10 Bits of Surprise: Detecting Malicious Users with Minimum Information

Reza Zafarani
Department of EECS
Syracuse University
reza@zafarani.net

Huan Liu
Computer Science and Engineering
Arizona State University
huan.liu@asu.edu

ABSTRACT

Malicious users are a threat to many sites and defending against them demands innovative countermeasures. When malicious users join sites, they provide limited information about themselves. With this limited information, sites can find it difficult to distinguish between a malicious user and a normal user. In this study, we develop a methodology that identifies malicious users with limited information. As information provided by malicious users can vary, the proposed methodology utilizes **minimum information** to identify malicious users. It is shown that as little as **10 bits** of information can help greatly in this challenging task. The experiments results verify that this methodology is effective in identifying malicious users in the realistic scenario of limited information availability.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*

Keywords

Malicious User Detection; Minimum Information; Information Verification; Behavior Analysis

1. INTRODUCTION

Social media sites are inundated with malicious users. In June 2012, Facebook reported that 8.7% – or 83 million – of its user accounts are fake [35]; that is roughly the size of Egypt's population and larger than the population of 230 countries in the world [37]. Facebook also reported that of that 8.7%, **1.5%** are "undesirable" accounts that are created for malicious purposes [35]. Twitter faces similar challenges. In its security filings, Twitter claims that 5% of its users are fake [10]; however, researchers estimate the percentage of its fake accounts to be as high as 10% [10]. These fake accounts are mostly sold on black market for as low as $0.05 and are used for malicious activities [10].

CIKM'15, October 19–23, 2015, Melbourne, Australia.

http://dx.doi.org/10.1145/2806416.2806535.

Malicious accounts may be created for different purposes. According to Cao et al. [4], some malicious accounts are created for profitable activities, such as click fraud, identity fraud, and malware distribution. Others are created for social purposes such as pranks, stalking, cyberbullying, or identity concealing. Online service providers find detecting and subsequently, suspending malicious accounts vital in order to protect their normal users against external threats.

Detecting malicious accounts dates back to the onset of social media. Comprehensive feature-based techniques, human-in-the-loop approaches, or techniques that use social-graphs are devised (see a review in Section 2). These techniques assume that a good amount of information about malicious users has been gathered. This information includes (1) the content that malicious users generate, (2) the users they befriend, or (3) the malicious activities they have exhibited. In short, their *content*, *links*, or *activities*. However, malicious users often do not have an incentive to generate content or befriend others and detecting them after they have exhibited malicious activities is not useful. Hence, malicious users need to be detected using their often limited content or link (i.e., friends) information. Even the limited information that malicious users provide can vary. Therefore, to detect malicious users with different levels of information, one needs to be able to detect malicious users even when there is **minimum information** available.

In this paper, we aim to detect malicious users with minimum information. We make the following contributions:

1. We introduce the first methodology to detect malicious users with minimum information. This methodology can be used as the first line of combat against malicious users on the Web.
2. We identify five general characteristics of malicious users and demonstrate how one can identify these characteristics in user-generated content online.
3. We demonstrate that with as little as **10 bits** of information, one can distinguish between normal and malicious users.
4. We show via experiments that the methodology is robust and at least as effective as techniques that have access to more information.

In Section 2, we review the malicious user detection literature. We formally define the malicious user detection problem with minimum information in Section 3. We detail characteristics of malicious users and how one can identify such characteristics in user content in Section 4. We detail our experiments in Section 5. Finally, we conclude in Section 6 with directions for future work.

2. LITERATURE ON MALICIOUS USER DETECTION

While detecting malicious users with minimum information is unexplored, identifying malicious users in general is not a new topic. Often, to identify malicious users, (1) feature-based techniques, (2) human-in-the-loop techniques, or (3) techniques that use social graphs are used. We review representative techniques for each category and discuss how the current work relates to these techniques.

I. Feature-based Techniques. In feature-based methods, different features are constructed to capture the behavior of the malicious user. These features are then used to construct a dataset that is trained by a supervised learning framework. For instance, Xie et al. [39], develop the *AutoRe* framework that identifies botnet campaigns. Their framework identifies traffic that is (1) bursty and (2) distributed. These features of traffic help identify botnets. The bursty and distributed nature of unwanted content is also used in detecting malicious posts on Facebook [12]. Wang [34] introduces a method that detects spam on Twitter using network features such as the number of followers or friends and content features such as duplicated tweets. Feature-based techniques have been discussed extensively for detecting unwanted content in social tagging systems [20, 24], social networks [29], email [21], online videos [3], and microblogging sites [2, 43]. Our work differs from the existing work in two aspects. First, current techniques for identifying malicious users often employ content or link information. Thus, one often needs a large collection of data instances to obtain guaranteed performances. Our approach employs minimum information across sites. Second, current literature is often context-dependent (e.g., site specific). Our method employs the minimum information that is universally available across sites and is robust even when information is collected from multiple sites.

II. Human-in-the-loop Methods. One approach of identifying malicious users is to employ human experts. Humans can naturally identify malicious users by their activities. Alternatively, one can combat malicious activities by technologies such as CAPTCHAs [33] or photo-based authentications [4] that are only solvable by humans. Although specific attacks are proposed for human-in-the-loop methods [26,41], they are in general considered effective. Unfortunately, verifying accounts by humans is time consuming. For example, Tuenti, a Spain-based social networking service, hires humans to investigate reported users and block malicious ones [4]. An employee can only process 250 to 300 reports an hour from the daily 12,000 reports received. This issue makes human-in-the-loop processes infeasible for large-scale networks. Our approach in this paper is automatic and can easily scale to billions of users.

III. Social Graph-based Techniques. In social graph-based methods, the information about the links (i.e., friendships) that the malicious individual has created helps detect the malicious user. For instance, Yang and colleagues [42] detect more than 100,000 fake accounts using social network features on RenRen social network. In particular, they find that invitation frequency, outgoing requests accepted, incoming requests accepted, and network clustering coefficient can help identify fake accounts. In other works, probabilistic, combinatorial, or random walk models have been applied to network information to identify malicious users. Examples include, *Sybilguard* [45], *Gatekeeper* [31], *SybilInfer* [7], *SumUp* [30], and *Sybillimit* [44]. These methods or variants can be applied on sites such as Twitter to identify malicious users [13]. Mislove et al. [32] show that most techniques in this area function by finding local communities around trusted nodes. Assuming the existence of a social graph is a strong assumption. One often requires specific privacy permission to obtain such graphs and in specific cases, this graph is not available. In cases where there is no social graph, our methodology is still easily applicable.

3. MALICIOUS USER DETECTION WITH MINIMUM INFORMATION

Who is a malicious user? The definition varies in the literature from users that harass other users to users that jeopardize the privacy of others [4]. We consider malicious users on a site, those whom normal users consider malicious. Clearly, the opinion of normal users can be subjective and has to be verified by experts. In section 5.1, we demonstrate how such human-verified data can be collected. Humans are known to be accurate in detecting malicious users on social media [15, 16, 27]. However, as discussed in our literature review, human-in-the-loop approaches are time consuming and expensive for large-scale networks. Hence, by investigating how humans detect malicious users, one can not only scale detection of malicious users, but can also protect against a wide spectrum of malicious activities that humans are able to detect on social media [4, 5].

Our goal in this paper is to identify such malicious users. Malicious users often provide little or no information. Hence, a method that can be universally employed on different sites is constrained to use the minimum information available on all sites. *Usernames* are the minimum information available on all social media sites [46]. Often, usernames are alphanumeric strings or email addresses, without which users cannot join sites. Because of their unique characteristics, usernames are shown to be surprisingly effective for identifying individuals [46]. We formalize our problem using usernames as the minimum information available on all sites. Other content or link information such as user profile information or friends, when added to usernames, should help better identify malicious individuals. However, the lack of consistency in the availability of such information on all social media sites, directs us toward formulating our problem with usernames.

When using usernames, the goal is to detect malicious users from their usernames. Hence, one can learn a function $\mathbf{M}(.)$ that given a username u, predicts whether the username belongs to a malicious user or not. We denote the $\mathbf{M}$ function as the *malicious user detection* function. Formally,

Definition. ***Malicious User Detection.*** *Given a username u, a malicious user detection method attempts to learn a malicious user detection function $\mathbf{M}(.)$ such that*

$$\mathbf{M}(u) = \begin{cases} 1 & \textit{If } u \textit{ belongs to a malicious user;} \\ 0 & \textit{Otherwise.} \end{cases}$$

Malicious users have distinctive characteristics. These characteristics leave traces in the usernames of malicious users. These traces can be captured using data features. Following the common machine learning and data mining practice, the malicious user detection function can be learned

using a supervised learning algorithm that utilizes these features and *labeled data*. In our problem, labeled data includes usernames that are known to be malicious or normal.

Supervised learning can be performed using classification or regression. Depending on the malicious user detection task at hand, one can even learn the probability that a username is malicious, generalizing our binary $\mathbf{M}$ function to a probabilistic one ($\mathbf{M}(u) = p$). This probability can help select the most likely malicious username. The learning of the malicious user detection function is the most straightforward. Therefore, we next elaborate on different characteristics of malicious users and how features can be constructed to capture traces introduced in usernames due to these characteristics. Note that the designed features may or may not help in the learning framework and are included as as long as they could be computed from usernames. Later on in Section 5, we will analyze the effectiveness of all features, and if it is necessary to find as many features as possible.

In summary, to detect malicious users, we (1) identify characteristics of malicious users, (2) construct features to identify traces of these characteristics in usernames, and (3) train a learning model to detect malicious users. Due to the interdependent nature of these user characteristics and feature construction, we discuss them together next.

4. CHARACTERISTICS OF MALICIOUS USERS

Humans detect malicious users on social media by their characteristics. By reviewing related literature from computer science, security, criminology, among other fields [4, 11, 29, 32, 39, 42], we identified five general characteristics of malicious users. Malicious users can have one (or a combination) of these characteristics. As researchers identify more characteristics of malicious users, our methodology can be extended with these characteristics and the corresponding features that can capture traces left by them in usernames.

4.1 Malicious Users are Complex and Diverse

Malicious users often generate (1) *complex* and (2) *diverse* information to ensure their anonymity.

I. Complexity. To measure complexity of usernames, it is natural to borrow techniques from complexity theory. We employ Kolmogorov complexity [22] to determine the complexity of a username. Kolmogorov complexity of a username is defined as the length of the shortest program capable of reproducing the username on a universal computer such as a Turing Machine. Hence, Kolmogorov complexity is the minimum quantity of information required to reproduce the username and measures its complexity.

For username u, let $K(u)$ denote its Kolmogorov complexity. While $K(u)$ defines the complexity of username u, it is well-known that its exact value cannot be computed [19]. Nonetheless, the following theorem helps compute the expected Kolmogorov complexity. Assume that usernames such as u are distributed as a random variable U, called *username space*, with probability distribution P.

THEOREM 1. *(from Li and Vitányi [22]) For random variable U, expected Kolmogorov complexity $\mathbb{E}(K(U))$ equals its entropy $H(U)$, plus a constant term that depends only on P.*

Hence, by computing the entropy of the username space, one can approximate the expected Kolmogorov complexity in usernames. However, the theorem discusses the entropy of the username space and it is not clear how one can connect this theorem to a specific username. For connecting properties of specific usernames to the entropy of the username space, we can use the concept of *information surprise* [6].

For username u, let $p(u)$ denote the probability of observing u. Information surprise for u is defined as

$$I(u) = -\log_2(p(u)). \tag{1}$$

Hence, for a rare username u with a small observation probability $p(u)$, information surprise $I(u)$ is much higher than that of a common username with a higher observation probability. It is well-known that information surprise is deeply connected to entropy:

THEOREM 2. *(from Cover and Thomas [6]) The expected information surprise $\mathbb{E}(I(U))$ for username space U (i.e., random variable U) is equivalent to its entropy $H(U)$.*

So, by combining Theorems 1 and 2, one can approximate the expected Kolmogorov complexity of usernames by computing the expected information surprise in them. The information surprise for a username u is computed by measuring $I(u) = -\log_2(p(u))$, which requires the probability of observing username u. The probability of observing username u, denoted in characters as $u = c_1c_2 \dots c_n$, is

$$p(u) = \Pi_{i=1}^{n} p(c_i|c_1c_2 \dots c_{i-1}). \tag{2}$$

We approximate this probability using an n-gram model,

$$p(u) \approx \Pi_{i=1}^{n} p(c_i|c_{i-(n-1)} \dots c_{i-1}). \tag{3}$$

Often, to denote the beginning and the end of a word special symbols are added such as $\star$ and $\bullet$. So, for username `sara`, the probability approximated using a 2-gram model is

$$p(sara) \approx p(s|\star)p(a|s)p(r|a)p(a|r)p(\bullet|a). \tag{4}$$

To estimate the probability of a username using an n-gram model, one needs to compute the probability of its comprising n-grams. The probability of these n-grams can be computed using a large set of usernames. For that, we use a dataset of 158 million Facebook usernames (later discussed in Section 5.1) to train a 6-gram model. This n-gram model was employed to compute the probability of a username and in turn, its information surprise.

Being able to compute information surprise, we obtain its values for a set of 33 million usernames. The set contains both normal and malicious users. The process followed to collect these usernames is later discussed in Section 5.1. For both normal and malicious users, we estimate the empirical probability density function using Kaplan-Meier estimate. Figure 1 plots the empirical probability density function of information surprise values for normal and malicious users.

The thick solid line in Figure 1 demonstrates the distribution of surprise values for normal usernames and the thin solid line depicts the distribution for malicious usernames. As shown in the figure, malicious usernames are more complex with the expected information surprise (i.e., expected Kolmogorov complexity) value of 23.11 bits and more diverse, ranging from 4.14 bits to 232.64 bits.

Unlike malicious usernames, normal usernames are less surprising and more concentrated around a mean value, with a mean of 12.49 bits and the information surprise value ranging from 3.90 to 31.93 bits. The figure shows that these

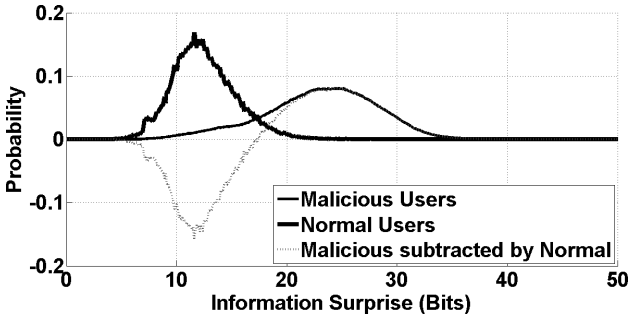

Figure 1: Probability Density Function for Information Surprise Values of Malicious and Normal Users.

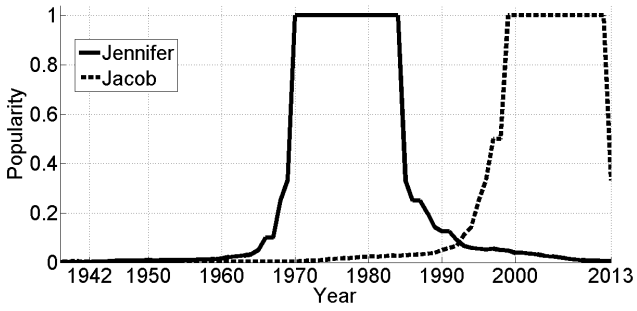

Figure 2: Popularity of names: *Jennifer* and *Jacob* over time. Higher values shows more popularity.

distributions are well separated indicating that by using the information surprise of a username, one might be able to accurately classify usernames into malicious or normal.

In Figure 1, the dashed line depicts the curve for the malicious usernames subtracted by the curve for the the normal usernames. Hence, when this gray line is above zero, it shows that for a specific information surprise value, the username is more likely to be malicious and whenever the gray line is below zero, we observe the opposite. We notice that for values between 3.91 and 17.96 the curve is below the zero line, showing usernames are more likely to be normal. In this range, the mean value is 10.9 bits. Thus, when the information surprise for a username is approximately 10 bits, the username is more likely to be normal. The title of this paper is inspired by this observation.

Hence, we include the information surprise of the username (i.e., its complexity) as another feature in our dataset.

II. Diversity. To create diverse information, malicious users often generate usernames that include digits. Therefore, we include the number of digits in the username as a feature. We also include the proportion of digits in the username as another feature in our feature set.

4.2 Malicious Users are Demographically Biased

In the criminology literature [11], it is well-known that crime correlates with demographic information. Thus, one expects to better detect malicious users by determining their demographics. Following the *diffusion of innovations* terminology [23], a malicious user has internal demographic attributes, external demographic attributes, or a combination of internal and external (i.e., mixed) attributes.

Internal attributes are endogenous attributes that the user has no control over such as his or her age. External attributes are attributes due to the environment that the malicious user lives in such as the language that the malicious user speaks. The level of knowledge that the malicious user has is an example of a user attribute that is mixed (internal+external). This is because it depends on both the environment that the malicious user lives in and on the internal attributes of a user such as his interests. To concretely profile a malicious user, one has to consider all these attributes. We select gender from internal attributes, language from external attributes, and knowledge (i.e., vocabulary size) from mixed demographic attributes to be predicted from usernames. Clearly, with more internal/external/mixed demographic attributes, one should better profile malicious users. We leave that as a future direction for this work. But, how can we detect gender, language, or other attributes of individuals from their usernames?

Psychological studies [14] show that users leave traces of their personal information and attributes in the information they generate such as their usernames. For example, our analysis of popular names by birth year of US social security records[1] since 1879 shows us that the frequency of different names change over time. For instance, in Figure 2, we depict the popularity of first names: *Jennifer* and *Jacob* over time. For each year, the popularity of each name is shown on a scale of [0,1], 1 being the most popular name and 0 being the least popular. Jennifer was the most popular female name between [1970-1984] whereas Jacob has been the most popular male name for [1991-2012]. Similar patterns can be observed for different English and non-English names given the diversity of the US population. Hence, given a name, one can estimate the most likely age. Names, interests, as well as other personal attributes are often abbreviated or used in usernames [46]. We use these information traces in usernames to predict gender, language, among other attributes.

I. Malicious User Gender. To predict gender from usernames, we train a classifier. The classifier decomposes a username into character n-grams and estimates the gender likelihood based on these n-grams. This classifier is trained on the n-grams of a labeled dataset of usernames, in which the gender for each username is known. We collect our labeled dataset from Facebook. Our labeled dataset contains a set of 4 millions usernames with their corresponding gender. The classifier predicts the gender of a username with up to 80% accuracy. Notice that because malicious users tend to hide their identity and gender; instead of the actual prediction, we include the classifier's confidence in the predicted gender as the feature.

II. Malicious User Language. To detect the language of the username, we train an n-gram statistical language detector [9] over the European Parliament Proceedings Parallel Corpus[2], which consists of text in 21 European languages (*Bulgarian, Czech, Danish, German, Greek, English, Spanish, Estonian, Finnish, French, Hungarian, Italian, Lithuanian, Latvian, Dutch, Polish, Portuguese, Romanian, Slovak, Slovene,* and *Swedish*) from 1996-2006 with more than 40 million words per language. The trained model detects the username's language, which is a feature in our feature

[1]http://www.ssa.gov/oact/babynames/
[2]http://www.statmt.org/europarl/

set. The *detected language* feature is limited to European languages. Our language detector will not detect other languages. The language detector is also challenged when dealing with words that may not follow the statistical patterns of a language, such as location names, etc. This issue can be tackled by including the distribution of alphabet letters in usernames as features [46]. Thus, in addition to predicted language, we include the alphabet distribution of the username as a feature.

III. Malicious User Knowledge. To approximate the level of knowledge of a malicious user, we can compute his or her vocabulary size. The vocabulary size can be computed by counting the number of words in a large dictionary that are substrings of the username [46]. This approach captures different possible interpretations of the username and approximates the level of knowledge of the malicious user. We include the vocabulary size as a feature. As this is a rough approximation, we will determine the efficiency of this feature in our feature importance analysis in Section 5.5.

4.3 Malicious Users are Anonymous

Malicious activity often requires a level of anonymity [1]. Theoretically, the maximum level of anonymity is achieved when a username has the maximum entropy [40]. We compute the entropy of the alphabet distribution of the username as well as the normalized entropy of the username to measure its level of anonymity. To normalize entropy, we divide it by $\log n$, where n is the number of unique alphabet letters used in the username. In addition, we also measure the uniqueness of letters in the username – that is, the number of unique letters used in the username divided by the username length. We include entropy, normalized entropy, and uniqueness as features.

4.4 Malicious Users are Similar

Malicious users tend to be similar. For instance, individuals marketing an illegal product `Dangerous-Pill` all share the name of the product `Dangerous-Pill` in their marketing content. This malicious content similarity can be captured in usernames by identifying specific (1) language patterns and (2) words in the usernames.

4.4.1 Language Patterns

To find finer grain language patterns of users, we employ character-level n-grams. Character-level n-grams have shown to be effective in detecting unwanted content [17, 18] and connecting users across social media sites [28]. We compute the normalized character-level bigrams of usernames and include them as features. Bigram features are normalized using TF-IDF. Bigrams allow for a language-agnostic solution [46] that can detect common patterns of malicious users conveniently.

For coarser grain language patterns, we investigate common habits of malicious users. For instance, it is known that the use of digits is an indication of unwanted content [20]. In particular, we notice that malicious users tend to start their usernames with digits; therefore, we include the number of digits at the beginning of the username as a feature. We also notice that malicious users repeat character letters more often that normal users. This strategy allows them to circumvent widely used statistical malware blockers [38]. Hence, we include the maximum number of times a letter has been repeated in the username as another feature.

4.4.2 Word Patterns

A well-known approach to identify malicious users or content is by finding specific keywords in the content generated by these users. Hence, we denote the existence of these specific keywords in usernames as an indication of malicious activity. We utilize two dictionaries, one containing keywords related to malicious activities and the other for offensive keywords[3] For each dictionary, we count the number of words in the dictionary that appear as the substring of the username. We include these two counts for the aforementioned two dictionaries as features.

4.5 Malicious Users are Efficient

In contrast with complex malicious users (Section 4.1), some malicious users demand efficiency. This is because the malicious user is interested in performing the malicious activity frequently, quickly, and at large-scale. For instance, when performing click-fraud, the malicious user is interested in creating many accounts, each clicking on specific ads. This efficiency can be observed in usernames in terms of (1) the username length; and (2) the number of unique alphabet letters in usernames. We include both as features. In addition, we can observe efficiency by determining the typing patterns of the malicious user.

Most people use one of the two well-known DVORAK and QWERTY keyboards, or slight variants such as QWERTZ or AZERTY [36]. It has been shown that the keyboard layout significantly impacts how random usernames are selected [8]. For example, `qwer1234` and `aoeusnth` are two well-known passwords commonly selected by QWERTY and DVORAK users, respectively. To model typing patterns of malicious users, for each username we construct the following 15 features for each keyboard layout (a total of 30 for both keyboard layouts),

1. (1 feature) The percentage of keys typed using the *same hand* that was used for the previous key. The higher this percentage the less users had to change hands for typing.
2. (1 feature) The percentage of keys typed using the *same finger* that was used for the previous key.
3. (8 features) The percentage of keys typed using each finger. Thumbs are not included.
4. (4 features) The percentage of keys pressed on rows: Top Row, Home Row, Bottom Row, and Number Row. Space bar is not included.
5. (1 feature) The approximate *distance* (in meters) traveled for typing a username. Normal typing keys are assumed to be $(1.8\text{cm})^2$ (including gap between keys).

We construct $15 \times 2 = 30$ features that capture the typing patterns of usernames for both keyboards and include them in our feature set.

We have detailed how characteristics of malicious users can be captured by meaningful features. These features help identify traces of malicious activities in usernames. Overall, for each username, we construct 1,413 features.

Clearly, not all characteristics of malicious users are covered by our features, and with more theories on characteristics of malicious users, more features can be constructed. We will empirically study if it is necessary to use all features and the effect of using different features on learning performance of detecting malicious users.

[3]All data available at: http://reza.zafarani.net/data/10bits

Following our approach, we compute the feature values over labeled data, and verify the effectiveness of our methodology by learning the malicious user detection function. Next, experiments for evaluating our methodology are detailed.

5. EXPERIMENTS

We evaluate our methodology to detect malicious users in this section. First, we verify if our proposed approach can identify malicious users well. Next, we verify if different learning algorithms can influence the prediction task. Then, we determine the sensitivity of our approach to different conditions. Finally, we perform feature importance analysis and determine how features designed for each characteristic of malicious users influence the detection outcome. Before we present the experiment details, we detail how experimental data was collected for this research.

5.1 Data Preparation

Our approach to detect malicious users employs a supervised learning framework. Hence, labeled data is required. This labeled data consists of usernames and their corresponding label: malicious or normal. To construct this labeled data and for our experiments, we collect four datasets.

I. Malicious Users (negative examples). We collect malicious usernames from sites such as dronebl.org, ahbl.org, among others (for a complete list see [25]). These sites gather lists of usernames that have been reported by other normal users for malicious purposes. Once reported, these accounts are manually verified by domain owners to be malicious. These lists are published to help sites promote their security. We collect a set of 32 million usernames that are manually reported as malicious by users on different sites. This set forms our negative examples.

II. Normal Users (positive examples). For collecting normal users, we require users that are manually labeled as normal. For that, we refer to Twitter verified accounts, all manually verified by Twitter employees. These accounts are all followed by the Twitter handle `verified`[4]. By crawling all the users this account follows, we collect a set of 45,953 usernames guaranteed to be normal. These usernames form our positive examples.

III. Facebook Users (positive+negative examples). To diversify the types of usernames we have collected, we also collect a set of 158 million usernames from Facebook, that is, 1 in 9 Facebook users in the world are included in our dataset. Note that the Facebook dataset is not completely normal as Facebook expects around 1.5% to be malicious. We employ this dataset for analyzing the sensitivity of our approach to different conditions (Section 5.4.2).

IV. Gender Dataset. We collect a different set of 4 million Facebook usernames for which we have the gender information. This dataset was used to train our gender prediction classifier in Section 4.2 to predict gender from usernames.

After collecting these four datasets of usernames[5], we compute the corresponding 1,413 features for all datasets and employ them in our experiments.

[4]http://twitter.com/verified

[5]We ensure that the alphabet used in both sets of usernames match. To avoid site-enforced specific patterns on how usernames should be created, we filter out usernames that are not in ASCII or alphanumeric. Our experiments show that this procedure does not influence our results.

Table 1: Malicious User Detection Performance.

Technique	AUC	F1
Our Approach	0.9932	0.9644
Baseline b_1: Keyword Detection	0.51	0.66
Baseline b_2: Username Randomness	0.70	≈ 0
Baseline b_3: Letter Repetition	0.61	≈ 0
Reference Point r_1: *Markines et al.* [24]	0.984	0.983
Reference Point r_2: *Gao et al.* [12]	0.945	–
Reference Point r_3: *Wang* [34]	0.917	0.917

5.2 Learning the Malicious User Detection Function

Once the negative and positive examples are prepared, learning the malicious user detection function can be achieved by training a classifier. Because our collected negative examples are more, we subsample the negative examples to have the same size as the positive examples. This way we create a dataset that has 50% positive examples and 50% negative ones. Using this dataset, we train a classifier. The random prediction on this dataset cannot achieve more than 50% accuracy. We train an ℓ_2-Regularized Logistic Regression using 10-fold cross validation and obtain an accuracy of 96.42%, an AUC of 0.9932, and an F1-measure of 0.9644.

As there are no comparable methods, we evaluate the effectiveness of our approach by devising three baseline methods for comparison. When individuals are asked to detect malicious users based on their usernames, they often look for specific "keywords", verify if the username looks "random", or look for "repetition of letters". Hence, they form our three baselines b_1, b_2, and b_3:

- **Baseline b_1: Keyword Detection.** We consider a username malicious if it contains a specific keyword. We use the same set of keywords used in Section 4.4.2 and train a classifier based on the single feature. b_1 results in an AUC of 0.5140 and F1-measure of 0.66.
- **Baseline b_2: Username Randomness.** For finding username randomness, b_2 uses the entropy of the username as a feature. Using our data labels, we perform logistic regression. b_2 achieves an AUC of 0.700 and F1-measure of ≈ 0.
- **Baseline b_3: Letter Repetition.** Similar to the procedure followed in baseline b_2, in b_3, we use the maximum number of times a letter is repeated in the username as a feature and train a logistic regression model using our data labels. b_3 achieves an AUC of 0.61 and an F1-measure of ≈ 0.

While the baseline performances demonstrate the difficulty of our problem, the proposed approach outperforms all baselines by at least 41%. The performance for our approach, and baselines are summarized in Table 1. As reference points, we also include in the table the performance of recent state-of-the-art techniques for detecting malicious users. These techniques have access to more information compared to our methodology and do not employ usernames; therefore, no improvement percentage will be reported. Our approach, with usernames only, outperforms these techniques. Next, we investigate if different learning algorithms can further improve the learning performance.

Table 2: Malicious User Detection Performance for Different Classification Techniques.

Technique	AUC	Accuracy
ℓ_2-Regularized ℓ_1-Loss SVM	0.9966	97.05%
ℓ_2-Regularized ℓ_2-Loss SVM	0.9913	96.05%
ℓ_2-Regularized Logistic Regression	0.9923	96.25%
ℓ_1-Regularized Logistic Regression	0.9971	**97.26**%

5.3 Choice of Learning Algorithm

To evaluate the choice of learning algorithm, we perform the classification task using a range of learning algorithms and 10-fold cross validation. The AUCs and accuracy rates are available in Table 2. These algorithms have different learning biases, and one expects to observe different performances for the same task. While we observe a slight increase in the performance, as shown in the table, results are not significantly different across algorithms. This shows that when sufficient information is available in features, the performance is not sensitive to the choice of learning algorithm.

In our experiments, ℓ_1-Regularized Logistic Regression is shown to be the most accurate method; therefore, we use it in the following experiments as the method of choice.

In our previous experiments, we assumed that there is no class imbalance between malicious and normal users. In reality this distribution is skewed. Furthermore, unlike our malicious users, all of our normal users are from one source (Twitter). Thus, we need to verify how this decision influences our results. We analyze the sensitivity of our approach to class imbalance and the distribution of normal users next.

5.4 Sensitivity Analysis

5.4.1 *Sensitivity to Class Imbalance*

In real-world networks such as Facebook and Twitter, the percentage of malicious users in the population is approximated to be at most 10% [10, 35]. In other words, for every 9 normal users there exists at most 1 malicious user. This rate could be different across networks. Thus, we perform a sensitivity analysis with respect to different ratios of malicious users. We construct datasets, where α percent of the dataset consists of malicious users and change α in the range $5 \leq \alpha \leq 50$. Values larger than 50 were not selected, because then we are assuming that malicious users are more than the normal ones.

Because we collected more negative examples, we sample the negative examples many times to guarantee that each negative example is seen at least once. Thus, for each α, many datasets are created. For each one of these datasets, we perform classification and average the performance metrics over all datasets created for a specific α.

Figure 3 depicts the average performance (accuracy, AUC, and F1-measure) of our methodology with different percentages of malicious users. As shown in the Figure, as the number of malicious users increase, AUC remains stable and F1-measure and accuracy slightly drop, but in all cases, all measures stay above 0.97.

5.4.2 *Sensitivity to the Distribution of Normal Users*

To determine the sensitivity of our classifier to different normal users, we use samples of Facebook users instead of our normal users in training. Facebook users are not completely normal and Facebook approximates that around 1.5% of its users are malicious [35]. If our classifier can detect Facebook malicious users, some positive instances (Facebook users) will be classified as negative (malicious). Hence, the performance is expected to slightly decrease. In theory, for all datasets that have at most 50% negative examples (malicious users) and Facebook users as positive examples, one expects at most a $50\% \times 1.5\% = 0.0075$ decrease in accuracy. Our experiments verify this expectation. Figure 4 plots the performance (accuracy, AUC, F1-measure) of the algorithm with different percentages of malicious users and using Facebook users as positive examples. We notice a slight drop in performance for all measures, but the performance remains high and is never below 0.9671. For comparison, we include the performance measures with the original normal users. Comparing the performance measures with those of normal users, we notice that the accuracy drops by at most 0.0035 (less than the expected maximum: 0.0075), AUC drops by at most 0.0012, and F1-measure drops by at most 0.0036.

In our experiments, we employ all 1,413 features to detect malicious users. Designing 1,413 features and computing their values is computationally expensive. Hence, we empirically determine whether all features are necessary next.

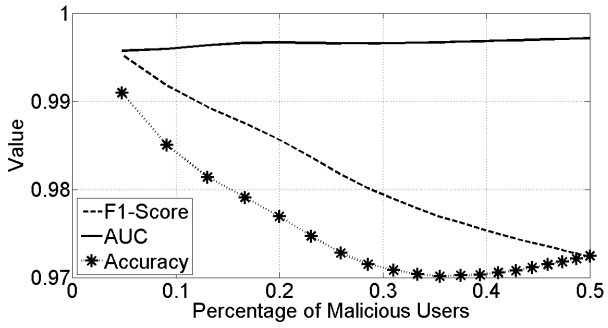

Figure 3: Performance (AUC, F1, and Accuracy) of our methodology for Different Percentages of Malicious Users.

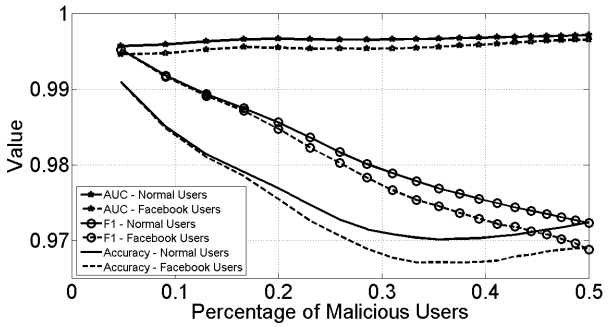

Figure 4: Performance Measures (F1, AUC, and Accuracy) of our Methodology for Different Percentages of Malicious Users when Facebook Identities were used instead of Normal Users.

5.5 Feature Importance Analysis

Here, we analyze how important different features are in learning the malicious user detection function. In other

Table 3: Malicious User Detection Performance for Different Groups of Features.

Feature Groups	AUC	Accuracy
Complexity-based	0.8032	83.16%
Demographic-based	0.9342	86.78%
Anonymity-based	0.7219	63.26%
Similarity-based	**0.9933**	**95.86%**
Efficiency-based	0.9299	87.19%

words, we find features that contribute the most to the classification task. This can be performed by standard feature selection measures such as Information Gain, χ^2, among others. Here, we use the χ^2 statistic to find the top features. The top 10 features in decreasing order of importance are:

1. The information surprise of the username
2. The number of digits used in the username.
3. The percentage of keys pressed on the top row of a QWERTY keyboard when typing the username.
4. The percentage of keys pressed on the top row of a DVORAK keyboard when typing the username.
5. The proportion of digits used in the username.
6. The approximate distance (in meters) traveled for typing a username with a DVORAK keyboard.
7. The percentage of keys pressed on the home row of QWERTY keyboard when typing the username.
8. The approximate distance (in meters) traveled for typing a username with a QWERTY keyboard.
9. The percentage of keys pressed on the bottom row of a DVORAK keyboard when typing the username.
10. Entropy of the username.

We notice that the complexity of the username is the most important feature and that 6 of the top 10 features are features that capture typing patterns. Using only these 10 features, we trained a logistic regression model and achieved an accuracy of 92.95% and an AUC of 0.973.

We also determine groups of features that contribute most to the classification. We divide features into groups based on the characteristic of malicious users they represent. We denote these features based on the discussion in Section 4 as (1) Complexity-based, (2) Demographic-based, (3) Anonymity-based, (4) Similarity-based, and (5) Efficiency-based. Table 3 summarizes the classification performance obtained using only these groups of features.

We observe that similarity-based features work the best and anonymity-based features are least effective. Note that similarity-based features are in general hard to construct as they require n-gram constructions. Surprisingly, efficiency-based or complexity-based features that are easier to compute, can classify malicious users accurately, with up to 87% accuracy. Our observations in this section allows users with limited time and resources to take informed decisions on the features and groups of features to construct.

6. CONCLUSIONS AND FUTURE WORK

In this research, we have introduced a methodology that can identify malicious users with minimum information. Our methodology looks into different characteristics of malicious users and systematically constructs features that can capture traces of malicious behaviors. With new theories on characteristics of malicious users, new features can be introduced into our methodology.

We categorize characteristics of malicious users into 5 general categories. In particular, malicious users can be (1) complex and diverse, (2) demographically biased, (3) anonymous, (4) self-similar, and (5) efficient. A malicious user can exhibit one or a combination of these characteristics. By introducing comprehensive features across these five categories, we train a learning framework that can detect malicious users. The evaluation of this framework demonstrates the effectiveness of this systematic approach.

We notice some interesting observations. First, we notice that usernames that carry approximately 10 bits of information surprise, are more likely owned by normal users. Second, with only minimum information, one can achieve an accuracy of 97%, an AUC of 0.9971, and robust performances with different class imbalances and irrespective of the learning algorithm. Finally, we identify that in case of limited time or resources, one can implement a limited set of features and obtain reasonable accuracy rates.

The findings in this paper have many implications. First, we note that our methodology is in general easy to implement with minimum dependency on the availability of information. Second, our methodology works with usernames from different sites. This is empirically shown in our experiments with usernames collected from a variety of sites. Finally, our methodology performs with reasonable accuracy, compared to state-of-the-art techniques that have access to additional information.

Future work of this research includes integrating additional information available across sites in a principled manner. However, this extension requires considering the heterogeneity of data available across sites. In addition, similar to the observation we had regarding the information surprise values of usernames, we are interested in how surprise values change for other content generated by users.

Acknowledgments. The authors would like to thank Fred Morstatter, Shamanth Kumar, and Ashwin Rajadesingan for valuable suggestions. This work was supported, in part, by the Office of Naval Research grant: N000141410095.

7. REFERENCES

[1] Anne Barron. Understanding spam: A macro-textual analysis. *Journal of Pragmatics*, 38(6):880–904, 2006.

[2] Fabrıcio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgılio Almeida. Detecting spammers on twitter. In *CEAS*, volume 6, page 12, 2010.

[3] Fabricio Benevenuto, Tiago Rodrigues, Virgilio Almeida, Jussara Almeida, Chao Zhang, and Keith Ross. Identifying video spammers in online social networks. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, pages 45–52. ACM, 2008.

[4] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *NSDI*, 2012.

[5] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. Who is tweeting on twitter: human, bot, or cyborg? In *ACSAC*, pages 21–30, 2010.

[6] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[7] George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *NDSS*, 2009.

[8] C. Doctorow. Preliminary Analysis of LinkedIn User Passwords. http://bit.ly/L5AHo3.

[9] T. Dunning. *Statistical Identification of Language*. CR Lab, New Mexico State University, 1994.

[10] J. Elder. Inside a Twitter Robot Factory. http://on.wsj.com/1bdQbEI.

[11] Lee Ellis, Kevin M Beaver, and John Wright. *Handbook of crime correlates*. Academic Press, 2009.

[12] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y Zhao. Detecting and characterizing social spam campaigns. In *IMC*, pages 35–47. ACM, 2010.

[13] Saptarshi Ghosh, Bimal Viswanath, Farshad Kooti, Naveen Kumar Sharma, Gautam Korlam, Fabricio Benevenuto, Niloy Ganguly, and Krishna Phani Gummadi. Understanding and combating link farming in the twitter social network. In *WWW*, pages 61–70. ACM, 2012.

[14] Sam Gosling. *Snoop: What your stuff says about you*. Basic Books, 2009.

[15] C Harris. Detecting deceptive opinion spam using human computation. In *Workshops at AAAI on Artificial Intelligence*, 2012.

[16] Muhammad Asim Jamshed, Wonho Kim, and KyoungSoo Park. Suppressing bot traffic with accurate human attestation. In *Proceedings of the 1st ACM asia-pacific Workshop on systems*, pages 43–48, 2010.

[17] Ioannis Kanaris, Konstantinos Kanaris, Ioannis Houvardas, and Efstathios Stamatatos. Words versus character n-grams for anti-spam filtering. *International Journal on Artificial Intelligence Tools*, 16(06):1047–1067, 2007.

[18] Ioannis Kanaris, Konstantinos Kanaris, and Efstathios Stamatatos. Spam detection using character n-grams. In *Advances in Artificial Intelligence*, pages 95–104. Springer, 2006.

[19] Eamonn Keogh, Stefano Lonardi, and Chotirat Ann Ratanamahatana. Towards parameter-free data mining. In *SIGKDD*, pages 206–215. ACM, 2004.

[20] Beate Krause, Christoph Schmitz, Andreas Hotho, and Gerd Stumme. The anti-social tagger: detecting spam in social bookmarking systems. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, pages 61–68. ACM, 2008.

[21] Ho-Yu Lam. *A learning approach to spam detection based on social networks*. PhD thesis, HKUST, 2007.

[22] Ming Li and Paul MB Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer, 2009.

[23] Vijay Mahajan and Robert A Peterson. *Models for innovation diffusion*, volume 48. Sage, 1985.

[24] Benjamin Markines, Ciro Cattuto, and Filippo Menczer. Social spam detection. In *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, pages 41–48, 2009.

[25] MediaWiki. Combating Spam - List of Proxy and Spambot IPs. bit.ly/1mwUqml.

[26] Greg Mori and Jitendra Malik. Recognizing objects in adversarial clutter: Breaking a visual captcha. In *CVPR*, volume 1, pages I–134. IEEE, 2003.

[27] Terri Oda and Tony White. Increasing the accuracy of a spam-detecting artificial immune system. In *CEC*, volume 1, pages 390–396. IEEE, 2003.

[28] Daniele Perito, Claude Castelluccia, Mohamed Ali Kaafar, and Pere Manils. How unique and traceable are usernames? In *PET*, pages 1–17. Springer, 2011.

[29] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting spammers on social networks. In *ACSAC*, pages 1–9. ACM, 2010.

[30] Dinh Nguyen Tran, Bonan Min, Jinyang Li, and Lakshminarayanan Subramanian. Sybil-resilient online content voting. In *NSDI*, volume 9, pages 15–28, 2009.

[31] Nguyen Tran, Jinyang Li, Lakshminarayanan Subramanian, and Sherman SM Chow. Optimal sybil-resilient node admission control. In *INFOCOM*, pages 3218–3226, 2011.

[32] Bimal Viswanath, Ansley Post, Krishna P Gummadi, and Alan Mislove. An analysis of social network-based sybil defenses. *SIGCOMM Computer Communication Review*, 41(4):363–374, 2011.

[33] Luis Von Ahn, Manuel Blum, Nicholas J Hopper, and John Langford. Captcha: Using hard ai problems for security. In *EUROCRYPT 2003*, pages 294–311. Springer, 2003.

[34] Alex Hai Wang. Don't follow me: Spam detection in twitter. In *SECRYPT*, pages 1–10. IEEE, 2010.

[35] T. Wasserman. 83 Million Facebook Accounts Are Fake. http://on.mash.to/1hdze2B.

[36] Wikipedia. Keyboard Layouts. http://bit.ly/kXso.

[37] Wikipedia. List of countries by population. http://bit.ly/1eTTUHe.

[38] Gregory L Wittel and Shyhtsun Felix Wu. On attacking statistical spam filters. In *CEAS*, 2004.

[39] Yinglian Xie, Fang Yu, Kannan Achan, Rina Panigrahy, Geoff Hulten, and Ivan Osipkov. Spamming botnets: signatures and characteristics. *Computer Communication Review*, 38(4):171–182, 2008.

[40] J. Yan, A. Blackwell, R. Anderson, and A. Grant. The Memorability and Security of Passwords-Some Empirical Results. *U. of Cambridge Tech. Rep.*, 2000.

[41] Jeff Yan and Ahmad Salah El Ahmad. A low-cost attack on a microsoft captcha. In *Proceedings of the 15th ACM conference on Computer and communications security*, pages 543–554. ACM, 2008.

[42] Zhi Yang, Christo Wilson, Xiao Wang, Tingting Gao, Ben Y Zhao, and Yafei Dai. Uncovering social network sybils in the wild. In *IMC*, pages 259–268. ACM, 2011.

[43] Sarita Yardi, Daniel Romero, Grant Schoenebeck, et al. Detecting spam in a twitter network. *First Monday*, 15(1), 2009.

[44] Haifeng Yu, Phillip B Gibbons, Michael Kaminsky, and Feng Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *Security and Privacy*, pages 3–17, 2008.

[45] Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. *Computer Communication Review*, 36(4):267–278, 2006.

[46] Reza Zafarani and Huan Liu. Connecting users across social media sites: a behavioral-modeling approach. In *SIGKDD*, pages 41–49. ACM, 2013.