The Doppelgänger Bot Attack: Exploring Identity Impersonation in Online Social Networks

Oana Goga
MPI-SWS

Giridhari Venkatadri
MPI-SWS

Krishna P. Gummadi
MPI-SWS

ABSTRACT

People have long been aware of malicious users that impersonate celebrities or launch identity theft attacks in social networks. However, beyond anecdotal evidence, there have been no in-depth studies of impersonation attacks in today's social networks. One reason for the lack of studies in this space is the absence of datasets about impersonation attacks. To this end, we propose a technique to build extensive datasets of impersonation attacks in current social networks and we gather 16,572 cases of impersonation attacks in the Twitter social network. Our analysis reveals that most identity impersonation attacks are not targeting celebrities or identity theft. Instead, we uncover a new class of impersonation attacks that clone the profiles of ordinary people on Twitter to create real-looking fake identities and use them in malicious activities such as follower fraud. We refer to these as the *doppelgänger bot attacks*. Our findings show (*i*) that identity impersonation attacks are much broader than believed and can impact any user, not just celebrities and (*ii*) that attackers are evolving and create real-looking accounts that are harder to detect by current systems. We also propose and evaluate methods to automatically detect impersonation attacks sooner than they are being detected in today's Twitter social network.

1. INTRODUCTION

Today, users sign on to most online social networks like Facebook and Twitter via *weak identities*, i.e., unverified identities (accounts) that do not require users to prove that their online identities match their offline, real world, personalities. Weak identities lower the sign-on barriers for users, offer users a certain level of anonymity, but they leave the sites vulnerable to a variety of fake identities or *Sybil* attacks. Malicious attackers are known to use Sybil identities to post spam content [31] and to tamper with the popularity of content on these sites [33]. Consequently, a number of prior works have focussed on understanding and detecting Sybil attacks in online social networks [20, 25, 21, 14].

IMC'15, October 28–30, 2015, Tokyo, Japan.

DOI: http://dx.doi.org/10.1145/2815675.2815699.

In this paper, we focus on a special class of Sybil (fake identity) attacks known as *identity impersonation attacks*, where the attacker *spoofs or assumes* the identity of another real-world user, *the victim*. As more and more personal data about users becomes publicly available on the Web, impersonation attacks become easier to carry out. For instance, an attacker can easily copy public profile data of a Facebook user to create an identity on Twitter or Google+.

There are many different types of impersonation attacks based on the attacker's motivation. (*i*) In a *celebrity impersonation attack*, the attacker exploits or maligns the public reputation of the victim, whose identity she impersonated. Popular and celebrity users are often targets for such impersonation attacks [19, 26]. (*ii*) In a *social engineering attack*, the attacker abuses victim's identity to trick the victim's friends into revealing sensitive information or providing services (e.g., transfer money to the attacker) [30, 22]. (*iii*) In a *doppelgänger bot attack*, the attacker is simply interested in evading detection by the Sybil or Spam defenses deployed by site operators. As Sybil detectors are often trained to detect non-real-looking identities (e.g., lacking profile photos or location information or bio), attackers could create more real-looking fake identities by copying the attributes of a real user.

Regardless of the attacker's motivation, identity impersonation attacks could seriously damage the victim's online reputation. As people's online data is increasingly aggregated by people's search engines [28] and used for a variety of purposes including evaluating their suitability for employment [27], impersonation attacks, particularly those that go undetected, can have serious adverse consequences for the victims, even in the offline world.

Despite the serious threat posed by impersonation attacks, few studies, to date, have systematically studied impersonation attacks in online social networks. Beyond a few anecdotal examples that are reported in the popular press, we lack large datasets about real-world impersonation attacks. Perhaps, not surprisingly, most social networking sites today lack frameworks to automatically detect impersonators. Instead, they rely on manual reports from victims about accounts that are impersonating them [32], which can be risky as victims usually become aware of their attackers after their online reputation has already been damaged.

Against this background, this paper presents, to the best of our knowledge, the first extensive study of real-world impersonation attacks in online social networks. Specifically, it makes the following contributions:

1. Methodology for gathering data about impersonation attacks. We identify the fundamental challenges associated with gathering data about impersonation attacks and propose practical methods to circumvent the challenges. Our method, as described in §2, consists of two steps: (1) identify doppelgänger accounts in a social network that *portray* the same person/entity; and (2) out of the doppelgänger accounts that portray the same entity, label which accounts are legitimate and which accounts are impersonators.

The second step is complicated by the fact that people can maintain multiple legitimate accounts in a social network. We applied our method to gather data on Twitter. We identified 54,304 doppelgänger pairs of Twitter identities that portray the same person. We successfully labelled 16,572 of the doppelgänger pairs as resulting from impersonation attacks and 3,639 pairs as cases when a person maintains two legitimate accounts in the Twitter network.

2. Characterizing impersonation attacks. Our analysis of the data we gathered reveals many interesting facts about identity impersonation attacks, including some that contradicted our expectations.

(1) Contrary to our expectation that most impersonation attacks would largely target celebrity / popular user identities, we discovered that many impersonation attacks target ordinary users. (2) Additionally, many impersonation attacks do not seem to attempt social engineering attacks or even try to exploit the public reputation of their victims. They appear to have been created by attackers looking to create real looking fake accounts that could pass undetected by current Sybil detection systems. We call these attacks as the doppelgänger bot attacks. Despite their motivation, doppelgänger bot attacks can still harm the online reputation of the victim identity. (3) We found that it takes Twitter on average 287 days to suspend the impersonating accounts from our dataset. The long delay in detecting the attacks call for developing methods to detect such attacks sooner.

3. Methods to automatically detect impersonation attacks. Our analysis characterizing impersonation attacks yield novel insights for detecting impersonation attacks. First, given a pair of doppelgänger identities, we could determine whether the identity-pair is managed by the same person or whether it is the result of an impersonation attack, by comparing the social network neighborhood and interests of the user. In the former scenario, the pair of doppelgänger identities share considerable overlap in network neighborhood and interests, while in the latter, they are considerably more dissimilar. Furthermore, we find that we can infer which of the pair of doppelgänger identities is legitimate (victim) and which is the impersonator (attacker), by comparing various reputation metrics, such as creation date, number of followers etc. We find that victim identities almost always have higher reputation and older creation dates than impersonating identities.

We leverage these insights to propose automatic methods (based on machine learning techniques) to detect impersonation attacks in §4. We detect 10,894 more cases of impersonation attacks and 9,354 cases of accounts managed by the same person when we test the scheme on the 34,091 unlabeled pairs of accounts that portray the same person. More than half of the impersonating accounts detected by our method were subsequently suspended by Twitter (over a 6 month period), which shows the effectiveness of our method at detecting impersonating accounts sooner than Twitter.

In summary, our paper proposes methods to gather data about a large number of real-world impersonation attacks in online social networks. Although our method does not capture all impersonation attacks, the resulting dataset allows us to do an insightful exploration of impersonation attacks, and to build an automatic detector that is not only able to detect more such attacks, but also detect them sooner. Our work represents a useful step towards enabling users to better protect their online reputation from identity impersonation attacks.

2. DATA GATHERING METHODOLOGY

In this section, we first propose a methodology for gathering data about impersonation attacks in real-world social networks. Later, we apply our methodology to gather data about impersonation attacks on Twitter.

Intuitively, an identity impersonation attack involves an attacker creating an account (identity) *pretending* to be some other real-world user (victim), i.e., the attacker's identity mimics or copies the features of the victim's identity. However, gathering data about such attacks is surprisingly difficult in practice (which might explain why few prior studies, if any, have successfully analyzed real-world impersonation attacks).

2.1 Challenges

To understand why identifying impersonation attacks is hard, consider the real-world example attack shown in Fig. 1. We discovered this attack on "Nick Feamster", a computer science researcher, during the course of our study. We alerted Nick, who confirmed the attack, and the impersonating account has since been suspended by Twitter. We realized three challenges in the process:

Figure 1: Example of impersonation attack.

1. How do we determine which identities are attempting to *portray or represent* the same user? For instance, there are many Twitter identities with the same user name "Nick Feamster", but we felt that *only* the account that shared similar bio and photo as Nick's original account, is attempting to pretend to be Nick. Our intuitive decision raises the question: how similar should the profiles of two identities be to qualify as portraying the same user?

2. After determining that a pair of identities portray the same user, how do we determine whether the identity-pair is the result of an impersonation attack or the

result of a user simply creating multiple (duplicate) accounts for herself? For instance, users on Twitter are permitted to create multiple identities, including pseudonymous identities. To determine the shared ownership of identities, we would need to contact and obtain confirmation from the identities' owners themselves (e.g., by sending messages to the identities). However, such confirmations are not only hard to obtain for a large-scale study, but also when we attempted it on Twitter, the Twitter identity we created to contact other Twitter users for the study got suspended for attempting to contact too many unrelated Twitter identities.

3. After determining that a pair of identities portraying the same user is the result of an impersonation attack (i.e., they are not duplicate accounts owned by the same user), how do we determine which identity is legitimate and which is the impersonating identity? In our attack example, we were lucky to know the person portrayed, Nick Feamster, in the offline world. This knowledge enabled us to differentiate the legitimate identity from the attacker's identity. But, in practice, the offline user portrayed by the online identities is often unknown, i.e., it is unclear how to contact the offline user. In these scenarios, it is unclear how to differentiate the legitimate identity from the impersonating identity, as both may claim to be the legitimate identity.

Below we propose a practical methodology that tackles some of these fundamental challenges and circumvents others. We applied it on Twitter to gather data about several hundreds to thousands of real-world impersonation attacks. Our methodology does not guarantee that we would discover all or even a representative sample of all impersonation attacks occurring in Twitter. Nevertheless, given the inherent difficulty in collecting any data about impersonation attacks, we feel that our work represents a first step in the direction of addressing these challenges.

2.2 Terminology

We introduce some terminology to both simplify and clarify our discussion in the rest of the paper. (i) **doppelgänger identities:** We refer to identities as doppelgänger identities when they are determined to be portraying or representing the same user. (ii) **avatar-avatar pair:** A pair of doppelgänger identities is referred to as an avatar-avatar pair when both identities are managed by the same owner. (iii) **victim-impersonator pair:** A pair of doppelgänger identities is referred to as a victim-impersonator pair when one of the identities is legitimate (victim identity) and the other is created by an attacker (impersonating identity). Using our terminology, the above challenges can be rephrased as follows:

1. How do we identify doppelgänger identities?
2. How do we determine that a pair of doppelgänger identities is an avatar-avatar pair or a victim-impersonator pair?
3. How do we determine which of the victim-impersonator identity-pair is the victim and which is the impersonator?

2.3 Data gathering strategy

In this section, we discuss our strategy to tackle the above challenges when gathering data. We defer a detailed discussion of the collected data to the next section. At a high-level, our strategy involves first automatically identifying a large number of pairs of doppelgänger identities and then differentiating them into avatar-avatar and victim-impersonator pairs.

2.3.1 Identifying doppelgänger pairs

The ideal way to determine if a pair of identities is a doppelgänger pair would be to ask human workers if both identities portray the same user. Unfortunately, such an exercise would be very expensive to scale to millions of potential doppelgänger pairs. So we built an *automated* rule-based matching scheme that is trained on human-annotated data to determine when the profile attributes of two identities match sufficiently for humans to believe that they portray the same user.

More concretely, in the Twitter social network, every identity is associated with a set of profile attributes, such as *user-name, screen-name, location, photo, and bio*. We collected pairs of identities with different levels of profile attribute matching. Specifically, we collected three-levels of matching profile pairs: (i) *Loosely matching identities:* pairs of identities that have similar *user-name or screen-name*;[1] Starting from an initial set of Twitter profiles (see §2.4), we discovered these identity pairs via the Twitter search API that allows searching by names. (ii) *Moderately matching identities:* pairs of identities that, in addition to sharing a similar user-name or screen-name, also share one additional similar profile attribute be it *location or photo or bio*.[2] In practice, we found that location information is often very coarse-grained, at the level of countries, so we defined a tighter matching scheme ignoring location information. (iii) *Tightly matching identities:* pairs of identities that in addition to sharing a similar user-name or screen-name also share similar *photo or bio*.

For each level of matching profile pairs, we estimated the fraction of profile pairs that humans would believe as portraying the same user as follows: We selected between 50 to 250 pairs of matching profiles at each level and setup an Amazon Mechanical Turk experiment, where we gave AMT workers two links corresponding to the two Twitter accounts and we asked them to choose between three options: *'the accounts portray the same person'*, *'the accounts do not portray the same person'*, or *'cannot say'*. For each assignment we asked the opinion of three different AMT workers and only consider the *majority agreement*, i.e., when at least two AMT workers choose the same answer.

We find that, by majority agreement, AMT workers believe that 4% of loosely matching, 43% of moderately matching, and 98% of tightly matching identity-pairs portray the same user. Thus, by selecting a more conservative matching scheme, we would increase precision (i.e., be more certain) of detecting doppelgänger pairs. But, in the process, we

[1]Determining attribute similarity is a challenging task in and of itself. There is a lot of prior work, including our own, on this topic [10]. We summarize the relevant details in the Appendix.

[2]Twitter accounts that do not have profile information available such as bio, locations and photos will be automatically excluded.

would be sacrificing recall – for instance, we found that the tightly matching identity scheme captures only 65% of the doppelgänger pairs caught by moderately matching identity scheme. Since our goal is to correctly identify a large set of impersonating attacks, even if it comes at the cost of not identifying all possible impersonation attacks, we propose to use the conservative tightly matching identity scheme to detect doppelgänger pairs in Twitter.

Potential limitations: While our scheme represents a first step in the direction of scalable automated detection of doppelgänger pairs in social networks like Twitter, currently, we apply it only within a single social network. So we miss opportunities to detect doppelgänger pairs across multiple social networking sites, e.g., when an attacker copies a Facebook user's identity to created a doppelgänger Twitter identity. While our basic scheme could be extended to match identities across sites, it is beyond the scope of this work.

2.3.2 Identifying victim-impersonator pairs

As discussed earlier, the ideal way to determine whether a doppelgänger pair is a victim-impersonator pair requires contacting the real offline user represented by the identity to inquire about the ownership of the identities. However, this approach is infeasible in practice. When we attempted this approach, our Twitter identity got quickly suspended as indulging in potentially spam activity. So we instead rely on a signal from Twitter, when it officially suspends *one, but not both*, of the doppelgänger identities. We crawled the doppelgänger identities periodically (once a week) over an extended period of time (a three month period) to look for identity suspensions. We treat the suspended identity of the doppelgänger pair as the impersonating identity and the other identity as the victim.

Our hypothesis is that at least some fraction of the impersonating identities would be eventually detected and reported to Twitter (either by the victim herself or some other users that know the victim), which would result in Twitter suspending the identity. One concern with our approach is that we may be detecting impersonating attacks that are being caught by some automated Twitter spam defense system (as opposed to reports filed by victims or other Twitter users). In this case, we would essentially be reverse-engineering Twitter's impersonation detection system rather than study impersonation attacks in the wild. We address this concern in §4.2, where we show that the analysis of the impersonation attacks gathered using our methodology can help us design new automated detectors that in turn could be used to detect a large number of yet undiscovered impersonation attacks in Twitter. Had the identities been suspended by an automated impersonation detection framework in Twitter, it is unlikely that we would have succeeded in designing significantly better performing detection systems.

Potential Limitations: Our victim-impersonator pair detection strategy allows us to detect large numbers of such pairs, but it likely captures only those impersonation attacks that have been detected by Twitter's reporting system. We would be under-sampling clever attacks that have not yet been detected.

2.3.3 Identifying avatar-avatar pairs

As discussed earlier, the ideal way to determine whether a doppelgänger pair is avatar-avatar by contacting the owners of the identities is unfeasible in practice. So we instead rely on observing interactions between the doppelgänger identities that clearly indicate that each identity is aware of the presence of the other identity. Specifically, we check whether one of the doppelgänger identity follows or mentions or retweets the other doppelgänger identity. If it is the case, it is very likely that the identities are managed by the same user. Otherwise, the legitimate identity would have reported the impersonating identity and have it suspended by Twitter.

Potential Limitations: Our avatar-avatar pair detection strategy under-samples scenarios where a user maintains multiple identities but keeps them distinct, i.e., does not link the identities and use them for very different purposes. However, we suspect that in such scenarios, users would likely assume different pseudonymous identities and would avoid providing the same profile information. Such identities would not be matched as doppelgänger identities in the first place.

2.4 Data gathered

In this section, we describe how we applied our above data gathering strategy to collect information about real-world impersonation attacks in the Twitter social network at scale.

We begin by selecting 1.4 million *random* Twitter accounts – the initial accounts.[3] We call the dataset we generate starting with these random Twitter accounts the RANDOM DATASET. For each account a in the RANDOM DATASET, we gather a set of up to 40 accounts in Twitter that have the most similar names as the account (using the Twitter search API). We call the resulting 27 million name-matching identity-pairs initial accounts. From these pairs, we identify doppelgänger pairs, victim-impersonator pairs, and avatar-avatar pairs as described in §2.3. Table 1 summarizes the dataset. Note that a significant fraction of doppelgänger identities are not labeled as either avatar-avatar or victim-impersonator pairs.

While our strategy for detecting doppelgänger and avatar-avatar pairs yielded sizable numbers, our strategy for detecting victim-impersonator pairs proved quite time consuming. It took a 3 months waiting time to discover 166 victim-impersonator pairs amongst the 18,662 doppelgänger pairs and few tens of identities keep getting suspended every passing week. To quickly identify more victim-impersonator pairs, we resorted to a focussed crawl in the neighborhood of the detected impersonating identities. Specifically, we conducted a breadth first search crawl on the followers of four seed impersonating identities that we detected. Our intuition is that we might find other impersonating accounts in the close network of an impersonating account.

We collected 142,000 accounts with the breadth first search crawl and we call the dataset generated from this biased set of initial accounts the BFS DATASET. We repeated the same analysis on the BFS DATASET that we conducted on RANDOM DATASET. We report the results in Table 1. In the same amount of time, we discovered 16,408 victim-impersonator pairs out of the 35,642 doppelgänger pairs, suggesting that our focussed crawl succeeded in identifying a large number of real-world impersonation attacks.

[3]Twitter assigns to every new account a numeric identity that allows random sampling.

Table 1: Datasets for studying impersonation attacks.

	Random Dataset	BFS Dataset
initial accounts	1.4 millions	142,000
initial accounts	27 millions	2.9 millions
doppelgänger pairs	18,662	35,642
avatar-avatar pairs	2,010	1,629
victim-impersonator pairs	166	16,408
unlabeled pairs	16,486	17,605

For each Twitter identity in doppelgänger pairs, we use the Twitter API to collect detailed information about a variety of their features. They include features related to:

1. Profile of the identity: We gather the data about the identity's *user-name, screen-name, location, photo, and bio.*

2. Activity of the identity: We gather data about the *creation date of the account, timestamp of the first tweet, timestamp of the last tweet, number of followings (the number of accounts a user follows), number of tweets posted, number of retweets posted, number of tweets favorited* and *number of mentions.*

3. Reputation of the identity: We collect data about the *number of followers* of an account, the *number of expert lists where the user appears* and the *klout score* [16] as metrics to measure the influence of an account. The *klout score* is a widely used metric to measure the social influence of an account.

3. CHARACTERIZING IMPERSONATION ATTACKS

In this section, we analyze the datasets we collected to characterize identity impersonation attacks in Twitter. We begin by investigating the different types of impersonation attacks found in our Random Dataset. Our analysis reveals the prevalence of a new type of impersonation attack that we call **doppelgänger bot** attack. We then explore the features of doppelgänger bot attacks and the potential for detecting such attacks.

3.1 Classifying impersonation attacks

Based on conventional wisdom and anecdotal evidence, we were expecting to discover two types of impersonation attacks in our Random Dataset: (i) *Celebrity impersonation attacks*, where attackers impersonate celebrities and popular Twitter users to either post untrustworthy information maligning the celebrity's reputation or take advantage of the celebrities' offline popularity to increase the visibility of their own posts (e.g., product promotions) or (ii) *Social engineering attacks* also known as identity theft attacks, where the attacker creates a fake account that clones the information of a victim account and then uses the fake account to connect and communicate with the victim's friends [5]. The ultimate goal here is to launch phishing attacks to harvest sensitive information about the victim or to trick the victim's friends into sending money to the attacker (that claims to be the victim).

We attempted to map the 166 victim-impersonator pairs in our dataset to these two types of attacks. In the process, we discovered that many of the victim-impersonators pairs corresponded to a small number of victims. Specifically, there were 6 different Twitter victims that in total accounted for half (83) of the victim-impersonator pairs. One hypothesis is that these six victims discovered multiple fake identities that were impersonating them and reported all such identities, leading them to be detected in our methodology. To avoid over-sampling these identities in our dataset, we only consider one pair of victim-impersonating identities for each of the 6 victims, which reduces our dataset to 89 victim-impersonator pairs.

3.1.1 Celebrity impersonation attacks

Twitter allows users to create accounts that are fan pages of celebrities, however, users have to specifically declare this in their bios. Our method to collect data about impersonation attacks in the previous section is consistent with Twitter's terms of service. If the fan account mentions or interacts in any way with the celebrity, it will be identified as an avatar, and if not, it will be identified as impersonator.

To identify celebrity impersonation attacks we simply check for victim-impersonator pairs where the victim is either a verified Twitter account [4] or has a popular following amongst Twitter users, i.e., it has more than 1000 or 10,000 followers.[5] Out of the 89 victim-impersonator pairs, we found *only three* are celebrity impersonation attacks out of which one is a impersonator of a football player and one of a journalist. In fact, 70 of the 89 victims have less than 300 followers, suggesting that most victims of impersonation in our dataset are not highly popular celebrities.

3.1.2 Social engineering attacks

While it is impossible to exactly know the intentions of the attacker, we could attempt to infer if an impersonating identity is attempting to launch a social engineering attack, by exploiting the observation that attackers try to contact the friends of the victims. So we select all victim-impersonator pairs where the impersonating account had *any interaction* with users that know the victim account, i.e., the impersonating account is friend of, follows, mentions or retweets people that are friends of or follow the victim account.

Our hypothesis is that it is unlikely that accounts that do not fall in the candidate set would try to mount social engineering attacks since they do not show any intent of contacting the people that know the victim. Out of the 89 victim-impersonator pairs in our dataset, *only two* accounts met our test for potentially being social engineering attacks.

3.1.3 Doppelgänger bot attacks

Our analysis of impersonation attacks above suggests that most attacks are not directed towards popular celebrities or attempting to trick a victim's friends. This observation raises the question: *What then might be motivating an attacker to launch an impersonation attack targeting non-celebrity users?*

One hypothesis is that these impersonation attacks are merely an attempt by attackers creating fake identities to evade detection by Twitter Sybil and spam defense systems. By simply copying the profile attributes of existing Twitter users, attackers could create new real-looking fake Twitter identities that might escape traditional spam defenses

[4]Twitter has an account verification program for highly popular users of its service.

[5]Less than 0.01% (0.007%) of Twitter users have more than 1000 (10,000) followers.

that analyze features of individual identities. Such identities could be used profitably to promote other Twitter users and content in the system (selling fake followers and fake content promotions is a growing business in Twitter). We refer to such attacks as **doppelgänger bot** attacks.

To check our hypothesis that many impersonating accounts are involved in follower fraud we analyzed whom they are following. Since the number of doppelgänger bots in our RANDOM DATASET is limited to less than a hundred, we investigated the doppelgänger bots in the BFS DATASET, where we have identified tens of thousands of victim-impersonator pairs using a BFS crawl of the Twitter network starting with four seed doppelgänger bot accounts (see Table 1). We found that the impersonating accounts in the BFS DATASET follow a set of 3,030,748 distinct users. Out of the users followed, 473 are followed by more than 10% of all the impersonating accounts. We checked if the 473 most followed accounts are suspected of having bought fake followers in the past using a publicly deployed follower fraud detection service [34]. Among those users for which the service could do a check, 40% were reported to have at least 10% fake followers.[6] The fact that a large fraction of impersonating accounts follow the same small set of users and that the users they follow are suspected of having bought fake followers strongly points to the possibility of impersonating accounts being involved in follower fraud.

3.2 Analyzing doppelgänger bot attacks

In this section, our goal is to better understand doppelgänger bot attacks with the ultimate goal of detecting doppelgänger bots. To this end, we focus on understanding the characteristics (i.e., various reputation metrics and activities) of doppelgänger bots and their victims. Our investigation reveals important differences between the characteristics of victim accounts and doppelgänger bots, which we leverage in a later section to detect doppelgänger bots. The doppelgänger bot attacks analyzed in this section are from the BFS DATASET.

3.2.1 *Characterizing victim accounts*

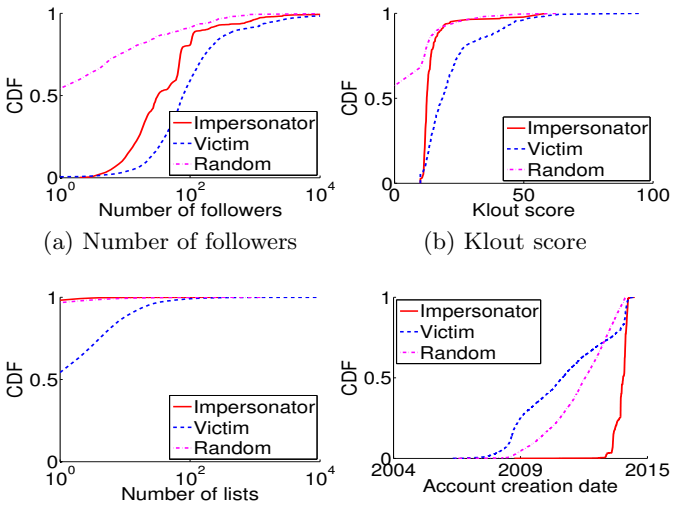
To understand who the doppelgänger bot attacks are targeting we proceed by asking several questions related to the reputation and activity of victim accounts. We focus our analysis on the differences in the statistical distributions of properties of victim accounts and randomly chosen Twitter users.

Reputation of victim accounts.

How popular are the victim accounts? Figure 2a shows the CDF of the number of followers of victim accounts. The median number of followers is only 73, which shows that most doppelgänger bots do not target famous people but ordinary users.

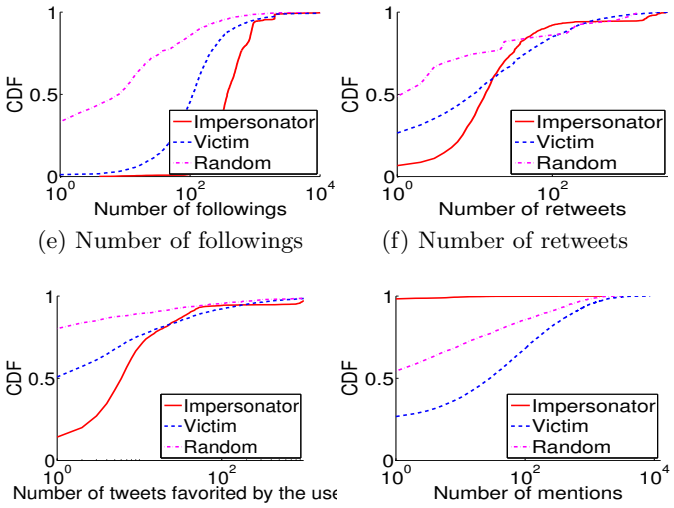
How influential are the victim accounts? Figures 2b and 2c show the CDF of klout scores and the number of expert lists where a victim account appears in, respec-

[6]For a point of comparison, we also checked whom the avatar accounts from avatar-avatar pairs in the BFS DATASET follow. There are only four accounts followed by 10% of the avatar accounts and they correspond to Justin Bieber, Taylor Swift, Katy Perry and Youtube, all of which are well-known celebrity / corporate accounts followed widely on Twitter.



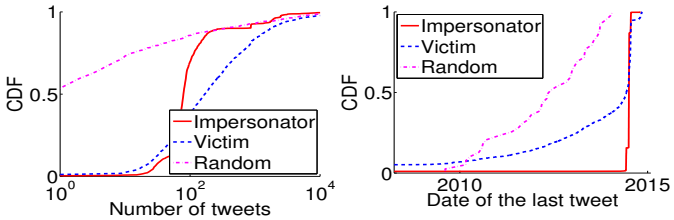
(a) Number of followers (b) Klout score

(c) Number of lists the account appears in (d) Account creation date



(e) Number of followings (f) Number of retweets

(g) Number of tweets favorited by the user (h) Number of mentions



(i) Number of tweets (j) Date of the last tweet

Figure 2: CDFs of different features for characterizing the reputation and activity of impersonating and victim accounts as well as Twitter accounts picked at random.

tively. 40% of victim accounts appear in at least one list and 30% of victim accounts have klout scores higher than 25 (For comparison purposes, researchers like Dina Papagiannaki – @dpapagia – and Jon Crowcroft – @tforcworc – have klout scores of 26 and 45 respectively, while Barack Obama –@barackobama–has a klout score of 99). The figures also show that the influence scores of victims are noticeably higher than those of random Twitter users, which indicates that many victim accounts, while not exactly celebrities, correspond to professional users with good reputation in Twitter.

How old are the victim accounts? Figure 2d shows that victims have generally older accounts than random Twitter users. The median creation date for victim accounts is October 2010 while the median creation date for random Twitter users is May 2012. Attackers target users that have been for a long time in the system.

Activity of victim accounts.

How active are the victim accounts? Figures 2e through 2j show the CDFs of various activity metrics for victims. They show that victims are considerably more active than random Twitter users. For instance, Figure 2i shows the CDF of the number of tweets per victim account. The median number of tweets is 181. In contrast, the median number of tweets for random users is 0, while the median number of tweets for random users that have at least one post is 20. Similarly, Figure 2j shows that 75% of victim accounts posted at least one tweet in 2013, while only 20% of random Twitter users posted at least one tweet in 2013. So the victims tend to be fairly active Twitter users that have been active recently too.

In summary, the victims of doppelgänger bot attacks are active users with a level of reputation and influence that is significantly higher than a random Twitter user (though they are not necessarily celebrities). Our finding shows that attackers target, inadvertently or not, users who put a significant amount of effort into building an online image with a good reputation.

3.2.2 Characterizing doppelgänger bot accounts

To understand how doppelgänger bot impersonating accounts behave we analyze their reputation and activity.

Reputation of doppelgänger bot accounts.

Figures 2a through 2d compares the CDFs of different reputation metrics for doppelgänger bots, their victims, and random Twitter accounts. The plots show that: (1) the *number of followers* and *klout score* of impersonating accounts is lower than the number of followers and klout score of victim accounts but higher than the ones of random accounts (Figures 2a and 2b); (2) impersonating accounts do not appear in any *lists* of other accounts (Figure 2c); and (3) most impersonating accounts were *created* recently, during 2013 (Figure 2d). At a high level, the reputation of doppelgänger bots is clearly lower than the reputation of victim accounts, however, it is higher than the reputation of random Twitter users. In the next section, we will show that these differences have important implications for the detectability of doppelgänger bots. Thus, impersonating accounts do not have suspicious markers of reputation.

Activity of doppelgänger bot accounts.

We refer again to Figures 2e through 2j. The plots show that: (1) the *number of followings, retweets* and *favorites* of impersonating accounts is higher than victim and random accounts (Figures 2e, 2f, and 2g). This finding is consistent with our hypothesis that these accounts are involved in illegal promotion of content and users; (2) However, the number of times doppelgänger bots *mention* other users is unusually low, which is consistent with the intuition that doppelgänger bots would not not wish to draw attention to themselves and their activities (Figure 2h); and (3) impersonating accounts do not show excessive markers of activity: the median number of users an impersonating account follows is 372, which is not very high compared with the median number a victim account follows which is 111 and impersonating accounts do not tweet excessively (Figure 2i), but are very active, i.e. their *last tweet* is in the month we crawled them (Figure 2j). These markers of activity not only support our hypothesis that doppelgänger bots might be involved in illegal promotion of content, but they also suggest that attackers may be trying to emulate normal user behavior and avoid being detected with abnormal (excessive) tweeting or following behavior.

3.3 Detecting doppelgänger bot attacks

Even if doppelgänger bots do not seem to intentionally want to harm their victims, they can still harm unintentionally the online image of their victims. For example, in our dataset a doppelgänger bot of a tech company tweeted "I think I was a stripper in a past life", which is clearly not the image the tech company wants to promote. Even worse, Twitter took in average 287 days to suspend these accounts.[7] Thus, the online image of victims was potentially harmed for several months.

The natural question that follows is *why do these accounts go undetected for such long periods of time?* In this section, we analyze the challenges in detecting doppelgänger bots; in particular our analysis shows that traditional sybil detection schemes are not able to accurately detect them. The low accuracy can be potentially explained by the fact that doppelgänger bots operate under the radar and the accounts do not have any characteristics that makes them look suspicious in *absolute*. We show that the key to detect doppelgänger bots is to study their characteristics *relative* to their corresponding victim accounts.

Using traditional sybil-detection schemes: reasoning about the absolute trustworthiness of accounts.

Doppelgänger bots are a special type of sybil accounts, thus we investigate whether we can detect them using existing methods to detect sybil accounts. At a high level, many traditional sybil-detection schemes exploit ground truth about good and bad users to create patterns of good and bad behavior. The schemes then decide whether an account is sybil or not by comparing its behavior to these patterns [3, 40, 29]. We emulate such behavioral methods by training a SVM classifier with examples of doppelgänger bots (bad behavior) and random Twitter accounts (good behavior) using the methodology in [3].

[7]We know with an approximation of one week when Twitter suspended the impersonating accounts and we know from the Twitter API when the account was created.

We consider all doppelgänger bots from the BFS DATASET as positive examples (16,408 accounts), and we pick 16,000 random Twitter accounts as negative examples. We use 70% of examples for training and 30% for testing and we train the classifier with all features that characterize the reputation and the activity of a single account we presented in §2.4.

Because doppelgänger bots are only a small fraction of all accounts in Twitter, our classification problem has a high class imbalance (i.e., there are many more negative examples than positive examples). Such scenarios require classifiers that can operate at a very low false positive rate. The smallest false positive rate our SVM classifier achieves is 0.1% for a 34% true positive rate (the true positive rate drops to zero for lower false positive rates). A 0.1% false positive rate is, however, very high in our scenario. For example, if we take the 1.4 million random accounts (in the RANDOM DATASET), a classifier with 34% true positive rate for a 0.1% false positive rate will detect 40 actual doppelgänger bots (34%×122) while mislabeling 1,400 legitimate accounts as doppelgänger bots. This accuracy is clearly unsatisfying.

A plausible reason why these schemes are not optimal is precisely because attackers intentionally create real-looking accounts and emulate the behavior of legitimate users so that they are harder to detect by current sybil account detection systems.

Distinguish doppelgänger bots from victim accounts: reasoning about the relative trustworthiness of accounts.

The previous section showed that, given a set of accounts, it is hard to detect which ones are doppelgänger bots. Here we try to approach the problem from a different perspective and we ask a different question: *given a victim-impersonator pair can we pinpoint the impersonating account?* Our intuition is that, if it is too hard to reason about the trustworthiness of an account in absolute, it might be easier to reason about its trustworthiness relative to another account.

To answer the question we refer back to Figure 2 that presents the CDFs for different features of impersonating, victim and random accounts. We can see that the characteristics of victim accounts in aggregate are very different from the characteristics of impersonating accounts. More precisely, victim accounts have a much higher reputation (number of followers, number of lists and klout scores) than impersonating accounts and they have a much older account creation date. In fact, in all the victim-impersonator pairs in the BFS DATASET and RANDOM DATASET, none of the impersonating accounts have the creation date after the creation date of their victim accounts and 85% of the victim accounts have a klout score higher than the one of the impersonating accounts. Thus, to detect the impersonating account in a victim-impersonator pair with no miss-detections, we can simply take the account that has the more recent creation date. This reasoning opens up solutions to detect doppelgänger bots, however, it does not solve the whole problem because we still have to detect whether a pair of accounts is a avatar-avatar pair or victim-impersonator pair. This is the focus of section §4.

How well humans can detect doppelgänger bots.

In this section, we investigate how well humans are able to detect doppelgänger bots. We focus on two questions: (1) *If humans stumble upon a doppelgänger bot, are they able to detect that the account is fake?* (i.e., the question of assessing the absolute trustworthiness of accounts) – this scenario is specific to a recruiter that knows the name of the person and searches for his accounts in different social networks to learn more about him and stumbles upon the doppelgänger bot; and (2) *If humans have access to both the impersonating and the victim account, are they able to detect the impersonating account better?* (i.e., the question of assessing the relative trustworthiness of accounts). The first question will show the severity of the doppelgänger bot attacks problem by analyzing whether humans are tricked into believing the doppelgänger bots represent the real person. The second question will show whether humans are also better at detecting impersonating accounts when they have a point of reference.

We build two AMT experiments to answer these questions. For the first AMT experiment, we select 50 doppelgänger bot accounts and 50 avatar accounts from the victim-impersonator pairs and avatar-avatar pairs. In each assignment, we give AMT workers a link to a Twitter account and we ask them to choose between three options: *'the account looks legitimate'*, *'the account looks fake'* and *'cannot say'*. We mix doppelgänger bot accounts and avatar accounts in the experiments to force users to examine each case afresh. In all experiments we ask the opinion of three AMT workers and we report the results for majority agreement. In this experiment, AMT workers are able to only detect 18% of the doppelgänger bots as being fake (9 out of 50). Thus, most AMT workers get tricked by doppelgänger bots.

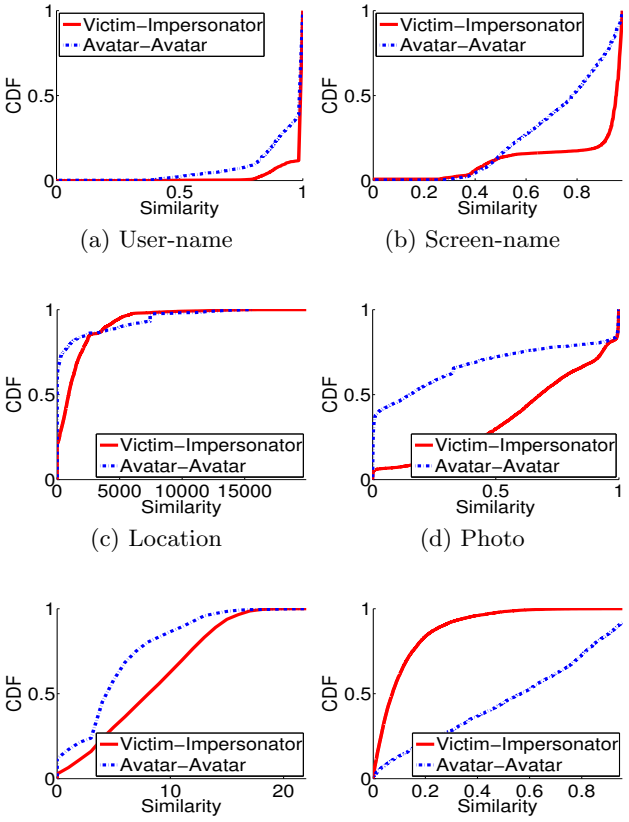
In the second experiment we show AMT workers two accounts that portray the same person. We picked the same 50 impersonating accounts (and their corresponding victims) and the same 50 avatar accounts (and their corresponding avatar doppelgänger) as in the previous experiment.[8] In each assignment, we give AMT workers two links corresponding to the two Twitter accounts and we ask them to choose between five options: *'both accounts are legitimate'*, *'both accounts are fake'*, *'account 1 impersonates account 2'*, *'account 2 impersonates account 1'*, and *'cannot say'*. In the second experiment, AMT workers were able to correctly detect 36% doppelgänger bots as fake. The experiment shows that there is a 100% improvement in their detection rate when they have a point of reference.

The results in this section have implications on both how to design automatic techniques to detect doppelgänger bots as well as how to design systems that better protect users from being deceived online by impersonation attacks.

4. DETECTING IMPERSONATION ATTACKS

The previous section showed that given a victim-impersonator pair of accounts, we can fairly accurately detect the impersonating account by comparing their account creation times and reputations. In this section, we investigate the extent to which we can detect whether a pair of accounts that portrays the same person (a doppelgänger pair) is a victim-impersonator pair or an avatar-avatar pair. We start by analyzing features that can signal the existence of an impersonation attack and we then propose a method to automatically detect such attacks.

[8]The AMT workers were different in the two experiments.

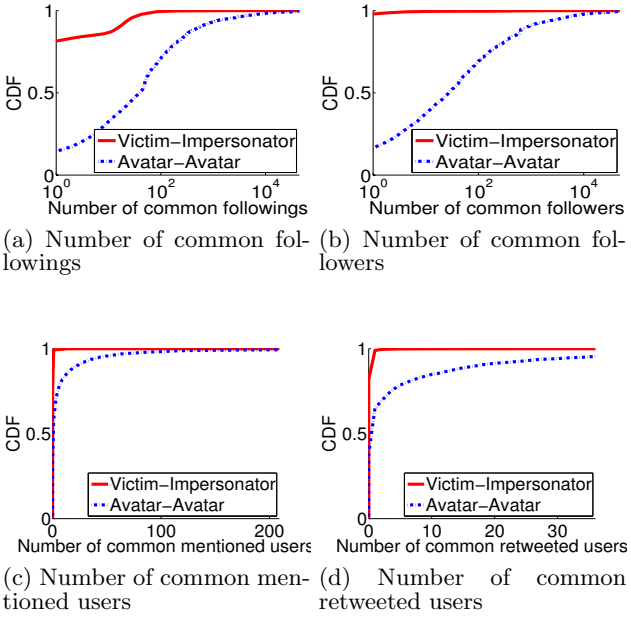


(a) User-name (b) Screen-name

(c) Location (d) Photo

(e) Bio (f) Interests similarity

Figure 3: CDFs of the similarity between accounts in victim-impersonator pairs and avatar-avatar pairs.



(a) Number of common followings (b) Number of common followers

(c) Number of common mentioned users (d) Number of common retweeted users

Figure 4: CDFs of the social neighborhood overlap between accounts in victim-impersonator pairs and avatar-avatar pairs.

Twitter suspension signals and direct account interactions are very good signals to create a dataset to study impersonation attacks, however, there are still many doppelgänger pairs that are not yet labeled (e.g., there are 16,486 unlabeled pairs in the RANDOM DATASET). Thus, a secondary goal of this section is to investigate whether we can detect additional victim-impersonator pairs and avatar-avatar pairs in the doppelgänger pairs in our dataset.

4.1 Features to detect impersonation attacks

To detect impersonation attacks we consider features that characterize pairs of accounts and that can potentially differentiate victim-impersonator pairs from avatar-avatar pairs. We consider all victim-impersonator pairs and avatar-avatar pairs from the RANDOM DATASET and BFS DATASET combined (we call the combined dataset the COMBINED DATASET) to analyze how well the features distinguish between the two kinds of pairs of accounts.

Profile similarity between accounts.

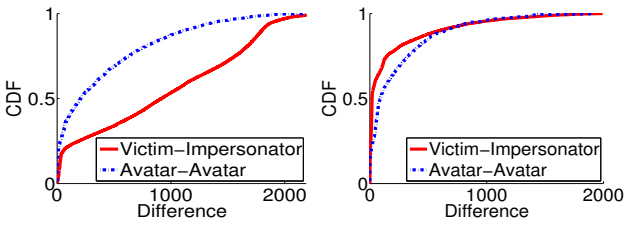
We first analyze the similarity between profile attributes such as *user-names, screen-names, locations, profile photos* and *bios* (refer to the Appendix for details on how we compute the similarity scores for different attributes). Even if these features were used to collect the dataset of doppelgänger pairs we can still use them to distinguish between avatar-avatar pairs and victim-impersonator pairs. In addition, we measure the similarity between the *interests* of two accounts. We use the algorithm proposed by Bhattacharya et al. [4] to infer the interests of a user.

Figure 3 compares the CDFs of the pairwise profile similarity between accounts in avatar-avatar pairs and victim-impersonator pairs. For user-names, screen-names, photo and interests similarity, a value of zero means no similarity while one means perfect similarity. For the location, the similarity is the distance in kilometers between the two locations, thus a value of zero means the locations are the same. For bio, the similarity is the number of common words between two profiles, the higher the similarity the more consistent the bios are.

We observe that the similarity between user-names, screen-names, profile photos and bios is higher for victim-impersonator pairs than avatar-avatar pairs. Thus, users that maintain multiple avatar accounts do not spend the effort to make their accounts look similar, while impersonators do. On the other hand, the similarity between the *interests* of avatar-avatar pairs is higher than the victim-impersonator pairs. We did not expect such high similarity between avatar-avatar pairs because we believed that people maintain distinct accounts to promote different sides of their persona.

Social neighborhood overlap.

We call the set of users an account interacts with in a social network the *social neighborhood* of the account. On Twitter, the social neighborhood of an account a consists of the followings and followers of a as well as the users mentioned by a and the users retweeted by a. An overlap in the social neighborhood suggests that two accounts are positioned in the same part of the social network graph. This can be in-



(a) Time difference between creation dates

(b) Time difference between the last tweet

Figure 5: CDFs of the time difference in days between accounts in victim-impersonator pairs and avatar-avatar pairs.

dicative of two things: (1) the two accounts correspond to avatars managed by the same user; or (2) potential evidence of social engineering attacks. We use four features to measure the social neighborhood overlap: the *number of common followings*, the *number of common followers*, the *number of overlapping users mentioned* and the *number of overlapping users retweeted* by both accounts, which we present in Figure 4.

There is a striking difference between avatar-avatar pairs and victim-impersonator pairs: while victim-impersonator pairs almost never have a social neighborhood overlap, avatar accounts are very likely to have an overlap. Social neighborhood overlap is also indicative of social engineering attacks but there are not many such attacks in our dataset to be visible in the plots.

Time overlap between accounts.

We add features related to the time overlap between two accounts: *time difference between the creation dates*, *time difference between the last tweets*, *time difference between the first tweets* and whether one account stopped being active after the creation of the second account (we call this feature *outdated account*). Figure 5 compares the difference between creation dates and the date of the last tweet. Figure 5a shows that there is a big difference between account creation times for victim-impersonator pairs while for avatar-avatar pairs the difference is smaller.

Differences between accounts.

Finally, we consider a set of features that represent the *difference* between different numeric features that characterize individual accounts: *klout score difference*, *number of followers difference*, *number of friends difference*, *number of tweets difference*, *number of retweets difference*, *number of favorited tweets difference*, *number of public list difference*. Our intuition was that the difference between numeric features of accounts in avatar-avatar pairs will be smaller than for victim-impersonator pairs, e.g., a small klout score difference between two accounts could be indicative of avatars of the same person while a large klout score difference could be indicative of an impersonation attack. To our surprise, the difference is generally slightly smaller for victim-impersonator pairs.

Overall, the best features to distinguish between victim-impersonator pairs and avatar-avatar pairs are the interest similarity, the social neighborhood overlap as well as the difference between the creation dates of the two accounts.

4.2 Automated detection method

To build an automated method to detect impersonation attacks we build a SVM classifier, with linear kernel, that distinguishes victim-impersonator pairs from avatar-avatar pairs. We use, from the COMBINED DATASET, the victim-impersonator pairs as positive examples, and avatar-avatar pairs as negative examples to train the classifier. We use *all* the features presented in §4.1 as well as *all* the features that characterize individual accounts presented in §2.4 for the training. Since the features are from different categories and scales (e.g., time in days and distances in kilometers), we normalize all features values to the interval [-1,1].

We use 10-fold cross validation over the COMBINED DATASET to train and test the classifier. The SVM classifier, for each pair of accounts, outputs a probability of the pair to be a victim-impersonator pair. To perform the detection of victim-impersonator pairs and avatar-avatar pairs, we then proceed as follows. If the probability is higher than a certain threshold $th1$ we conclude that the pair is a victim-impersonator pair and if the probability is lower than a certain threshold $th2$ (different than $th1$) the pair is a avatar-avatar pair. Note that if $th1 > th2$, some pairs may remain unlabeled. This is done on purpose here because it is preferable in our problem to leave a pair unlabeled rather than wrongly label it (i.e., label avatar-avatar pairs as victim-impersonator pairs or vice versa). We therefore select thresholds $th1$ and $th2$ such that there are very few false positives (i.e., few victim-impersonator pairs mislabeled as avatar-avatar pairs or vice versa). The resulting classifier is able to achieve a 90% true positive rate for a 1% false positive rate to detect victim-impersonator pairs and a 81% true positive rate for a 1% false positive rate to detect avatar-avatar pairs.[9] Thus, we can detect a significant fraction of victim-impersonator pairs using only features that compare the reputation and activity of accounts in a doppelgänger pair. Therefore, it is possible to detect impersonation attacks automatically rather than wait for victims to report them or wait for the accounts to do something clearly malicious in order to be suspended by Twitter.

Potential limitations: Our detection method above, while quite effective today at detecting whether a doppelgänger pair is the result of an impersonation attack, is not necessarily robust against adaptive attackers that might change their strategy to avoid detection in the future. Similar to spam detection, system operators to constantly retrain the detectors (classifiers) to account for new attacker strategies. Also note that the accuracy percentages above only refer to the accuracy of detecting whether a doppelgänger pair is a victim-impersonator pair and does not include the accuracy of detecting a doppelgänger pair or the accuracy of detecting the doppelgänger bot account within a victim-impersonator pair of accounts.

4.3 Classifying unlabeled doppelgänger pairs

We apply the classifier over the 17,605 unlabeled pairs from the BFS DATASET and the 16,486 unlabeled pairs from the RANDOM DATASET. With a threshold corresponding to 1% false positive rate (for both detecting victim-impersonator pairs and avatar-avatar pairs) the classifier is able to identify 4,390 avatar-avatar pairs and 1,863 victim-impersonator

[9]Since there is little to no class imbalance in this classification problem, contrary to §3.3, a 1% false positive rate is low enough.

Table 2: Unlabeled doppelgänger pairs in our dataset that we can labeled using the classifier.

	BFS Dataset (17,605 unlabeled)	Random Dataset (16,486 unlabeled)
victim-impersonator pairs	9,031	1,863
avatar-avatar pairs	4,964	4,390

pairs in the Random Dataset (see Table 2). Thus, the classifier can identify a large additional number of avatar-avatar pairs and victim-impersonator pairs that were not caught in the initial dataset. For example, on top of the 166 examples of victim-impersonator pairs we initially labeled, the classifier labels 1,863 additional victim-impersonator pairs.

We re-crawled all doppelgänger pairs (from both datasets) in May 2015 (the initial crawl ended in Dec 2014), and 5,857 out of the 10,894 victim-impersonator pairs detected by our classifier were suspended by Twitter. This result shows the effectiveness of our method at detecting victim-impersonator pairs sooner than Twitter.

5. RELATED WORK

The closest to our work are a few studies on social engineering attacks which we will review in more detail. Also related are studies of matching accounts across social networks and sybil account detection techniques that we review at a more higher level.

Social engineering attacks.

We focus in this paper on a broader set of attacks that impersonate people, of which, social engineering attack are a subclass. Bilge et al. [5] demonstrated the feasibility of automatically creating cloned profiles in social networks, however, they did not propose techniques to detect the attacks. The closest to our work are three studies [17, 13, 15] that made some initial investigations toward detecting profile cloning. The studies hinted at the fact that cloned profiles can be identified by searching for profiles with similar visual features, but they either stopped at returning a ranked list of accounts that are similar with the victim account [17, 13], or to just test their technique on simulated datasets [15]. In contrast, we actually detect accounts that portray the same person in real-world social networks with high accuracy and we also detect whether they are involved in an impersonation attack or they are legitimate. Furthermore, we propose a technique to gather data about impersonation attacks in real-world social networks and we do the first, to our knowledge, characterization of impersonation attacks in Twitter. On the protection part, He et al. [11] proposed ways to protect against friend requests coming from cloned profiles by using adjacent mediums such as instant chats to verify if the request comes from the real person. To protect users we showed that humans are much better at detecting impersonating accounts when they can also see the victim account. Thus a system that protect users from friend requests coming from cloned profiles could simply just show the user all the accounts that portray the same person with the account that is requesting the friendship.

Account matching.

There are a number of works that propose methods to match the accounts a user has on multiple social networks that are related to our techniques to detect doppelgänger pairs. Note, however, the subtle difference, our goal is to find accounts that people think they *portray* the same person which is slightly different than detecting accounts that *belong* to the same user. To detect doppelgänger pairs we firstly have to only rely on visual features of accounts and then understand when humans get confused.

While there are several studies that exploited visual features similar to the features we use in this paper to match accounts (refer to [9] for an overview), none of these studies applied their methods to detect impersonation attacks. Most of the studies build classifiers that are able to detect whether two accounts belong or not to the same person. We drew inspiration from these works, however, we could not directly apply these techniques to gather doppelgänger pairs because we could not estimate their accuracy as there is no existing ground truth of accounts that portray the same person in the same social network.

Sybil account detection.

One of the most widely used approach today to detect fake accounts is to build behavioral profiles for trusted and untrusted users [3, 40, 29]. The behavioral profile can include, for example, the act of sending messages to other identities, following identities, or rating a particular piece of content. We showed that behavioral profiles are not optimal for detecting impersonating accounts and that we have to exploit features that characterize pairs of identities to identify impersonating accounts.

To assess the trustworthiness of identities, another type of information typically available on social networks is trust relationship *between* identities (e.g., friendship relationship between identities). Researchers have proposed a variety of schemes such as SybilGuard [39] and SybilRank [6] that analyze trust networks between identities to assess the trustworthiness of identities and thus identify Sybil attackers [39, 36, 35]. The key assumption is that an attacker cannot establish an arbitrary number of trust edges with honest or good users in the network. This assumption might break when we have to deal with impersonating accounts as for them it is much easier to link to good users, but it would be interesting to see whether these techniques are able to detect doppelgänger bots.

A third approach to identify suspicious identities is to crowdsource this task to experts who are familiar with identifying suspicious profiles or actions. Social networking services typically have a tiered approach where suspicious profiles reported by end users are further verified by a group of experts before taking a decision to suspend the account or show Captchas to those suspicious users [6]. In fact, researchers recently explored the possibility of using online crowdsourcing services such as Amazon Mechanical Turk (AMT) to crowdsource the task of detecting sybil identities in a social network [37]. Our AMT experiments showed, however, that such techniques are not optimal for detecting impersonating accounts because AMT workers get tricked easily to believe that impersonating accounts are legitimate.

6. CONCLUSION

We conducted the first study to characterize and detect identity impersonation attacks in Twitter. The key enabler of this study is our method to gather data of impersonation attacks. Our method is general and can be used to gather data in other social networks such as Facebook and LinkedIn.

Besides celebrity impersonators and social engineering attacks, we discovered a new type of impersonation attacks where attackers copy the profiles of legitimate users to create real-looking fake accounts that are used to illegally promote content on Twitter. Our analysis revealed that attackers target a wide range of users and anyone that has a Twitter account can be victim of such attacks.

Finally, we proposed an automated technique to detect impersonation attacks, that is able to detect 1,863 more impersonation attacks in our dataset (up from 166).

Our findings reveal a new type of privacy threat against the online image of users. Many surveys [12, 38] state that U.S. firms do background checks for job applicants that involve mining data from their online profiles. In this scenario, the doppelgänger bot attacks can potentially have a significant negative impact on the online image of users if the employer stumbles by mistake across the impersonating account.

7. REFERENCES

[1] Bing Maps API. http://www.microsoft.com/maps/developers/web.aspx.

[2] Get better results with less effort with Mechanical Turk Masters – The Mechanical Turk blog. http://bit.ly/112GmQI.

[3] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on Twitter. In *CEAS'10*.

[4] P. Bhattacharya, M. B. Zafar, N. Ganguly, S. Ghosh, and K. P. Gummadi. Inferring user interests in the twitter social network. In *RecSys '14*.

[5] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: Automated identity theft attacks on social networks. In *WWW'09*.

[6] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *NSDI'12*.

[7] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *IJCAI'03*.

[8] S. Corpus, 2015. http://anoncvs.postgresql.org/cvsweb.cgi/pgsql/src/backend/snowball/stopwords/.

[9] O. Goga. *Matching User Accounts Across Online Social Networks: Methods and Applications*. PhD thesis, Université Pierre et Marie Curie, 2014.

[10] O. Goga, P. Loiseau, R. Sommer, R. Teixeira, and K. Gummadi. On the reliability of profile matching across large online social networks. In *KDD*, 2015.

[11] B.-Z. He, C.-M. Chen, Y.-P. Su, and H.-M. Sun. A defence scheme against identity theft attack based on multiple social networks. *Expert Syst. Appl.*, 2014.

[12] Internetnews. Microsoft survey: Online 'reputation' counts, 2010. http://www.internetnews.com/webcontent/article.php/3861241/Microsoft+Survey+Online+Reputation+Counts.htm.

[13] L. Jin, H. Takabi, and J. B. Joshi. Towards active detection of identity clone attacks on online social networks. In *CODASPY '11*.

[14] A. M. Kakhki, C. Kliman-Silver, and A. Mislove. Iolaus: Securing online content rating systems. In *WWW'13*.

[15] M. Y. Kharaji, F. S. Rizi, and M. Khayyambashi. A new approach for finding cloned profiles in online social networks. *International Journal of Network Security*, 2014.

[16] Klout. Klout, 2014. http://klout.com/.

[17] G. Kontaxis, I. Polakis, S. Ioannidis, and E. Markatos. Detecting social network profile cloning. In *PERCOM'11*.

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 2004.

[19] Mediabistro. Was twitter right to suspend 'christopher walken'?, 2009. https://www.mediabistro.com/alltwitter/was-twitter-right-to-suspend-christopher-walken_b5021.

[20] A. Mislove, A. Post, K. P. Gummadi, and P. Druschel. Ostra: Leveraging trust to thwart unwanted communication. In *NSDI'08*.

[21] M. Mondal, B. Viswanath, A. Clement, P. Druschel, K. P. Gummadi, A. Mislove, and A. Post. Defending against large-scale crawls in online social networks. In *CoNEXT'12*.

[22] Nairobiwire. Sonko's facebook impersonator arrested, 2014. http://nairobiwire.com/2014/07/mike-sonko-arrested-swindling-public.html?utm_source=rss&utm_medium=rss&utm_campaign=mike-sonko-arrested-swindling-public.

[23] D. Perito, C. Castelluccia, M. Ali Kâafar, and P. Manils. How unique and traceable are usernames? In *Proceedings of the 11th Privacy Enhancing Technologies Symposium (PETS)*, 2011.

[24] Phash. http://www.phash.org.

[25] A. Post, V. Shah, and A. Mislove. Bazaar: Strengthening user reputations in online marketplaces. In *NSDI'11*.

[26] Seattlepi. Racism and twitter impersonation prompt lawsuit for kirkland teen, 2010. http://www.seattlepi.com/local/sound/article/Racism-and-Twitter-impersonation-prompt-lawsuit-893555.php.

[27] Social Intelligence Corp. http://www.socialintel.com/.

[28] Spokeo. http://www.spokeo.com/.

[29] T. Stein, E. Chen, and K. Mangla. Facebook immune system. In *SNS'11*.

[30] Turnto23. Impersonator continuously creating fake facebook profiles of a well known bakersfield pastor. http://www.turnto23.com/news/local-news/impersonator-continuously-creating-fake-facebook-profiles-of-a-bakersfield-pastor.

[31] Twitter. Explaining twitter's efforts to shut down spam. https://blog.twitter.com/2012/shutting-down-spammers, 2012.

[32] Twitter. Twitter reporting impersonation accounts, 2014. https://support.twitter.com/articles/20170142-reporting-impersonation-accounts.

[33] B. Viswanath, M. A. Bashir, M. Crovella, S. Guha, K. Gummadi, B. Krishnamurthy, and A. Mislove. Towards detecting anomalous user behavior in online social networks. In *USENIX Security'14*.

[34] B. Viswanath, M. A. Bashir, M. B. Zafar, L. Espin, K. P. Gummadi, and A. Mislove. Trulyfollowing: Discover twitter accounts with suspicious followers. http://trulyfollowing.app-ns.mpi-sws.org/, April 2012. Last accessed Sept 6, 2015.

[35] B. Viswanath, M. Mondal, A. Clement, P. Druschel, K. Gummadi, A. Mislove, and A. Post. Exploring the design space of social network-based sybil defenses. In *COMSNETS'12*.

[36] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove. An analysis of social network-based sybil defenses. In *SIGCOMM '10*.

[37] G. Wang, M. Mohanlal, C. Wilson, X. Wang, M. J. Metzger, H. Zheng, and B. Y. Zhao. Social turing tests: Crowdsourcing sybil detection. In *NDSS'13*.

[38] Wikibin. Employers using social networks for screening applicants, 2008. http://wikibin.org/articles/employers-using-social-networks-for-screening-applicants.html.

[39] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: Defending against sybil attacks via social networks. In *SIGCOMM '06*.

[40] C. M. Zhang and V. Paxson. Detecting and analyzing automated activity on twitter. In *PAM'11*.

APPENDIX

Here we first explain how we computed similarity between various attribute values (e.g., names and photos) of accounts and then describe the procedure we used to determine when two attribute values (e.g., two names or two photos) are "similar enough" to be deemed to represent the same entity.

A. SIMILARITY METRICS

Name similarity Previous work in the record linkage community showed that the *Jaro string distance* is the most suitable metric to compare similarity between names both in the offline and online worlds [7, 23]. So we use the Jaro distance to measure the similarity between user-names and screen-names.

Photo similarity Estimating photo similarity is tricky as the same photo can come in different formats. To measure the similarity of two photos while accounting for image transformations, we use two matching techniques: (*i*) *perceptual hashing*, a technique originally invented for identifying illegal copies of copyrighted content that works by reducing the image to a transformation-resilient "fingerprint" containing its salient characteristics [24] and (*ii*) *SIFT*, a size invariant algorithm that detects local features in an image and checks if two images are similar by counting the number of local features that match between two images [18]. We use two different algorithms for robustness. The perceptual hashing technique does not cope well with some images that are resized, while the SIFT algorithm does not cope well with computer generated images.

Location similarity For all profiles, we have the textual representations of the location, like the name of a city. Since social networks use different formats for this information, a simple textual comparison will be inaccurate. Instead, we convert the location to latitude/longitude coordinates by submitting them to the Bing API [1]. We then compute the similarity between two locations as the actual geodesic distance between the corresponding coordinates.

Bio similarity The similarity metric is simply the number of common words between the bios of two profiles after removing certain frequently used *stop words* (as is typically done in text retrieval applications). As the set of stop words, we use a popular corpus available for several languages [8].

B. SIMILARITY THRESHOLDS

Clearly the more similar two values of an attribute, the greater the chance that they refer to the same entity, be it a user-name or photo or location. To determine the threshold similarity beyond which two attribute values should be considered as representing the same entity, we rely on *human* annotators. Specifically, we attempt to determine when two attribute values are similar enough for humans to believe they represent the same entity.

We gathered human input by asking Amazon Mechanical Turk (AMT) users to evaluate whether pairs of attribute values represent the same entity or not. We randomly select 200 pairs of profiles and asked AMT users to annotate which attribute values represent the same entity and which do not. We followed the standard guidelines for gathering data from AMT workers [2].

For each attribute, we leverage the AMT experiments to select the similarity thresholds to declare two values as representing the same entity. Specifically, we select similarity thresholds, such that more than 90% of values that represent the same entity (as identified by AMT workers) and less than 10% of the values that represent different entities (as identified by AMT workers) have higher similarities. Consequently, we determine that two user-names or screen-names represent the same name if they have a similarity higher than 0.79, and 0.82 respectively. Two locations represent the same place if they are less than 70km apart. Two photos represent the same image if their SIFT similarity is higher than 0.11 and two bios describe the same user if they have more than 3 words in common.