we apply our anomaly detection technique from Section 5 with the same training data and model parameters that we used in Section 6 to 2,767 users (out of 3,517) who fall within our 26-month training window. The penultimate column in Table 3 lists the number of users tested in each campaign, and the last column lists the number of users flagged as click-spam.

[8]The Facebook user interface does not always show the identity of all users who like a page.

Of the 2,767 users that clicked our ads in this experiment, 1,867 were flagged as anomalous. Figure 10 plots the like activity of the users we flagged as anomalous relative to our normal user behavior dataset, and the black-market user dataset that serves as our ground-truth for anomalous user activity. The flagged users from our ad dataset have an order of magnitude more like activity than the black-market users, and nearly two orders of magnitude more like activity than normal users; they also like twice as many categories as black-market users and almost an order of magnitude more categories than normal users.

7.3 Anomaly Classification

To better understand the click-spam we observed, we attempt to classify the ad users as one of our three ground-truth anomalous behaviors: black-market, compromised, and collusion. Note that anomaly classification in this section is unrelated to the anomaly detection approach from Section 5.

We use the k-Nearest Neighbor (kNN) algorithm for classification. We train the classifier using ground-truth labels for black-market, compromised, and colluding users. The input feature vectors can be formed in different ways: First, we can capture user behavior by projecting it on to the normal and residual subspace. The normal projection reflects normal behavior and the residual projection captures noisy or deviant behavior of a user. Second, we know that user behavior can also be expressed using temporal, spatio-temporal and spatial features. By leveraging all these different combinations, we built 6 classifiers using 6 different feature vectors (2 projections

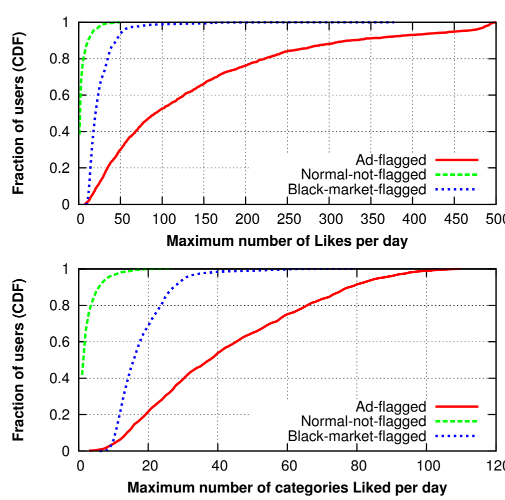


Figure 10: **Characterizing activity of users flagged in the ad set. Note that most flagged ad users like a much larger number of categories/likes per day than normal and black-market users.**

Campaign	Ad target	Cost per like (€)	Total spent (€)	Users		
				Total	Tested	Flagged
1	US	1.62	192.43	119	76	43
2	UK	1.95	230.05	118	69	27
3	AU	0.87	158.89	182	88	38
4	Egypt, Philippines, Malaysia	0.08	47.69	571	261	135
5	India	0.13	30.00	230	199	137
6	India	0.11	22.71	209	169	99
7	India	0.09	22.61	250	199	114
8	India, US, UK	0.22	242.72	1,099	899	791
9	India	0.12	30.00	247	215	143
10	India	0.07	50.00	741	632	372

Table 3: **Anomalies flagged for different ad campaigns. We observe a significant fraction of anomalous clicks for all campaigns.**

$\times$ 3 features). Each classifier, given an unlabeled user from the ad set, predicts a label for the user.

We use a simple ensemble learning technique of *majority voting* to combine the results of all the classifiers; this also means that there could be test instances that may not be labeled due to lack of consensus. We choose the most recent six-month time window (March to August 2013) in our dataset and use all known misbehaving users (black-market, compromised and colluding) in that window for training the classifier and apply this technique to the 1,408 flagged ad users who fall in that window. To balance classes for training, we randomly under-sample larger classes (black-market and colluding) and use 780 users in each of black-market, colluding and compromised set for training. For each classifier, we pick a value of k that gives the lowest misclassification rate for 10-fold cross validation on the training data. We next apply our trained classifier to predict the unlabeled ad users. Results are averaged over 50 different random trials and we observe an average misclassification rate of 31% (standard deviation of 0.5) based on cross-validation in the training phase. Table 4 shows the statistics for the labels predicted for the flagged ad users. We find that the majority of ad users (where we had majority agreement) are classified as black-market or compromised.

Classified As	Number of users
Black-market	470
Compromised	109
Colluding	345
Unclassified (no consensus)	484

Table 4: **Anomaly class predicted for the ad users that are flagged.**

While the level of anomalous click traffic is very surprising, it is still unclear what the incentives are for the attacker. One possibility is that black-market accounts and compromised accounts are clicking (liking) ads to generate cover traffic for their misbehavior. Another possibility is that the attacker is trying to drain the budget of some advertiser by clicking on ads of that advertiser. We plan to explore this further as part of future work.

8 Corroboration by Facebook

We disclosed our findings to Facebook in March 2014, and included a preprint of this paper. Our primary intent in doing so was to follow responsible disclosure procedures, and to allow Facebook to identify any ethical or technical flaws in our measurement methodology. We were informed that Facebook's automated systems detect and remove fake users and fraudulent likes.

Table 5 tabulates the users (flagged by our detector) and likes that were removed between the time we conducted our experiments and June 2014. While very few users were removed by Facebook, a sizable fraction of their likes across all pages were indeed removed confirming the accuracy of our detector. To establish a baseline for the fraction of users and likes removed by Facebook's automated systems we find that from our random user dataset (Section 4) only 2.2% users, and 32% of all their likes were removed over a ten month period. For black-market, compromised, and colluding users (ground-truth anomalous user dataset from Section 6), over 50% of all their likes had been removed over 6–10 months. Over 85% of the all likes of users that clicked our ad were removed within four months. Recall that our ad was targeted to normal Facebook users and we did not use any external services to acquire ad likes; nevertheless, 1,730 of the 3,517 likes we were charged for in February 2014 had been removed by Facebook's fraudulent like detection system by June 2014, corroborating our earlier result that a large fraction of users that clicked on our ad are anomalous both by our definition as well as Facebook's.[9] As of this writing we have not received any

[9] While Facebook allows users to un-like pages, according to Facebook insights [30] we had only 56 un-likes across all our pages, which we exclude from our analysis.

Removed by Facebook's automated systems				
	Users	**likes on all pages**	**likes on our page**	**Timespan**
Normal User Dataset (Section 4)				
Random users	262/12K	179K/561K	n/a	10 months
Ground-Truth Anomaly Dataset (Section 6)				
Black-market	228/2987	715K/1.5M	2829/3475	10 months
Compromised	3/171	80K/134K	n/a	7 months
Colluding	9/269	181K/254K	1879/2259	6 months
Facebook Ads Dataset (Section 7)				
Ad clicks	51/1867	2.9M/3.4M	1730/3517[9]	4 months

Table 5: **Fraction of users and likes flagged by us removed by Facebook's automated system, as of June 2014.**

credit adjustments for the likes charged to our advertiser account that Facebook's fraudulent like detection system since identified and removed.

9 Related Work

We survey approaches to detecting misbehaving identities along three axes.

Leveraging Hard-to-earn Attributes: Manual verification of users would be ideal to avoiding Sybils in crowdsourcing systems but does not scale for large-scale systems. Additionally, normal users may not join the system for privacy reasons due to the effort required to be verified. Current systems typically employ CAPTCHA or phone verification to raise the barrier by forcing the attacker to expend greater effort. Although pervasive, attackers try to evade these schemes by employing Sybil identities that use sites like Freelancer or Amazon's Mechanical Turk to exploit the differences in value of human time in different countries [51]. However, steps taken by service providers to raise the barrier for fake account creation complements our proposed defense because each account flagged as anomalous raises the cost for the attacker.

In OSNs, where identities are associated with each other through hard-to-earn endorsement and friend edges, several graph-based Sybil detection schemes have been developed over the years [10, 52, 61, 66, 67]. Such schemes make assumptions about the OSN graph growth and structure, for example that creating and maintaining edges to honest identities requires significant effort [48], or that honest OSN regions are fast-mixing [66, 67]. However, recent studies cast doubts on these assumptions and subsequently on the graph-based Sybil defense techniques. Specifically, Yang et al. [65] observe that Sybils blend well into the rest of OSN graphs, while Mohaisen et al. [49] find that most OSN graphs are not fast-mixing, and that detection schemes may end up accepting Sybil identities and/or wrongly expelling honest identities [62].

Supervised Learning: Most existing work on detecting misbehaving identities in social networks leverage supervised learning techniques [14,40,53]. Lee et al. [40] propose a scheme that deploys honeypots in OSNs to attract spam, trains a machine learning (ML) classifier over the captured spam, and then detects new spam using the classifier. Rahman et al. [53] propose a spam and malware detection scheme for Facebook using a Support Vector Machines-based classifier trained using the detected malicious URLs. The COMPA scheme [14] creates statistical behavioral profiles for Twitter users, trains a statistical model with a small manually labeled dataset of both benign and misbehaving users, and then uses it to detect compromised identities in Twitter.

While working with large crowdsourcing systems, supervised learning approaches have inherent limitations. Specifically they are attack-specific and vulnerable to *adaptive* attacker strategies. Given the adaptability of the attacker strategies, to maintain efficacy, supervised learning approaches require labeling, training, and classification to be done periodically. In this cat-and-mouse game, they will always lag behind attackers who keep adapting to make a classification imprecise.

Unsupervised Learning: Unsupervised learning-based anomaly detection has been found to be an effective alternative to non-adaptive supervised learning strategies [12, 45, 60, 63, 64]. For example, Li et al. [45] propose a system to detect volume anomalies in network traffic using unsupervised PCA-based methods. AutoRE [64] automatically extracts spam URL patterns in email spam based on detecting the bursty and decentralized nature of botnet traffic as anomalous.

In crowdsourcing scenarios, Wang et al. [63] proposed a Sybil detection technique using server-side clickstream models (based on user behavior defined by click-through events generated by users during their social network browsing sessions). While the bulk of the paper presents supervised learning schemes to differentiate between Sybil and non-Sybils based on their clickstream behavior, they also propose an unsupervised approach that builds clickstream behavioral clusters that capture normal behavior and users that are not part of normal clusters are flagged as Sybil. However, their approach still requires some constant amount of ground-truth information to figure out clusters that represent normal click-stream behavior. Tan et al. [60] use a user-link graph along with the OSN graph to detect some honest users with supervised ML classifier and then perform an unsupervised analysis to detect OSN spam. CopyCatch [3] detects fraudulent likes by looking for a specific attack signature — groups of users liking the same page at around the same time (*lockstep behavior*). CopyCatch is actively used in Facebook to detect fraudulent likes, however as evidenced in Table 5, it is not a silver-bullet.

While we welcome the push towards focusing more on unsupervised learning strategies for misbehavior detection, most of the current techniques are quite ad hoc and complex. Our approach using Principal Component Analysis provides a more systematic and general framework for modeling user behavior in social networks, and in fact, our PCA-based approach could leverage the user behavior features (e.g., user click-stream models [63]) used in existing work for misbehavior detection.

10 Conclusion

We propose using Principal Component Analysis (PCA) to detect anomalous user behavior in online social networks. We use real data from three social networks to demonstrate that normal user behavior is low-dimensional along a set of latent features chosen by PCA. We also evaluate our anomaly detection technique using extensive ground-truth data of anomalous behavior exhibited by fake, compromised, and colluding users. Our approach achieves a detection rate of over 66% (covering more than 94% of misbehavior) with less than 0.3% false positives. Notably we need no *a priori* labeling or tuning knobs other than a configured acceptable false positive rate. Finally, we apply our anomaly detection technique to effectively identify anomalous likes on Facebook ads.

Acknowledgements

We thank the anonymous reviewers and our shepherd, Rachel Greenstadt, for their helpful comments. We also thank Arash Molavi Kakhki for his assistance with the Yelp dataset and Mainack Mondal for helping with the ad experiments. This research was supported in part by NSF grants CNS-1054233, CNS-1319019, CNS-0905565, CNS-1018266, CNS-1012910, and CNS-1117039, and by the Army Research Office under grant W911NF-11-1-0227.

References

[1] Apache Mahout. http://mahout.apache.org/.

[2] Facebook Advertisement Value (BBC). http://www.bbc.co.uk/news/technology-18813237.

[3] Beutel, A., Xu, W., Guruswami, V., Palow, C., and Faloutsos, C. Copycatch: Stopping group attacks by spotting lockstep behavior in social networks. In *Proc. of WWW* (2013).

[4] Bhattacharya, P., Ghosh, S., Kulshrestha, J., Mondal, M., Zafar, M. B., Ganguly, N., and Gummadi, K. P. Deep Twitter Diving: Exploring Topical Groups in Microblogs at Scale. In *Proc. of CSCW* (2014).

[5] Cattell, R. B. The Scree Test For The Number Of Factors. *Journal of Multivariate Behavioral Research 1*, 2 (1966).

[6] Click Farms (Washington Post). http://www.washingtonpost.com/blogs/wonkblog/wp/2014/01/06/click-farms-are-the-new-sweatshops.

[7] Collusion Network Site #1. http://addmefast.com/.

[8] Collusion Network Site #2. http://likesasap.com/.

[9] Croux, C., Filzmoser, P., and Oliveira, M. Algorithms for ProjectionPursuit Robust Principal Component Analysis. *Journal of Chemometrics and Intelligent Laboratory Systems 87*, 2 (2007).

[10] Danezis, G., and Mittal, P. SybilInfer: Detecting Sybil Nodes Using Social Networks. In *Proc. of NDSS* (2009).

[11] Dave, V., Guha, S., and Zhang, Y. Measuring and Fingerprinting Click-Spam in Ad Networks. In *Proc. of SIGCOMM* (2012).

[12] Dave, V., Guha, S., and Zhang, Y. ViceROI: Catching Click-Spam in Search Ad Networks. In *Proc. of CCS* (2013).

[13] Ding, Q., and Kolaczyk, E. D. A Compressed PCA Subspace Method for Anomaly Detection in High-Dimensional Data. *IEEE Transactions on Information Theory 59*, 11 (2013).

[14] Egele, M., Stringhini, G., Kruegel, C., and Vigna, G. COMPA: Detecting Compromised Accounts on Social Networks. In *Proc. of NDSS* (2013).

[15] Facebook Accounts Compromised. http://mashable.com/2013/12/04/hackers-facebook-twitter-gmail/.

[16] Facebook Ads Fake Clicks. http://www.businessinsider.com/mans-600000-facebook-ad-disaster-2014-2.

[17] Facebook Advertisement Value (Salon). http://www.salon.com/2014/02/14/facebooks_big_like_problem_major_money_and_major_scams/.

[18] Facebook Black Market Service #1. http://fbviro.com.

[19] Facebook Black Market Service #2. http://get-likes.com/facebook-likes-store/.

[20] Facebook Black Market Service #3. http://www.buyfbservices.com.

[21] Facebook Black Market Service #4. http://twittertechnology.com/facebook.

[22] Facebook Black Market Service #5. http://www.pagelution.com/buy-facebook-fans.html.

[23] Facebook Black Market Service #6. http://teamfollowpromo.com/facebooklikes.

[24] Facebook COSN Group. https://www.facebook.com/groups/cosn2013.

[25] Facebook Directory. https://www.facebook.com/directory/people/.

[26] Facebook Graph Search. https://www.facebook.com/about/graphsearch.

[27] Fake Likes on Facbook and Instagram. http://www.huffingtonpost.com/2013/08/16/fake-instagram-likes_n_3769247.html.

[28] Facebook Malware Extension. http://bit.ly/1mIum7L.

[29] Facebook Internet Explorer Malware. http://blogs.technet.com/b/mmpc/archive/2013/11/14/febipos-for-internet-explorer.aspx.

[30] Facebook Page insights. https://www.facebook.com/help/336893449723054.

[31] Facebook Quarterly Report (2013). http://investor.fb.com/releasedetail.cfm?ReleaseID=821954.

[32] Facebook's estimate of fraction of undesirable accounts for 2013. http://investor.fb.com/.

[33] Facebook SIGCOMM Group. https://www.facebook.com/groups/sigcomm/.

[34] Facebook Timeline. https://www.facebook.com/about/timeline.

[35] Febipos.A Malware. http://www.microsoft.com/security/portal/threat/encyclopedia/Entry.aspx?Name=Trojan:JS/Febipos.A.

[36] GAO, H., HU, J., WILSON, C., LI, Z., CHEN, Y., AND ZHAO, B. Y. Detecting and Characterizing Social Spam Campaigns. In *Proc. of IMC* (2010).

[37] GRIER, C., THOMAS, K., PAXSON, V., AND ZHANG, C. M. @spam: The Underground on 140 Characters or Less. In *Proc. of CCS* (2010).

[38] HADDADI, H. Fighting Online Click-Fraud Using Bluff Ads. *ACM SIGCOMM Computer Communications Review 40*, 2 (2010).

[39] HUANG, L., JOSEPH, A. D., NELSON, B., RUBINSTEIN, B. I., AND TYGAR, J. D. Adversarial Machine Learning. In *Proc. of AISec* (2011).

[40] K. LEE AND J. CAVERLEE AND AND S. WEBB. Uncovering Social Spammers: Social Honeypots + Machine Learning. In *Proc. of SIGIR* (2010).

[41] KOREN, Y. Collaborative filtering with temporal dynamics. In *Proc. of KDD* (2009).

[42] KRISHNAMURTHY, B., AND WILLS, C. E. On the Leakage of Personally Identifiable Information via Online Social Networks. *ACM SIGCOMM Computer Communications Review 40*, 1 (2010).

[43] L2 Norm. http://mathworld.wolfram.com/L2-Norm.html.

[44] LAKHINA, A., CROVELLA, M., AND DIOT, C. Diagnosing Network-wide Traffic Anomalies. In *Proc. of SIGCOMM* (2004).

[45] LI, X., BIAN, F., CROVELLA, M., DIOT, C., GOVINDAN, R., IANNACCONE, G., AND LAKHINA, A. Detection and Identification of Network Anomalies Using Sketch Subspaces. In *Proc. of IMC* (2006).

[46] LIBERTY, E. Simple and Deterministic Matrix Sketching. In *Proc. of KDD* (2013).

[47] Zeus Botnet Cleanup (Microsoft). http://blogs.technet.com/b/microsoft_blog/archive/2014/06/02/microsoft-helps-fbi-in-gameover-zeus-botnet-cleanup.aspx.

[48] MISLOVE, A., POST, A., GUMMADI, K. P., AND DRUSCHEL, P. Ostra: Leverging Trust to Thwart Unwanted Communication. In *Proc. of NSDI* (2008).

[49] MOHAISEN, A., YUN, A., AND KIM, Y. Measuring the Mixing Time of Social Graphs. Tech. rep., University of Minnesota, 2010.

[50] MOLAVI KAKHKI, A., KLIMAN-SILVER, C., AND MISLOVE, A. Iolaus: Securing Online Content Rating Systems. In *Proc. of WWW* (2013).

[51] MOTOYAMA, M., MCCOY, D., LEVCHENKO, K., SAVAGE, S., AND VOELKER, G. M. Dirty Jobs: The Role of Freelance Labor in Web Service Abuse. In *Proc. of USENIX Security* (2011).

[52] QUERCIA, D., AND HAILES, S. Sybil Attacks Against Mobile Users: Friends and Foes to the Rescue. In *Proc. of INFOCOM* (2010).

[53] RAHMAN, M. S., HUANG, T.-K., MADHYASTHA, H. V., AND FALOUTSOS, M. Efficient and Scalable Socware Detection in Online Social Networks. In *Proc. of USENIX Security* (2012).

[54] RINGBERG, H., SOULE, A., REXFORD, J., AND DIOT, C. Sensitivity of pca for traffic anomaly detection. In *Proc. of SIGMETRICS* (2007).

[55] RUBINSTEIN, B. I. P., NELSON, B., HUANG, L., JOSEPH, A. D., HON LAU, S., RAO, S., TAFT, N., AND TYGAR, J. D. Stealthy Poisoning Attacks on PCA-based Anomaly Detectors. *ACM SIGMETRICS Performance Evaluation Review 37*, 2 (2009).

[56] RUBINSTEIN, B. I. P., NELSON, B., HUANG, L., JOSEPH, A. D., LAU, S.-H., TAFT, N., AND TYGAR, D. Compromising PCA-based Anomaly Detectors for Network-Wide Traffic. Tech. rep., EECS Department, University of California, Berkeley, 2008.

[57] SHARMA, A., AND PALIWAL, K. K. Fast Principal Component Analysis Using Fixed-point Algorithm. *Journal of Pattern Recognition Letters 28*, 10 (2007).

[58] STRINGHINI, G., EGELE, M., KRUEGEL, C., AND VIGNA, G. Poultry Markets: On the Underground Economy of Twitter Followers. In *Proc. of WOSN* (2012).

[59] STRINGHINI, G., WANG, G., EGELE, M., KRUEGEL, C., VIGNA, G., ZHENG, H., AND ZHAO, B. Y. Follow the Green: Growth and Dynamics in Twitter Follower Markets. In *Proc. of IMC* (2013).

[60] TAN, E., GUO, L., CHEN, S., ZHANG, X., AND ZHAO, Y. E. UNIK: Unsupervised Social Network Spam Detection. In *Proc. of CIKM* (2013).

[61] TRAN, N., LI, J., SUBRAMANIAN, L., AND CHOW, S. S. Optimal Sybil-resilient Node Admission Control. In *Proc. of INFOCOM* (2011).

[62] VISWANATH, B., POST, A., GUMMADI, K. P., AND MISLOVE, A. An Analysis of Social Network-based Sybil Defenses. In *Proc. of SIGCOMM* (2010).

[63] WANG, G., KONOLIGE, T., WILSON, C., WANG, X., ZHENG, H., AND ZHAO, B. Y. You Are How You Click: Clickstream Analysis for Sybil Detection. In *Proc. of USENIX Security* (2013).

[64] XIE, Y., YU, F., ACHAN, K., PANIGRAHY, R., HULTEN, G., AND OSIPKOV, I. Spamming Botnets: Signatures and Characteristics. In *Proc. of SIGCOMM* (2008).

[65] YANG, Z., WILSON, C., WANG, X., GAO, T., ZHAO, B. Y., AND DAI, Y. Uncovering Social Network Sybils In the Wild. In *Proc. of IMC* (2011).

[66] YU, H., GIBBONS, P. B., KAMINSKY, M., AND XIAO, F. SybilLimit: A Near-optimal Social Network Defense Against Sybil Attacks. In *Proc. of IEEE S &P* (2008).

[67] YU, H., KAMINSKY, M., GIBBONS, P. B., AND FLAXMAN, A. SybilGuard: Defending Against Sybil Attacks via Social Networks. In *Proc. of SIGCOMM* (2006).