Identifying Compromised Accounts on Social Media Using Statistical Text Analysis

Dominic Seyler  
University of Illinois at Urbana-Champaign  
201 N Goodwin Ave  
Urbana, Illinois, USA 61801  
dseyler2@illinois.edu

Lunan Li  
University of Illinois at Urbana-Champaign  
201 N Goodwin Ave  
Urbana, Illinois, USA 61801  
lunali3@illinois.edu

ChengXiang Zhai  
University of Illinois at Urbana-Champaign  
201 N Goodwin Ave  
Urbana, Illinois, USA 61801  
czhai@illinois.edu

ABSTRACT

Compromised social media accounts are legitimate user accounts that have been hijacked by a third (malicious) party and can cause various kinds of damage. Early detection of such compromised accounts is very important in order to control the damage. In this work we propose a novel general framework for discovering compromised accounts by utilizing statistical text analysis. The framework is built on the observation that users will use language that is measurably different from the language that a hacker (or spammer) would use, when the account is compromised. We use the framework to develop specific algorithms based on language modeling and use the similarity of language models of users and spammers as features in a supervised learning setup to identify compromised accounts. Evaluation results on a large Twitter corpus of over 129 million tweets show promising results of the proposed approach.

1 INTRODUCTION

Account compromising (a.k.a. account hijacking) has become a major issue for public entities and regular users alike. One of the most prominent examples is the hostile takeover of the Associated Press Twitter account in April 2013 [11]. A tweet which claimed an attack on the White House caused a stock market panic and major stock indexes to fall, as well as a wakening of the US-dollar. Also in 2013, another news report found that over a quarter million accounts on Twitter were compromised [9]. Despite massive efforts to contain account hacking, it is still an issue today. More recently, public entities that suffered an attack include ABC News [14], McDonald’s [4], Amnesty International, Forbes and others [16]. Shockingly, the Times magazine reported that Russian hackers were able to compromise a large number of Twitter accounts for the purpose of spreading misinformation in the 2016 U.S. Presidential Election [3]. This incidents are prime examples of the severity of damage that hackers can cause when their messages are backed by the trust that users put into the entity, whose account was compromised. For regular users a compromised account can be a very embarrassing experience, since often questionable posts are published with their identity in social media. As a result, 21% of users that fall victim to an account hack abandon the social media platform [18].

Compromised accounts are legitimate accounts that a malicious party takes control over, with the intention of gaining financial profit [18] or spreading misinformation [5]. These accounts are especially interesting for hackers/spammers, since they can exploit the trust network that the legitimate user has established [5, 15]. For example, this exploitation makes it more likely that users within the personal network of the compromised account click on spam links, since they erroneously believe that their source is legitimate. Finding these hijacked accounts is especially challenging, since they exhibit similar traits than regular accounts [15]. For example, the user’s personal information, friends, likes, etc. are all legitimate at the time of the account takeover. [18] found that the detection of an account takeover can take up to five days, with 60% of the takeovers lasting an entire day. A time window this large will give hackers ample time to reach their goal. Furthermore, simple matching against a blacklist cannot be used in the compromised account detection task. Only after analyzing the changes in the account’s behavior, patterns can be identified that expose the account as being compromised. This task therefore requires methodology that is specific to accounts that have been compromised, as opposed to regular spam accounts.

In this work we introduce a novel framework for compromised account detection that is based on the observation that a regular user’s textual output will differ from a spammer’s textual output. When we further assume that an account is only compromised once 1, we can separate the user’s text from the spammer’s by defining a begin and end of an account takeover. As a specific implementation of the framework, for a user account under consideration, we model the user’s and spammer’s language as two smoothed multinomial probability distributions that are estimated using the textual output of the user and spammer, respectively. We can then use a similarity measure between probability distributions as indicators that an account is being compromised. Even though we do not know the begin and end of an account takeover our method leverages the fact that the average difference for random account begin and end dates will be higher for compromised accounts as for benign accounts. Using features derived from the similarity measure, we can then train a classifier that can reliably detect compromised accounts.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2018 Copyright held by the owner/author(s).

1This assumption is not essential as our general idea is also applicable when an account is compromised more than once.
A major challenge in our evaluation is to find the ground truth in a social media dataset. Finding compromised accounts manually within an enormous dataset is clearly infeasible and there is no dataset publicly available that contains such information. We therefore introduce an approach where we artificially create compromised user accounts, by injecting spam into these accounts. Spam, in our case, is simulated by switching part of a user’s post stream with another random user’s posts. Using this methodology we gain full control over the ground truth and can vary such parameters as amount of accounts compromised and begin/end of an account takeover. To show applicability in a real-world environment, we find evidence that non-artificially compromised accounts can be detected when the algorithm is trained on synthetic data.

2 RELATED WORK

Existing work on compromised user accounts can be divided into two broad categories: Category one are papers that present methods for compromised account detection, which we label as "detection" [5, 6, 15, 19]. The second category strives to deeper investigate characteristics of accounts that are known to be compromised, which we label as "analysis" [8, 13, 18, 23]. The entire amount of works belonging to the detection category is manageable having only three distinct works. We take this as evidence that this problem has great potential for new findings in terms of methodology specific to the problem domain. A similar observation has been made in [1], a recent survey paper, and in [15, 19].

A study that profiled compromised accounts [18], found that hackers gain control of these accounts either directly, by finding user name and password information, or through an application by using a valid OAuth token. In the case where compromised accounts are exploited for financial gains, [18] found three major schemes, namely: (1) The sale of products, e.g., weight loss supplements. (2) The sale of influence in the social network, e.g., a compromised account would advertise the sale of followers and simultaneously it would be used to generate these “fake” followers. (3) Lead generation, where users were tricked into filling surveys.

The work that is closest to our problem setting is [6] and a near-identical version of the paper in terms of methodology [5]. There, the authors learn behavioral profiles of users and look for statistical anomalies in features based on temporal, source, text, topic, user, and URL information. However, compared to our method the textual features are superficial, since they only consider the change in language (e.g., from English to French) of messages. Also, topic features are somewhat primitive, since they only consider message labels, such as hashtags, as topics. Thus, there is a large potential for deeper semantic and/or statistical analysis of the textual content of social media posts to detect these compromised accounts. To the best of our knowledge, there exists no approach that has utilized temporal statistical characteristics of language for compromised account detection.

The major contribution of [19] is a novel evaluation framework, where user accounts are artificially compromised by switching out part of two users’ tweet streams. This allows for complete control over the ground truth and thereby potentially enabling the generation unlimited training and testing data. In addition to the proposed evaluation framework, the authors investigate reasons for the sparseness of research on compromised account detection as compared to fake account detection. They further improve the method proposed in [6], such that it can detect the hijacking event of an account earlier.

The work [15] performs and extensive study of behavioral patterns using click-stream data on Facebook. In general, it is hard for academics to obtain such click data, since they are usually only accessible for the social media provides themselves or through a full-fledged user study. From the observations made in the behavioral study, the authors derive features that model public (e.g., posting a photo) and private (e.g., searching for user profiles) behavior of users. To find compromised accounts the authors look at different click-streams associated with a single user and calculate feature variance. An account is considered compromised if the difference of a new click-stream does not fall within the variance estimated on the user’s click-stream history. Here as well, the authors use data (i.e., click-streams) of other users to simulate an compromised account. Also, their method seems to be tailored to social media platforms that are similar to Facebook. In contrast, our method is general enough that it can theoretically be applied to any social media platform that uses text (e.g., Twitter, Facebook, LinkedIn, etc.)

Another work that was shown to be applicable to compromised account detection is [20]. There, the authors use principal component analysis (PCA) to characterize normal user behavior, which allows them to distinguish it from abnormal behavior. Their features do not consider text and it is noticeable that compared to other detection categories, their approach performed worst on compromised account detection. An interesting contribution is the generation of ground-truth data for detecting compromised accounts. There, they reverse-engineer a Trojan that can perform certain actions on a user’s social media account incognito. They then monitor actions that were send by a server that controls the Trojan and flag the accounts that posted a suspicious message as compromised.

The work that is closest to ours methodologically is [12]. There, the authors use language models estimated on a tweet containing a URL, the URL page’s content and other tweets related to trending topics using KL-Divergence. The focus of this work is to find spam tweets in isolation and therefore the applicability to compromised account detection is not obvious. A further shortcoming of the approach is that it is limited to tweets that contain URLs.

Our work shares a similar goal with work on privacy protection in information retrieval (e.g., [2, 7, 17, 25], and work in a recent workshop on the topic [21]) but with a different focus on detecting compromised user accounts rather than protecting the private information reflected in user’s search queries.

3 PROBLEM DEFINITION

Before discussing our framework in detail we define the problem formally. Let $U$ be the set of all users. Further, let $m_u \in u \in U$ be a tweet $m$ from user $u$ at time $t$. Our goal is to find all compromised user accounts $U^{\text{comp}} \subset U$, where a compromised account for user $u$ is denoted as $u^{\text{comp}} \in U^{\text{comp}}$.

For classification we turn the compromised account detection task into a binary classification problem. Here, the goal is to decide
We now turn to describe the proposed framework for identifying
repeated for a certain user, the higher is the sampling rate. Higher
repeated multiple times for different values of \( \theta \) using
any similarity measure \( t \)

to divide the tweet space of a user into two non-overlapping sets
incidents.

mised only once, which covers the majority of account hijacking
assumption our framework makes is that an account is compro-
based on the assumption that a spammer’s textual output will de-

4.1 A General Framework

and scalability. In our work we decided to treat this problem as a
for datasets greater than 100,000 users.

re-trained only periodically, the computational cost is still enormous
computationally infeasible for large datasets. Even if the classifier is
classifier for all

system point-of-view. However, this would require to re-train the
or benign. This certainly represents the best case from an interactive

There are definitely trade-offs between desirable practicability
and scalability. In our work we decided to treat this problem as a
binary classification problem that is decided on a per-user basis. We
argue that this case delivers the best balance between practicability
and scalability.

4 APPROACH

4.1 A General Framework

We now turn to describe the proposed framework for identifying
compromised accounts. As mentioned before, the framework is
based on the assumption that a spammer’s textual output will de-
viately significantly from a regular user’s textual output. Another
assumption our framework makes is that an account is compro-
mised only once, which covers the majority of account hijacking
incidents.

To capture this discrepancy between language usage, we propose
to divide the tweet space of a user into two non-overlapping sets \( M^{user} \) and \( M^{spam} \). We randomly assign two timepoints: \( t_{start} \) signals the start of the account takeover and \( t_{end} \) signals its end. All \( m_t \) with \( t \in [t_{start}, t_{end}] \) make up \( M^{spam} \). All \( m_t \) with \( t \in [t_{0}, t_{start} - 1] \cup t \in [t_{end} + 1, t_{-1}] \) make up \( M^{user} \).

Now we can measure the difference between \( M^{user} \) and \( M^{spam} \) using any similarity measure of our choice. This procedure can be
repeated multiple times for different values of \( t_{start} \) and \( t_{end} \). \( t \) can be of different granularity, where the minimum is per-post and the
maximum can be chosen at will. The more often the procedure is
repeated for a certain user, the higher is the sampling rate. Higher
sampling rates make for better approximations of the true difference
between \( M^{user} \) and \( M^{spam} \). This strategy thus provides a flexible
tradeoff between accuracy and efficiency.

The similarity measures can then be employed as features in the
downstream task. For example, if the similarity for a certain user
passes a threshold \( \epsilon \). This threshold can be learned in the training
phase, such that the number of classified compromised accounts
that match the labeled compromised accounts is maximized. Nat-
urally, any other useful features (if available) can also be further
added to improve accuracy.

4.2 Instantiation with Language Modeling

In what follows, we describe our practical instantiation of the frame-
work using language modeling and supervised learning. We create
a classifier that can distinguish compromised from benign user
accounts based on a set of features derived from our framework.
We measure this as the difference between two word probability
distributions of a user and a spammer. We select KL-divergence [10]
as our method of choice to compare the difference of probability
distributions.

More specifically, we assume that when a user writes a tweet she
draws words from a probabilistic distribution that is significantly
different from the distribution a spammer draws words from. Let \( \theta^{user} \) and \( \theta^{spam} \) be two word probability distributions (i.e., lan-
guage models) for the user and spammer, respectively. We therefore
need to select two time points \( t_{start} \) and \( t_{end} \) that mark beginning
and end of the account takeover by a spammer. As described by
our framework, all messages \( m_t \) that fall within the time inter-
val \([t_{start}, t_{end}]\) will contribute to the spammer’s language model
\( \theta^{spam} \), whereas \( m_t \notin [t_{start}, t_{end}] \) will contribute to \( \theta^{user} \).

Figure 1 presents an overview of our approach for distinguish
compromised from benign accounts. For a particular user, we sam-
ple her tweet stream for random \( t_{start}, t_{end} \) pairs. As seen in
the figure, the algorithm can, for example, select \( t_2 \) and \( t_3 \) as
\( t_{start} \) and \( t_{end} \), respectively. Thus, all tweets that fall between
\([t_0, t_1], [t_2, t_3]\) contribute to \( \theta^{user} \). All tweets that fall in \([t_2, t_3]\)
contribute to \( \theta^{spam} \). Then, the KL-divergence for these specific
\( \theta^{user} \) and \( \theta^{spam} \) contributes one sample for this user. This pro-
cess is repeated for different sample rates (i.e., 5, 10, 25, 50), where
each time \( t_{start} \) and \( t_{end} \) are selected at random, with constraint
\( t_{start} < t_{end} \). The sample rates are selected based on a feasibility
analysis that we describe in Section 5. As features for our classi-
fier we select the maximum, minimum, mean and variance of the
sampled KL-divergence scores. These features are then combined
in a logistic regression model, which learns the optimal weighting
for each feature based on the training data. The classifier is then
evaluated on held-out data to measure its performance (Section 6).

4.3 Language Modeling Details

When dealing with language models, we try to estimate the joint
probability of \( P(w_1, w_2, w_3, \ldots, w_N) \) for all words \( w_1, \ldots, w_N \) in
the text. According to the chain-rule, this is equivalent to comput-
ing \( P(w_1)P(w_2|w_1)P(w_3|w_2)\ldotsP(w_N|w_1, \ldots, w_{N-1}) \). Because of
the combinational explosion of word sequences and the extensive
amount of data needed to estimate such a model, it is common to
use n-gram language models. N-gram language models are based
on the Markov assumption that a word $w_1$ only depends on $n$ previous words ($P(w_k|w_1, ..., w_{k-1}) \approx P(w_k|w_{k-n+1}, ..., w_{k-1})$). The simplest and computationally least expensive case is the uni-gram. Here, the Markov assumption is that a word $w_k$ is independent of any previous word, i.e., $P(w_k|w_1, ..., w_{k-1}) \approx P(w_k)$.

All $P(w_k)$ for $k = 1, 2, ..., N$ make up a language model $\theta$, which is a multinomial probability distribution where each word in the document is an event in the probability space. The parameters for the language models $\theta$ are estimated using maximum-likelihood. It can be shown that maximizing the likelihood of the uni-gram language model is equivalent to counting the number of occurrences of $w_k$ and dividing by the total word count ($P(w_k) = \frac{c(w_k)}{N}$, where $c(w_k)$ is the word count of $w_k$). The proof is out of the scope of this work and therefore omitted.

This distance between two probability distributions can be estimated using Kullback-Leibler-divergence [10]. In the discrete case, for two multinomial probability distributions $P$ and $Q$ the KL-divergence is given in Equation 1. From the equation it can be observed that $D_{KL}(P, Q) \neq D_{KL}(Q, P)$. However, it is common practice to still think of $D_{KL}$ as a distance measure between two probability distributions [24].

$$D_{KL}(P, Q) = \sum_i P(i) \log \left( \frac{P(i)}{Q(i)} \right)$$  

Another issue with $D_{KL}$ is that the sum runs over $i$, which is the event space of $P$ and $Q$. Thus, it requires the event space to be equivalent for both distributions. In our case $\theta$ is a uni-gram language model (i.e. a multinomial probability distribution). Now, let $v(\theta)$ denote the vocabulary set of $\theta$. As a result of maximum-likelihood estimation, in most cases $v(\theta^{user}) \neq v(\theta^{spam})$. Thus, we have to smooth the probability distributions such that $v(\theta^{user}) = v(\theta^{spam})$, which is required in order to calculate $D_{KL}$. To achieve this, we define $v(\theta) = v(\theta^{user}) \cup v(\theta^{spam})$. Then, we set $v(\theta^{user}) = v(\theta)$ and $v(\theta^{spam}) = v(\theta)$ and estimate each multinomial distribution of $\theta^{user}$ and $\theta^{spam}$ using the Laplace estimate.

5 EXPERIMENTAL DESIGN

This section discusses five research questions that we answer experimentally. We first perform a feasibility analysis to find evidence that (I) compromised user accounts do exhibit higher difference in language models compared to benign accounts. To show this we calculate KL-divergence for all possible combinations of $t_{begin}$ and $t_{end}$ for a subset of users and manually investigate the characteristics of these accounts.

The second part of our feasibility analysis finds proof that (II) the average KL-divergence can be estimated by randomly sampling a certain number of points with different begin/end dates. For this we leverage the same data as in our first feasibility study and plot, for different sample rates, the estimated average KL-divergence and mean squared error, compared to the actual averages.

Following the feasibility study, we show (III) how effective our method is in detecting compromised accounts in a simulated environment. We perform a metrical evaluation where we train and test our algorithm on synthetic data and use standard metrics (i.e. Accuracy, Precision, Recall, $F_1$-score) for performance measure. We further investigate (IV) how long it takes for our method to classify an account as compromised. For this we plot the percentages of compromised accounts that where correctly identified against an ascending number of injected tweets. This is done as to approximate the number of tweets it takes for the classifier to make an accurate prediction.

The final question we answer is (V) how effective is our method on a real (non-synthetic) dataset. For this purpose we perform an empirical evaluation, where we train our algorithm on synthetic data and evaluate on the original data. Since in this case there are no classification labels available, we perform a manual evaluation. We will now first introduce our dataset, followed by the experimental design, followed by presentation and discussion of the results and ending in an error analysis.

5.1 Dataset

As mentioned earlier, since there are no public datasets available that are applicable to our task, we decided to simulate account hijackings by injecting spam into regular user accounts. For simulation we leverage a large Twitter corpus of roughly 467 million posts from 20 million users covering a seven month period from [22]. The dataset only contains textual, temporal and user information. Relationships between users, tweets, etc. are unknown.

Since this dataset was created for a different purpose, it has no information about which Twitter accounts are actually compromised. Thus, a major challenge is to create a gold standard which labels accounts as compromised. Finding these accounts manually within a dataset of 20 million users is clearly infeasible. On the other hand, if we use an existing approach to label the ground truth automatically we would assume this other method to be perfect. A fair comparison to our approach would be rendered impossible.

Simulating account hijackings enables us not only to create a gold standard but also to have full control over such parameters as amount of accounts compromised and begin/end of account takeover. For dataset creation we follow an approach inspired by [19], where part of the tweets of two user accounts are swapped to artificially create an compromised account. [19] argues that these artificially compromised accounts are often harder to detect than actually compromised accounts, since these posts do not necessarily contain keywords or URLs that can be usually found in spam posts.

In [19] the date for an account takeover is chosen at random but always ends with the last tweet of the account. We go one step further and choose the beginning and end at random, meanwhile ensuring that only a certain pre-defined fraction of the tweets is compromised. This allows us to test the effectiveness of our method in different scenarios where different percentages of tweets are compromised within an account. Our algorithm for dataset creation works as outlined in Algorithm 1. Variables prob_comp and percentage_comp fall in range [0, 1] and specify how many user accounts are compromised and what percentage of tweets is compromised in a user account, respectively.

5.2 Feasibility Studies

In our feasibility studies we use a subset of the data by selecting 495 users at random, where each compromised account contains 50% compromised tweets. For each user we put all tweets into daily
buckets and calculate the KL-divergence for all possible combinations of $t_{\text{begin}}$ and $t_{\text{end}}$. The reason this cannot be done for the whole dataset is simply because of computational cost. We therefore operate on this subset to investigate whether our method is feasible and can be approximated to avoid increased computational effort.

5.3 Metrical Evaluation

Our metrical evaluation aims to show the performance of the algorithm in numerical terms, meaning by using standard metrics for performance measure. For testing how well the classifier performs, we need binary labels that indicate whether an account is compromised. Since this kind of dataset is not publicly available we decided to simulate the account hacking. Using this methodology we can generally generate as much training and testing data as needed.

For our experiments we utilize the dataset we created as described in Section 5.1. To get meaningful estimates for the language models $\theta_{\text{user}}^e$ and $\theta_{\text{spam}}^e$ we select the 100,000 users with most tweets in our data. Our resulting set of users have a maximum of 86,593 and a minimum of 571 tweets. Furthermore, we filter users with less than 10 days of coverage, resulting in our final dataset of 99,697 users with 129,442,756 tweets in total.

In our first study we investigate the performance of our classifier using various metrics. We report a confusion matrix with details on class-based performance. Furthermore, we measure Precision, Recall and $F_1$-score for the class representing the predictions of compromised accounts (1-labels). For training we select 67% of user accounts and for testing 33% of user accounts at random. In this experiment 50% of the tweets of a compromised account are compromised. We also perform an ablation study where features are tested in ensemble and in isolation.

In addition, we show the effectiveness of our method for different levels of difficulty in our second study. We experiment with various settings for the percentage of compromised tweets in compromised accounts with random begin and end dates of account take over. More concretely, we select 50%, 25%, 10% and 5% of tweets to be compromised, each representing a more difficult scenario. Further, we employ 10-fold cross validation to get the most representative results, ensuring that each data point has been utilized for testing. In both experiments the probability of an account being compromised is set to 0.5 to obtain a balanced dataset.

As mentioned before we perform a third study, with the goal to approximate how long it takes for our method to classify an account as compromised. Here we combine all accounts with varying percentages of tweets compromised (i.e. 50%, 25%, 10% and 5%) and group them by the actual number of tweets that where injected. We then plot the percentages of compromised accounts that where correctly identified for 0 to 600 injected tweets with bucket sizes of 50.

5.4 Empirical Evaluation

Our empirical evaluation aims to show that once the algorithm is trained on artificially compromised accounts, it can detect compromised accounts in the original dataset. To achieve this we manually evaluate the accounts that have the highest probability according to our classifier, which was trained on the synthetic data.

For our experiments, the dataset is similar as described in the previous section. However, one major change is that we only inject spam into the training dataset. We randomly select 70% of users for training and 30% of users for testing. In this experiment 25% of tweets of a compromised account in the training dataset are compromised.

Since there is no ground truth for accounts in the testing dataset, we define four specific metrics to evaluate whether an account is considered as compromised or not. The four metrics are listed as following: (1) The account has a specific and sharp topic change in certain time periods. (2) The account has a specific tweet posting frequency change in certain time periods. (3) The account has a specific language change in certain time periods. (4) The account posts highly repeated tweets or similar contents in certain time periods. If two or more of those metrics are met, an account is considered as being compromised.
6 EXPERIMENT RESULTS

6.1 Language Model Similarity of Compromised Accounts

In the first part of our feasibility analysis we answer research question (I) and examine whether any significant differences in KL-divergence between regular (benign) and compromised user accounts exist.

To achieve this we manually inspect the differences in user accounts by leveraging a heatmap as a visual cue. Figure 2 depicts four heatmaps of two benign (Figure 2a and 2b) and compromised user accounts (Figure 2c and 2d). The x-axis of each figure depicts different values for $t_{\text{begin}}$, while the y-axis depicts different values for $t_{\text{end}}$. The color palette ranges from blue (low KL-divergence) to red (high KL-divergence). The left part of the plot below the diagonal is intentionally left empty, since these values would represent nonsensical variable assignments for $t_{\text{begin}}$ and $t_{\text{end}}$.

It is immediately obvious that we find more high KL-divergence values for compromised compared to benign accounts. In the figures high values are expressed by red and dark red colors. From this observation it can be inferred that the average KL-divergence will be higher for accounts that are compromised. By manually inspecting over 100 of these user account heatmaps we find that many of them follow this general trend. We understand this as preliminary evidence that a method which utilizes KL-divergence for detecting compromised accounts is feasible.

We further noticed in Figures 2c and 2d that the account takeover happened where the KL-divergence reaches its maximum (see the dark circle in Figure 2c and tip of the pyramid in Figure 2d). Unfortunately, this is not true for all inspected accounts, but it hints that there might be potential to find the most likely period of an account takeover with our current framework. This could be done by finding $t_{\text{begin}}$ and $t_{\text{end}}$ that maximizes the difference of the language models $\theta_{\text{user}}$ and $\theta_{\text{spam}}$.

6.2 Estimate KL-divergence Using Random Sampling

In our second feasibility analysis we investigate research question (II), whether the average KL-divergence of a user account can be approximated using random sampling. More specifically, we try to find a reasonable estimate by calculating the KL-divergences only for a subset of the $t_{\text{begin}}$ and $t_{\text{end}}$ pairs.

Since we have calculated the KL-divergence for every possible combination of $t_{\text{begin}}$ and $t_{\text{end}}$ for all of the 495 users, we know the actual average KL-divergence. We then try different sampling rates. For every sampling rate we average over the samples and compare them to the actual average that is calculated using all $t_{\text{begin}}$, $t_{\text{end}}$ pairs.

The result is shown in Figure 3. In the figure we plot the actual average KL-divergence for compromised (comp) and benign against the averaged samples (sample) for different sample rates. The sample rates range from 1 to 311. The left y-axis shows the averaged KL-divergence. The plot shows that the average KL-divergence for compromised accounts is about 0.1 higher than for benign accounts. This confirms our findings in Section 6.1. Furthermore we see that for small sample rates (< 121) there are minimal deviations for the average (±0.01). The higher the sample rate the lower these deviations become, as our estimate gets better.

Since our estimates only deviate slightly we also investigate the mean squared error (mse) for each sampling rate. Here, the mse is defined as:

$$\frac{1}{n} \sum_{u \in U_{\text{test}}} (\text{sampled}_\text{avg}(u) - \text{actual}_\text{avg}(u))^2$$

In Figure 3 we plot the mse for compromised and benign accounts. The values for mse are plotted on the right y-axis. For very small sampling rates (< 50) we see errors of over 0.07 and over 0.06 for compromised and benign accounts, respectively. Once the sample rate is greater than 120 the mse is close to 0. We therefore conclude that a sampling rate in the range of [50, 120] is sufficient for our experiments.

6.3 Effectiveness on Synthetic Data

In this subsection we show (III) how effective our method is in detecting compromised accounts in a simulated environment and investigate (IV) how long it takes for our method to classify an account as compromised.

In Figure 4 we present a confusion matrix that measures performance of our classifier on held-out data on a 67%/33% train/test dataset split. From the matrix we can see that the classifier performs slightly better in classifying benign accounts. In total, 13,308 compromised accounts are identified correctly and 3,222 compromised accounts remain misclassified.

Table 1 presents the ablation study with different evaluation measures for the same data. It shows the performance of all features in ensemble and individually. We observe that we gain maximum performance over all metrics when all features are utilized (see column “Ensemble”). For both classes the classifier reaches an accuracy of 0.85. When focusing on the class of compromised accounts only, the classifier achieves high Precision of 0.88, with slightly lower Recall at 0.80. Both measures are combined in the $F_1$-score, which reaches 0.84.

When features are investigated individually, we find that the features “Max” and “Variance” perform equally low, with “Variance” having better recall. It is noteworthy that for both Recall is higher than Precision, even though for the ensemble the opposite holds true. This might be an indicator that these features contribute meaningful signals to the overall model, that improve Recall overall. The features “Min” and “Mean” also perform almost equally in isolation and much better than the other two. We find it surprising that the minimum sample of the two probability distribution is such a strong signal. We hypothesize that this might be due to the fact that the minimum will be higher for compromised accounts when the sample rate is small and most $t_{\text{start}}$ and $t_{\text{end}}$ dates fall within the actual range of the begin and end of an account hijacking.

The results for our next experiment can be observed in Table 2. As expected, best performance is reached when 50% of tweets in a compromised account are compromised, with accuracy at 0.85. When the number of compromised tweets is reduced by half the accuracy drops by 0.1 to 0.75. When the amount of compromised tweets is set to 10% and 5% accuracy further reduced to 0.63 and 0.57, respectively. In all scenarios a small confidence interval (±0.01) is observed, showing that each fold barely deviates from the average.
Figure 2: KL-divergence heatmap for different benign (Figure 2a and 2b) and compromised user accounts (Figure 2c and 2d). The x-axis of each figure depicts different values for $t_{\text{begin}}$, while the y-axis depicts different values for $t_{\text{end}}$. The color palette ranges from blue (low KL-divergence) to red (high KL-divergence).

| Measure | Ensemble | Max | Min | Mean | Variance |
|---------|----------|-----|-----|------|----------|
| Accuracy | 0.85 | 0.58 | 0.77 | 0.75 | 0.56 |
| Precision | 0.88 | 0.58 | 0.82 | 0.77 | 0.55 |
| Recall | 0.80 | 0.59 | 0.70 | 0.70 | 0.70 |
| $F_1$ | 0.84 | 0.59 | 0.75 | 0.73 | 0.61 |

Table 1: Ablation study using different measures of classifier performance on held-out data with 67%/33% train/test split. Features are tested in ensemble and individually.

Table 2: Ten-fold cross validation results for different percentages of compromised tweets in an account, measured in accuracy with 95% confidence interval.

| Compromised % | Accuracy |
|---------------|----------|
| 50            | 0.85(±0.01) |
| 25            | 0.75(±0.01) |
| 10            | 0.63(±0.01) |
| 5             | 0.57(±0.01) |

To answer research question (IV), how long it takes for our method to classify an account as compromised, we approximate the number of tweets it takes for the classifier to make an accurate prediction. In Figure 5 we plot the percentage of correctly identified compromised accounts against an ascending number of injected tweets. We can see that for a small number of tweets [0, 49], our method identifies about 42% of compromised accounts correctly.

Then, a general upwards trend in performance is observable until the maximum is reached at 600 tweets at about 76%. From this we can conclude that our method is less effective for a small amount of injected tweets, which makes it less suitable for real-time detection tasks. It is, however, more effective for accounts that carry a modest to large amount of spam tweets, which makes our method more suitable for retrospective analysis of the data.
Figure 3: Results for different sample rates in interval [1, 311]. The left y-axis shows the averaged KL-divergence, the right y-axis shows the mean squared error (mse).

Figure 4: Confusion matrix for evaluation on held-out data. Compromised accounts have class label 1, benign accounts have class label 0.

6.4 Effectiveness on Real Data

The final question we answer experimentally is (V), how effective is our method on a real (non-synthetic) dataset. In our first step we categorize $k = 20$ accounts with the highest probability of being compromised into six categories (news, advertisement/spam, re-tweet bot, compromised, regular users, unknown). The results can be found in Table 3. We find that most of these accounts belong to categories with high variation in language, i.e., news, advertisement/spam, re-tweet bot. One of the accounts is found to be compromised. The algorithm also misclassified seven user accounts as being compromised. One account was categorized as "unknown", since it was posting in a language that none of the evaluators speaks.

| Category          | Count |
|-------------------|-------|
| News              | 5     |
| Advertisement/Spam| 4     |
| Re-tweet Bot      | 2     |
| Compromised       | 1     |
| Regular User      | 7     |
| Unknown           | 1     |

Table 3: Category assignment of manually evaluated accounts.

| Account Status   | Count |
|------------------|-------|
| Abandoned        | 7     |
| Active           | 6     |
| Deleted          | 4     |
| Tweets protected | 2     |
| Suspended        | 1     |

Table 4: Current account status of manually evaluated accounts (Dec 2017).

These results show that our algorithm can also be used to detect "unusual" accounts and users, thus potentially enabling development of novel text mining algorithms for analyzing user behavior on social media, which should be a very interesting future research topic.

We further inspect the current (as of December 2017) state of each account. We found a total of five different states (i.e., abandoned, active, deleted, tweets protected and suspended). Table 4 lists the results. We find that most accounts are abandoned, which is not surprising given the age of the dataset. Noteworthy is also that one of the accounts was suspended by Twitter. The account that was identified as compromised was set to tweets protected, which could be an indicator that said user had become more conscious about tweets that were posted from her account and therefore decided to not share her tweets publicly.

6.5 Compromised Account Profiling

We now further investigate the compromised account and discuss the details which made us believe that a compromise had happened. While manually investigating the account’s tweets they where mostly discussing celebrities, movies and music, with very few links in any messages. At some point in the tweet stream a near-duplicate message is posted hundreds of times with only brief pauses between tweets. The pattern that all tweets had in common is shown in Figure 6. Here, @[username] were different users, that were most likely followers of the account. With this scheme the hacker tries to directly grab the attention of a targeted user. The placeholder [link] was one of two links that were identified as suspicious by the link-shortening service the hacker utilized to hide the actual URL. Unfortunately, both links do not exist anymore. We therefore cannot provide additional details about the target of these links. After this “attack” of messages, the tweets return to a similar pattern as before.

From our standpoint we therefore conclude that this account was compromised for the following reasons: (1) The account was discussing different topics before the attack. (2) The account was posting few links in messages. (3) A near-duplicate message with
Figure 5: Percentages of compromised accounts that were correctly identified (y-axis) for 1 to 600 injected tweets with bucket sizes of 50 (x-axis).

**Pattern:** @[username] as discuss [sic] earlier heres [sic] the link, another service that will pay $ for your tweets, anyone can join [link]

**Profit scheme:** Lead generation

**Current status:** Tweets protected

Figure 6: Profiling of compromised account.

Suspicious intend and links was repeatedly posted hundreds of times at a certain point. (4) These messages stop and the account continues with regular activity. From the content of this messages we conclude that the hacker was pursuing a led generation scheme [18], where users are lured into clicking a link that will offer some sort of payment for their information. It is reasonable to assume that if our algorithm were applied at much larger scale to all the Twitter users, it would most likely be able to detect many more compromised accounts.

### 6.6 Error Analysis

From Table 3, we can see that five news accounts, four advertisement/spam accounts and two re-tweet bots are misclassified as compromised accounts. We meet this problem because our method uses a similarity function, in this case the KL-divergence, as the measurement to determine whether an account is being compromised or not. One common characteristic of news accounts and advertisement accounts is that they rarely use the same words or sentence patterns because there are different news and products everyday. Thus, the high language diversity in these accounts results in the KL-divergence between any two time periods to be high, which leads to the misclassification.

A potential way to improve the method would be to order the samples chronologically instead of treating them as a set. An algorithm could then look for extreme changes in KL-divergence, which should mostly occur in compromised accounts and less in news or spam accounts, where the KL-divergence is constantly high.

The regular user accounts do not exhibit any obvious pattern for the misclassification. We found that some of these misclassified accounts do exhibit one or more of the following characteristics that could potentially influence the KL-divergence samples: (1) excessive usage of re-tweets (2) lots of messages with distinct links (3) lots of repeated messages (4) many misspellings of the same word. Further exploration of this is another very interesting future research direction.

7 DISCUSSION AND FUTURE WORK

We proposed a novel general framework based on statistical text analysis for detecting compromised social media accounts. Following the framework, we proposed a specific instantiation based on uni-gram language models and KL-divergence measure, and designed features accordingly for use in a classifier that can distinguish compromised from benign accounts. We address the challenge in evaluating this task by developing a method to create a synthetic dataset with compromised user accounts, where we inject spam into the artificially hacked user accounts. Our metrical and empirical experiments on a large Twitter corpus validated the efficacy of our method, showing that the algorithm can detect compromised accounts with up to 85% accuracy when trained and tested on synthetic data. Moreover, we find evidence that our method can identify "real" compromised accounts when trained on synthetic and tested on the original data.

Our work makes a novel connection between security and information retrieval and opens up many interesting new research directions where we can potentially apply many information retrieval and text mining techniques to address the security challenges in social media. Below we briefly discuss some of the interesting future research directions enabled by our work:

**Ranking user accounts:** Instead of framing the problem as binary classification, we can aslo frame it as ranking user accounts in descending order of the likelihood of being compromised and study how to develop such ranking algorithms. Such a ranked list could be passed to human experts to sequentially examine and determine whether there are compromised account within this list. This would allow us to create a pool of accounts with relevance labels, e.g. [0 := not compromised, 1 := shows weak indication of compromise, 2 := shows strong indication of compromise, 3 := is compromised without doubt] that mirror the experts opinion about the severity of indicators that an account is compromised. These labels can then be utilized as a test collection to compare the ranking accuracy of different compromised account detection approaches.
More features for classification: The classifier can easily use many other features that we can construct based on text analysis. For example, one can look at the whole stream of the tweets for a user and examine the time series of term frequencies for each term to see if there’s any interruption. Further, it would be advantageous to incorporate features that utilize the social graph. One could, for example, incorporate the relationship of friends into a model, or make use of other social network-based statistics, such as the number of re-tweets. Although we didn’t explore these directions, the flexibility and extensibility of the proposed framework would allow us to plug in more features to potentially further improve the accuracy.

Deeper semantic analysis: A potential limitation of our approach is the relatively simple modeling of language. Extending our approach could be envisioned by adding a notion of topics to the user model. Each user could then be represented as a mixture of topics for different time points. In this scenario, another signal for an account takeover would be if a user deviates significantly from her topic cluster. In the same context, our uni-gram model does not take word ambiguity or word similarity into account. Therefore, a user model that contains many ambiguous or semantically close words might be more accurate if these words are grouped together.

One concept that appears useful for this task might be word embeddings. On the other hand, reducing the granularity of the user’s character model may lead to the solution of word ambiguity or word similarity. Therefore, a user model that contains many ambiguous or semantically close words might be more accurate if these words are grouped together.

More features for classification: The classifier can easily use many other features that we can construct based on text analysis. For example, one can look at the whole stream of the tweets for a user and examine the time series of term frequencies for each term to see if there’s any interruption. Further, it would be advantageous to incorporate features that utilize the social graph. One could, for example, incorporate the relationship of friends into a model, or make use of other social network-based statistics, such as the number of re-tweets. Although we didn’t explore these directions, the flexibility and extensibility of the proposed framework would allow us to plug in more features to potentially further improve the accuracy.

Improve evaluation: When creating compromised user accounts artificially, our approach uses tweets of another legitimate user account. [19] argues that this makes the problem harder than necessary, since the language of two users might by less different than that of a user and a spammer. We therefore propose to inject real spam messages into the user accounts instead of tweets of regular users. One could also think of a generative method that is trained on regular user’s vocabulary to drastically might result in the model becoming too general and therefore failing to accurately model the individual user’s characteristics.

Forensic Analysis: For detailed forensic analysis it might be required to find the most likely period of an account takeover. We realized that there is a potential of doing so with our current framework. We believe that this could be achieved by choosing $t_{start}$ and $t_{end}$ such that they maximize the difference between the user’s and spammer’s language. Finding the maximum efficiently is a hard problem, since it’s computationally expensive to compute the difference for all $t_{start}$ and $t_{end}$ pairs. Therefore, an approximation algorithm would be required.

REFERENCES

[1] Kayode Sakariyiah Adewole, Nor Badril Anuar, Amirrudin Kamein, Kastunt Dewi Varathan, and Syed Abdul Razak. 2017. Malicious accounts: dark of the social networks. Journal of Network and Computer Applications 79 (2017), 41–67.

[2] Avi Arampatzis, Pavlos S. El fragidis, and George Drosatos. 2011. Enhancing Deniability Against Query-logs. In Proceedings of the 33rd European Conference on Advances in Information Retrieval (ECIR’11). Springer-Verlag, Berlin, Heidelberg, 117–128. http://dl.acm.org/citation.cfm?id=1996889.1996906

[3] Massimo Calabresi. 2017. Inside Russia’s Social Media War on America. (May 2017). http://time.com/4748952/inside-russia-social-media-war-america/

[4] Chris Ciacia. 2017. McDonald’s Twitter account hacked, blasts Trump. (March 2017). http://www.foxnews.com/tech/2017/03/16/mcdonalds-twitter-account-blasts-trump.html

[5] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2017. Towards Detecting Compromised Accounts on Social Networks. IEEE Trans. Dependable Sec. Comput. 14, 4 (2017), 447–460.

[6] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2013. COMPA: Detecting Compromised Accounts on Social Networks. In 20th Annual Network and Distributed System Security Symposium, NDSS.

[7] Yi Fang, Archana Godavarthy, and Haibing Lu. 2016. A Utility Maximization Framework for Privacy Preservation of User Generated Content. In Proceedings of the 2015 ACM International Conference on the Theory of Information Retrieval (ICTIR ’15). ACM, New York, NY, USA, 281–290. https://doi.org/10.1145/2970398.2970417

[8] Chris Grier, Kurt Thomas, Vern Paxson, and Chao Michael Zhang. 2010. @spam: the underground on 140 characters or less. In Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS, 27–37.

[9] Heather Kelly. 2013. Twitter hacked; 250,000 accounts affected. (February 2013). http://www.cnn.com/2013/02/01/tech/social-media/twitter-hacked

[10] S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. Ann. Math. Statist. 22, 1 (03 1951), 79–86.

[11] Edmund Lee. 2013. AP Twitter Account Hacked in Market-Moving Attack. (April 2013). https://www.bloomberg.com/news/articles/2013-04-23/dow-jones-drops-recovery-after-false-report-on-ap-twitter-page

[12] Juan Martinez-Romo and Lourdes Araujo. 2013. Detecting malicious tweets in trending topics using a statistical analysis of language. Expert Syst. Appl. 40, 8 (2013), 2992–3000.

[13] Benjamin Murauer, Eva Zangerle, and Günther Specht. 2017. A Peer-Based Approach on Analyzing Hacked Twitter Accounts. In 50th Hawaii International Conference on System Sciences, HICSS.

[14] Andrea Park. 2017. ABC News, “Good Morning America” Twitter accounts hacked, praise Trump. (March 2017). http://multimedia.cbs.com/news/abc-news-good-morning-america-twitter-accounts-hacked-praise-trump/

[15] Xin Ruan, Zhenyu Wu, Haining Wang, and Sushil Jajodia. 2016. Profiling Online Behaviors for Compromised Account Detection. IEEE Trans. Information Forensics and Security 11, 1 (2016), 176–187.

[16] John Russell. 2017. Prominent Twitter accounts compromised after third-party app Twitter Counter hacked. (March 2017). https://techcrunch.com/2017/03/15/twitter-counter-hacked/

[17] Xiaohua Shen, Bin Tan, and ChengXiang Zhai. 2007. Privacy Protection in Personalized Search. SIGIR Forum 41, 1 (June 2007), 4–17. https://doi.org/10.1145/1273221.1273222

[18] Kurt Thomas, Frank Li, Chris Grier, and Vern Paxson. 2014. Consequences of Connectivity: Characterizing Account Hijacking on Twitter. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security. 489–500.

[19] David Trang, Fredrik Johansson, and Magnus Rosell. 2015. Evaluating Algorithms for Detection of Compromised Social Media User Accounts. In ENIC. IEEE Computer Society. 75–82.

[20] Bimal Viswanath, Muhammad Ahmad Bashir, Mark Crovella, Saikat Guha, Krishena P. Gummadi, Balachander Krishnamurthy, and Alan Mislove. 2014. Towards Detecting Anomalous User Behavior in Online Social Networks. In Proceedings of the 23rd USENIX Security Symposium. 223–238.

[21] Hui Yang, Ian Soboroff, Li Xiong, Charles L.A. Clarke, and Simon L. Garfinkel. 2016. Privacy-Preserving IR 2016: Differential Privacy, Search, and Social Media. In Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’16). ACM, New York, NY, USA, 1247–1248. https://doi.org/10.1145/2911451.2917763

[22] Jaewon Yang and Jure Leskovec. 2011. Patterns of temporal variation in online media. In Proceedings of the Forth International Conference on Web Search and Web Data Mining. WSDM, Irwin King, Wolfgang Nejdl, and Hang Li (Eds.). ACM. 177–186.

[23] Eva Zangerle and Günther Specht. 2014. “Sorry, I was hacked”: a classification of compromised twitter accounts. In Symposium on Applied Computing, SAC. 587–593.

[24] ChengXiang Zhai and Sean Massung. 2016. Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining. Morgan & Claypool.

[25] Simon Zhang and Grace Hui Yang. 2017. Deriving Differentially Private Session Logs for Query Suggestion. In Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR ’17). ACM, New York, NY, USA, 51–58. https://doi.org/10.1145/3121850.3120706