On URL Changes and Handovers in Social Media

Hossein Hamooni, Nikan Chavoshi, and Abdullah Mueen

University of New Mexico

Abstract. Social media sites (e.g. Twitter and Pinterest) allow users to change the name of their accounts. A change in the account name results in a change in the URL of the user's homepage. We develop an algorithm that extracts such changes from streaming data and discover that a large number of social media accounts are performing *synchronous* and *collaborative* URL changes. We identify various types of URL changes such as handover, exchange, serial handover and loop exchange. All such behaviors are likely to be automated behavior and, thus, indicate accounts that are either already involved in malicious activities or being prepared to do so.

In this paper, we focus on *URL handovers* where a URL is released by a user and claimed by another user. We find interesting association between handovers and temporal, textual and network behaviors of users. We show several anomalous behaviors from suspicious users for each of these associations. We identify that URL handovers are instantaneous automated operations. We further investigate to understand the benefits of URL handovers, and identify that handovers are strongly associated with reusable internal links and successful avoidance of suspension by the host site. Our handover detection algorithm, which makes such analysis possible, is scalable to process millions of posts (e.g. tweets, pins) and shared publicly online.

1 Introduction

Social media sites, such as Twitter, Pinterest, Tumblr and Instagram allow users to broadcast messages and content (URLs, images, videos) publicly to their followers. Many of these sites allow users to change their homepage URLs by changing their account names. Users may need to change their URLs for many reasons, such as marriage, rebranding, business acquisition and closing, and so on. Such events are relatively rare for any human or business user in social media sites. Surprisingly, we observe unusually high numbers of URL changes in some Twitter users.

For example, we identify a user changing its URL 283 times in 78 days, equivalent to roughly one change every six hours. Some of the URLs, released and claimed in the same day, are shown in Figure 1. We identify an even more abnormal scenario where a URL, `twitter.com/MalumaOficiaI`, belonged to ten users in three months. Each user *handed over* the URL to another user collaboratively. In Figure 2, we show the sequence of handovers where nodes are user accounts and an arrow represents the direction of a handover. Such abnormal URL *handovers* are highly unlikely to be performed by a group of normal users, and most likely are generated by automated bots. As Twitter is one of the most popular social media site, we set to study such URL manipulating bots in Twitter.

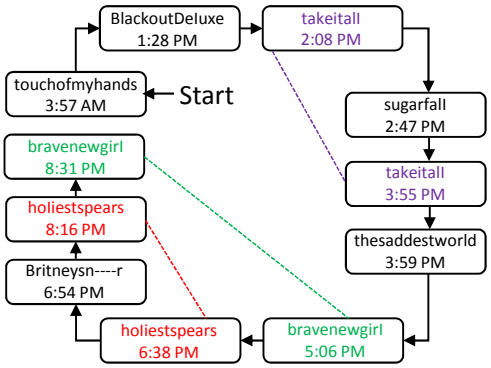

Fig. 1. A user (Twitter id: 2664619086) with ten URL changes on 7 November 2015. Some URLs are used more than once which form a loop of URLs. Repetitive URLs are connected by dotted lines.

There have been dozens of papers on mining Twitter data [12][11][16][5][14]. However, URL changes have not been studied with due diligence. An estimated 8.5% accounts in Twitter are bot accounts [15]. Our work shows that bot accounts carry out automated URL changes on a regular basis. Irregular URL changes waste resources on Twitter, create many broken URLs, and mislead Twitter users to spam account pages. These negative consequences of URL manipulation motivate this work.

In this paper, we investigate to discover *why* and *how* users make such abnormal changes. We develop a parallel algorithm using the map-reduce framework to identify URL changes in streaming data. Our algorithm is incremental and scalable to support social media similar to Twitter in size and traffic. We extract a set of 231K URL changes in Twitter over a period of three months (10/15-01/16). Note that we use only 1% of the data that Twitter publicly shares. We perform temporal, textual, and graph-based analyses on this data and discover several interesting facts about URL changes. Our findings are summarized below.

- Both URL changes and URL handovers are atomic operations.
- URLs that are handed over are more frequently mentioned by other users.
- URL handovers are associated with changes in content after the handover.
- URL handovers can be temporally correlated.
- URL changes are done in an organized and collaborative way by large groups of users.

The rest of the paper is organized as follows. We begin with a background section that provides examples of various types of URL changes and handovers. We describe our algorithm to discover URL changes and handovers in Section 3. We provide association analysis with temporal, textual and graph-based features in Section 4. We investigate *why* and *how* frequent handovers are performed in Section 5. We discuss related work in Section 6, and conclusion in Section 7.

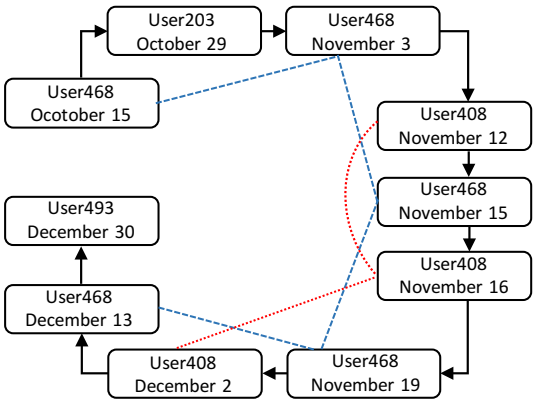

Fig. 2. The URL `twitter.com/paradisecameron` was handed over among four users nine times. User468 appears in the handover chain exactly every other time. The dashed lines connecting the same users show *loops* in this chain.

2 Background

We start with sufficient background information so readers are more familiar with URL changes and handovers.

URL Changes: Imagine a Twitter user with the name `tom_hanks`. The URL to the profile page of this user would be `twitter.com/tom_hanks`. If this user changes the screen name to `thanks`, the URL of its profile page will change to `twitter.com/thanks`. Such a change in the URL does not affect the social connections of `tom_hanks` in the Twitter network. All of the followers and followings of the account before and after the change remain the same. However, the URL change invalidates the old URL, which will no longer be accessible from other places on the Internet. URL changes also invalidate all of the old *mentions*[1] within Twitter, since mentions are the short form of URLs. In some social media, such as Pinterest, the old URL still functions because the site automatically redirects visitors to the new URL unless the old URL is taken by some other account.

URL Handovers: A URL handover consists of two URL changes, in which one user releases a URL and another user claims that URL. Let us consider the example in Figure 3 to describe URL handovers in reality. A user (user1) changes its screen name (URL) from `Tom` to `John`. The name `Tom` is then free on the network and can be claimed by any other user. If another user (user2) claims the name `Tom` by releasing its previous name `Bill`, a handover happens. We say the URL `twitter.com/Tom` has been handed over from user1 to user2. Here the user1 is the *from-account* and the user2 is the *to-account*. We also define the *handover lag* as the time duration between user1 releasing the URL `twitter.com/Tom` and the user2 claiming it.

[1] Twitter users can mention other users by using the '@' symbol which creates a link to the profile page of the mentioned user. For example `@thanks` is a link to the address `twitter.com/thanks`

In sites like Pinterest, the old URLs are redirected to the new ones. When a user changes URL, he does not need to worry about his followers who can still visit him via the old URL. However, in Twitter, old URLs are not redirected automatically. Therefore, a user often creates a new account to keep the old URL and leaves a pointer to the new URL. Thus, human users can do valid and legitimate handovers. However, in such handovers, one of the from-account or to-account should be inactive (e.g. no tweeting) after the handover under the assumption that no user wants to divide his followers among many accounts. We identify a *suspicious handover* if either of the following statements are true for a handover.

– Both of the from-account and to-account continue posting after the handover. **OR**
– Both of the from-account and to-account were active before the handover.

In the remainder of the paper, we will refer to a suspicious handover as simply a handover unless otherwise specified.

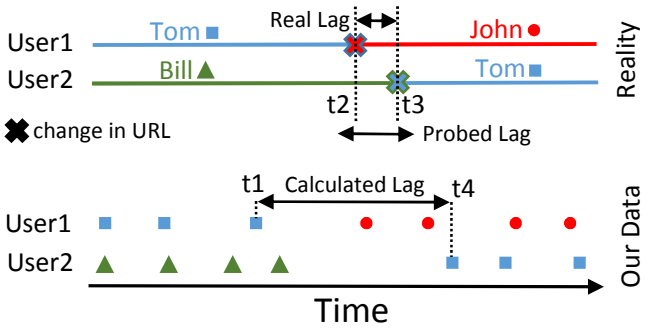

Fig. 3. $User_1$ handed over the URL `Tom` to $User_2$. The release time is t_2 and the claim time is t_3, and the real handover lag is $t_3 - t_2$. We calculate an upper bound, $t_4 - t_1$, for the handover lag based on the last tweet of $User_1$ at t_1 before the handover, and the first tweet from $User_2$ at t_4 after the handover.

2.1 Data Collection

We use the Twitter streaming API to collect data and produce a set of suspicious handovers. The Twitter streaming API caps the number of tweets sent to the client to a small fraction of the total volume of Tweets at any given moment [3]. We have never exceeded 48 Hz in practice. Our data collection module receives the tweets which contain the timestamp of the tweet, the URL, the user ID, the follower count, and some other information about the author of the tweet. The Twitter API provides tweets that satisfy a given condition, such as the tweet matches a given keyword, the tweet has a given topic, the tweet is made from a geo-location, or the tweet is authored by a specific set of users. We consider each tweet as a singleton object with a set of predefined features including timestamp, user ID, URL, geo-location, number of followers, number of accounts the user is following, and tweet content. In order to detect handovers, we consider three relevant features: the timestamp, the user ID, and the URL. Although a

user can change the URL of the account, the user ID is fixed for an account throughout the lifetime of the account.

We use different keyword filters to collect tweets for a week. We sort users based on their number of tweets and pick the top 40,000 as the seed for the rest of the data collection from the Twitter streaming API. We make the data collection process parallel on eight computers, each of which listens to 5,000 users continuously. This parallelization maximizes the number of tweets that can be collected from the streaming source.

We have started collecting data from Twitter on 15 October 2015 and continued until 31 December 2015. We have collected 130 million tweets with 5.7 million unique users[2] and 6 million unique URLs.

2.2 Complex Handovers

One URL change involves two URLs and one user account. One handover involves three URLs and two user accounts (Figure 3). However, URL changes and handovers can produce much more complex scenarios that are extremely unlikely to happen in a network that is built for independent social entities. A few complex scenarios are given below.

- A user changes the URL multiple times and forms a *chain* of URLs. An example is shown in Figure 1.
- Some chains of URLs create a *loop* when the user reclaims an old URL (i.e. $A \rightarrow B \rightarrow C \rightarrow D \rightarrow A$).
- A URL can be handed over in a chain from user A to user B, and then from user B to user C. This is a suspicious behavior because it shows that multiple accounts are interested in having the same URL. It gets more suspicious if each of these accounts own the URL for a short time.
- The handovers on a URL can also create a loop of users. This indicates that they either have a signaling mechanism to let each other know when the URL is free and ready to claim, or they are controlled by the same entity (Figure 2).

Although a single URL change may not be an abnormal behavior, the chance of all of the abnormal scenarios described above happening inadvertently is very low. We find a multitude of evidence showing that users are performing such changes and handovers automatically using computer programs.

3 Detecting URL Handovers

Since Twitter does not provide an event flag representing a URL change, we devise an algorithm to identify handovers based on the last tweet from the from-account and the first tweet from the to-account before and after the handover respectively. Figure 3 shows a toy example of handover detection using the streaming data provided by Twitter. Note that the handover lag can be calculated as the time between the last and the first tweets from the from-account and to-account, respectively.

[2] Although our data collection seed contained 40,000 users, in total we collected tweets from 5.7 million different users.

Computationally, handover detection is very similar to the *group-by order-by* queries for relational databases. We require grouping the tweets from the same URL and sorting the tweets for the same URL based in order of timestamps. We need to compare successive pairs of tweets from the same URL to detect change in their user IDs. Each such change in user ID corresponds to a handover. The process is further complicated by the scale of the data. A single processor cannot manage millions of tweets in reasonable time, guiding us to develop parallel solutions. We adopt map-reduce framework to distribute the computation and discuss our algorithm below.

Every map-reduce algorithm has two key components: a map function (mapper), and a reduce function (reducer). There can be other useful functions such as *filters* in a map-reduce framework. We discuss each of these components in this section. For clarity we define the input and the output of our map-reduce framework. The input is a set of tweets $T = \{tw_1, tw_2, \ldots, tw_n\}$ and the output is a set of URLs $U = \{url_1, url_2, \ldots, url_m\}$ where $url_i = \{(user_1, t_1, t_2), (user_2, t_3, t_4), \ldots, (user_k, t_{2k-1}, t_{2k})\}$, $k \geq 2$ and $t_j \leq t_{j+1}, \forall j\ 1 \leq j \leq 2k-1$.

Mapper: The map function in our framework converts a tweet object to an object that can be used by the reducers. It produces a set of key-value tuples where the key is the URL of the tweet and the value is the user ID plus two timestamps. Initially both timestamps are equal to the tweet timestamp, but they will be converted to a start timestamp and an end timestamp in the next steps, which reflect the period of time in which the URL was associated with each account. In other words, initially: $mapper(tweet_i) =< url_i, \{(user_i, t_i, t_i)\} >$

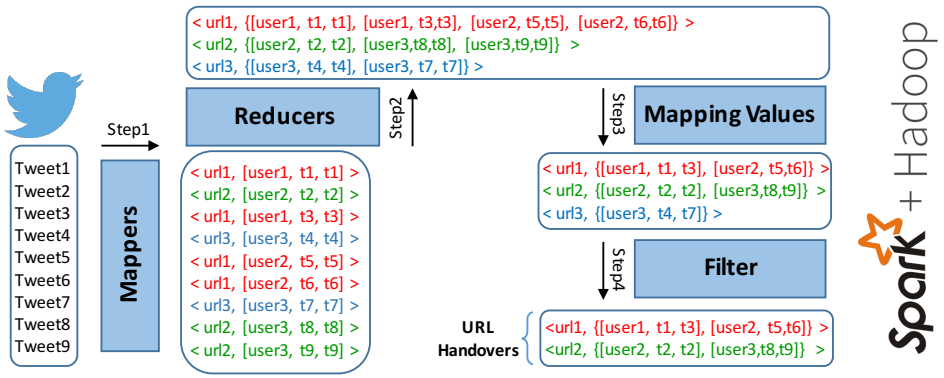

Fig. 4. The process of detecting URL handovers in the Twitter network. 2 URL handovers are detected from 9 tweet objects in this example.

Reducer: The reduce function plays the key role in our map-reduce framework. The map-reduce framework guarantees that all objects with the same URL will be reduced together and that they produce one last merged object. A merged object in our case is $< url_i, \{(user_a, t_1, t_1), (user_b, t_2, t_2), \ldots, (user_z, t_k, t_k)\} >$ where $t_i \leq t_{i+1}$. The reducer function takes two key-value tuples as the input and produces one merged tuple

as the output. As mentioned earlier, the value part is made up of a set of user IDs, each of which has a starting and an end time. The reducer function takes these two lists and creates a sorted output based on the start times of each object[3]. Since all the lists have just one element at the beginning of the reduce task, they are trivially sorted. As the reducer combines them, the merged lists are also sorted. In other words, the input to the reducer function is two sorted list of length m and n, and it just takes $O(m+n)$ steps for the reducer to sort them by using the merge sort approach.

Mergers: After the reducer produces a sorted list of user IDs with timestamps for each URL, we need to merge all the consecutive tweets with the same user ID to create a shorter list for each URL where there is a start and an end time for each user ID. For example, if the output of the reducer for a URL is $\{(A,1,1),(A,4,4),(B,7,7),(B,9,9),(A,15,15),(A,20,20)\}$ then the output of running the map value function would be: $\{(A,1,4),(B,7,9),(A,15,20)\}$

Filter: At this stage, we have lists of users associated with every URL in our map-reduce framework. However, we are not interested in detecting URLs that are only associated with one user ID. To filter out these URLs from the output of the map-reduce framework, we use a simple *filter* function. This function checks the length of the list of the users and outputs only the lists that have more than one user ID.

Our algorithms have detected a total of 13,831 URL handovers involving 12,326 unique URLs and 21,257 unique users in the 78 days of data collection. We also detect 231,800 users who changed their URL at least once in this time period. We share the entire set of handovers and the source code to detect them in [1].

4 Handover Analysis

In this section we analyze the handovers detected by our method to observe several suspicious behaviors related to the the user's temporal profile, tweet content, and the frequency of URL changes. We also discuss how multiple users can be connected through handovers. We finally analyze the handover lags to show that the handovers are automated.

4.1 Temporal Profile

We investigate questions related to the temporal profile of a user involved in a handover. We extract hourly time series of every user in every handover for 78 days. As mentioned in Section 2.1, each tweet object contains the timestamp of that tweet in millisecond resolution, and the number of followers of the user at that time. We construct hourly *activity time series* of every user by aggregating the total number of activities the user performs in each hour. Note that Twitter does not guarantee to provide all of the tweets of a user; therefore we achieve a lower bound time series on user activity. As we shall see, such partial data is enough to reveal abusive behaviors on Twitter.

Similarly, we create the follower time series of a user which shows the changes in the popularity of that user. We receive follower information embedded in the tweets,

[3] At this stage of the algorithm, the start time and the end time of the objects are still the same

yielding some unevenly spaced measurements of the follower counts of a user. We interpolate the in-between follower counts by the last received count with an assumption that follower counts change very slowly (particularly for non-popular and old accounts).

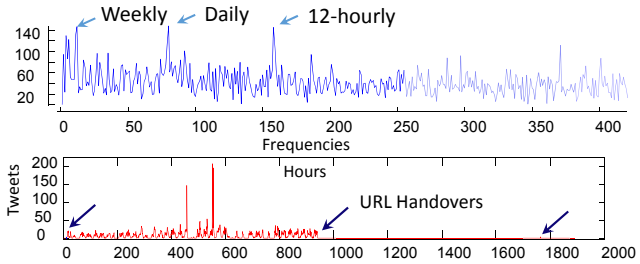

Fig. 5. (top) Frequency distribution of hourly count of handovers. (bottom) An example user with daily periodicity and a strong activity association with handover.

Activity Association: We first consider the distribution of handovers over 11 weeks. We only consider handovers that have less than a day of calculated lag. This ensures that the real lag is at most 24 hours, a reasonable value. In Figure 5(top) we show the frequency distribution of the hourly aggregates of handover counts over 1890 hours. We use the method in [19] and identify three sharp peaks pointing to weekly, daily and 12-hourly periodicity. Figure 5(bottom) shows an example activity sequence of a user with daily and weekly periodicity.

We investigate if the handovers are related to a change in activity patterns. We check if the average activity levels of a user in the 6-hour windows before and after a handover are significantly different. 91% of the times the difference is less than 1 tweet an hour. Therefore, we conclude there is no significant change in the activity level before and after the handovers. However, exceptions are possible. Figure 5(bottom) shows an example where the activity starts and stops with handovers.

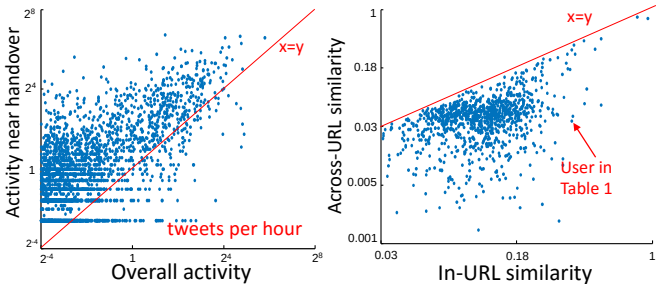

Fig. 6. Comparing the normal activity of a user with its activity near the handovers. 97.4% of the users have higher activity-per-hour around handovers. Both x-axis and y-axis are in log scale. (right) The plot shows how the content of the tweets changes with URL changes.

Next, we consider the association between handovers and the activity level around them. We calculate the average activity-per-hour for every from-account in the 6-hour

window just before releasing the URL, and the same for every to-account in the 6-hour window just after claiming the URL. We compare such pre- and post-handover activities with the average activity-per-hour of the user, calculated over the entire duration of data collection. We identify a significant difference in activity level before and after handover. Quantitatively, 97.4% of the users are more active than usual when performing handovers (Figure 6 (left)).

Cross-user Association: We further consider cross-user associations in temporal profiles of handovers. We use standard time series motif discovery tools [13] to identify the most frequent activity time series. Note that the expected similarity in activity time series between two users for 11 weeks is almost zero. Interestingly, we identify a motif of three users who have almost identical activity patterns with an average correlation coefficient of 0.96. Furthermore, the accounts perform URL handovers within the same hour in the same manner (e.g. to-from-to). The motifs are shown in Figure 7. The URLs that were handed over by these accounts are all related to celebrities such as MacMiller, Rihanna, Drake, Megan Fox and Lil Wayne.

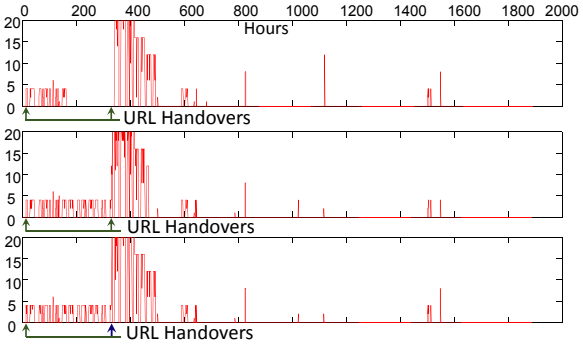

Fig. 7. Three accounts with almost identical activity profile and correlated handovers. Handovers initiate change in activity patterns.

We consider the motif as a significant discovery because it reveals that offenders work in correlation, possibly using the same codebase, and that they hand over at the same time to swap or pass URLs that they do not want to lose. In the future, we will investigate how to scale handover detection in real time so we can track the interest areas of the offenders to take countermeasures.

4.2 Content Profile

Users tweet about various topics. The topic of a tweet can be determined by analyzing the keywords in it. We first remove all useless words like *is, are, the, to, from, RT*[4], ..., and then process the tweet content. We use the content of the tweets to determine the similarity between two tweets, and also two sets of tweets. We use the Jaccard similarity coefficient [20] as our metric. The similarity between two sets of tweets X and Y can be defined as the average similarity of all pairs of tweets in them. We use this measure

[4] The word "RT" appears at the beginning of all retweets and has nothing to do with the content.

to calculate the similarity between two Twitter users, or between two different periods of time (before and after the handover) for the same user to profile content changes around handovers.

Table 1. The tweets from the same user with 2 URLs. The user change its URL from `zflexins` to `loveyorslf` on 11 December 2015. All the tweets of the left column are about Justin Bieber, and the ones in the right column are about Harry Styles (both are famous singers). The average in-URL similarity is 0.35, while the across-URL similarity is 0.03.

www.twitter.com/zflexins	**www.twitter.com/loveyorslf**
RT @justinbieber:UK! Tonight on @CapitalOfficial from 7pm 'Justin Bieber's Capital Album Party Replay'. Hear the tracks from #Purpose	harry styles coisa mais linda gente!!!
RT @JBCrewdotcom: Another photo of Justin Bieber with a fan at the M&G in Tokyo, Japan yesterday. (December 4) https://t.co/ofAYAjzP1M	harry s,tao precioso gente como vcs nao gostam dele???????? https://t.co/o0x2DG38JI
RT @JBCrewdotcom: Another video of Justin Bieber singing at a restaurant in Japan today. (December 5) https://t.co/jZqaMaezrO	vou tweetar video de harry stylesN
RT @favjarbara: interviewer: what do you think about justin bieber's relationships?bp: hahaha he's mine	harry w kendall eu to gRITANDO AQUI, OPSSS https://t.co/MURzVWnc0Q
RT @NME: Justin Bieber announces UK Arena tour dates for 2016 https://t.co/ECsRUqEPxk	@KendallJBrasil: 31/12- Mais fotos de Kendall e Harry Styles em St. Barts, Frana. https://t.co/CytM8Hixk

Content Association: We consider the content of tweets for a user before and after the URL change[5] to see *if the content changes with the change in URL*. Let T_1 and T_2 be the sets of tweets of a user from its first and second URL respectively. We calculate the in-URL similarity as the weighted average of $Sim(T_1, T_1)$ and $Sim(T_2, T_2)$, and the across-URL similarity as $Sim(T_1, T_2)$. For example, Table 1 shows the tweets of a user with two different URLs: `zflexins` and `loveyorslf`. The user tweeted 98 times with the first URL about Justin Bieber, and 94 times with the second URL about Harry Styles. These are two of the most popular celebrities in Twitter with millions of followers. There is a clear change in the topic of the tweets after the URL change. The average in-URL similarity for this user is 0.35 while the across-URL similarity is 0.03. It is humorous that the content of the first tweet after URL change is: `RIP zflexins`. Both of these URLs are now associated with some other accounts.

In order to check this hypothesis for other users, we select a random set of handover users that have exactly two URLs associated with them in our dataset. We filter out the users for which $|T_1| < 5$ or $|T_2| < 5$, and finally come up with 1,051 users. Figure 6(right) shows the comparison of in-URL with across-URL similarity for each of these users. We have 100% of the users with higher in-URL similarity than the across-URL

[5] We specifically are interested in URL changes that were part of a handover.

similarity. It means that the overall topic of the tweets changes when a user changes its URL, especially if that URL change is a part of URL handover.

4.3 URL Change Analysis

In this experiment, we check if the frequency of URL changes (average number of URL changes per day) of a user has any relation with the probability of that user being involved in a URL handover, since we believe both a high number of URL changes and being involved in a handover are suspicious behaviors. There are 231,800 users in our dataset which changed their URL at least once during our data collection. Figure 8 (I) shows the percentage of these users for different frequencies of URL changing. The probability of a user being involved in a handover given the frequency it has changed its URL is shown in Figure 8 (II). The higher the change frequency, the larger the probability of performing handovers. The reason why we do not show users with more than 9 URL change frequency on the left side is that they comprise less than 1% of our dataset. However, we can say almost all of this 1% have done a URL handover by looking at the right hand side of the figure.

4.4 Connectivity Profile

We create a bipartite graph where the left side is the set of all users and the right side is the set of all URLs, and a link between a user u_i and a URL v_j exists if u_i owned v_j at some point in our dataset, and the URL was used for a handover. A handover is defined as a subgraph with three nodes and two edges in which two nodes from the user side have a link to the same node on the URL side.

We use the classic co-clustering approach to identify clusters in the user-URL bipartite graph [9]. Any balanced cluster with more than three members points to organized teamwork by the accounts. It is very unlikely that a large balanced cluster was created in this bipartite graph by accident.

We find a cluster of size 2,273 (1,205 users + 1,068 URLs) which has 2,399 edges. The average degree of each node in this cluster is 2.11. It is highly unlikely that such a cluster is formed randomly, and thus this cluster supports our original hypothesis that correlated and frequent handovers are signatures of automated accounts managed by the same entity. If we had more data, we could have identified more handovers, and the cluster could have been much larger. About 6% of the all users that has performed suspicious behavior (URL handover) are in this particular cluster. This again proves that our suspicion is correct beyond a doubt since such a huge cluster can not be formed randomly.

If we consider all of the clusters with more than three members (non-trivial handover clusters), they cover 31% of all users who have been involved in handovers. Although any URL handover is a suspicious behavior, this provides us with additional evidence of misbehavior from this 31%. We believe that the majority of the other 69% also belongs to a non-trivial cluster, but we are not able to catch them due to lack of data. As we show in the next section, social media sites are slow in suspending such offenders. We have detected thousands of automated spammers, even without the complete dataset, and yet Twitter has suspended only a fourth of them in six weeks.

These users who are doing URL handovers usually change their URLs more than once. Not all of their URLs are included in the discussed bipartite graph since we just add the URLs which have been handed over. If we include all of the URLs of the users who have done a handovers (even the URLs that have not been used in any handover so far) in our graph and re-cluster, the biggest cluster would have 1,205 users and 6,040 URLs. These newly added URLs are good candidates for our active probing technique (future work) since they belonged to a suspicious user at some point in the past.

4.5 Lag Profile

To examine whether these handovers are organized from a central source as opposed to independent actions, we perform an analysis on handover time-lag. The real lag between releasing a URL and claiming it back is not detectable from the publicly available tweets. Our active probing tool, which is not scalable because of a capacity limit set by Twitter, estimates handover lag at most an hour longer than the real lag.

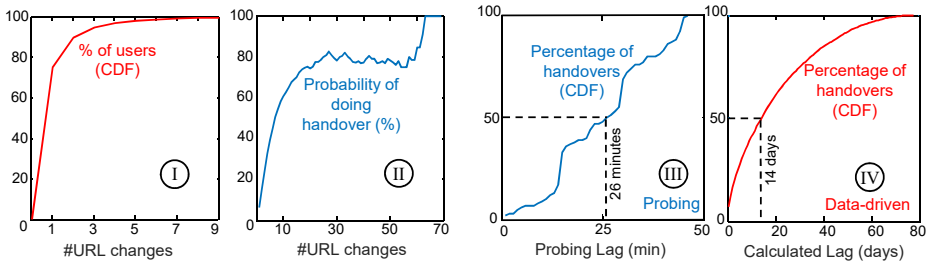

Fig. 8. (I) Percentage of users (out of 231,800 users) based on the frequency of changing URL. (II) The probability of a user doing a URL handover given the URL change frequency. The probability reaches 1 for a user with frequency higher than 68 (almost one URL change every day). (III) The distribution of handovers based on their lags calculated using our probing technique. (IV) The distribution of handovers based on their lags calculated using the data-driven approach. 50^{th} percentiles are shown in both III and IV

We can only probe 180 users and/or URLs every 15 minutes. We start probing every day with a list of 100 users who we know have high numbers of URL changes (based on our dataset). We have done this experiment for 8 consecutive days and observed 210 handovers. Figure 8 (III) shows the CDF of the percentage of handovers for different lags. The sharp increases at minutes 15, 30, and 45 are the result of the discontinuities in the probing algorithm caused by Twitter API limitations imposed on our algorithm.The approximately linear CDF illustrates the remarkable fact that handovers are instantaneous operations. We can verify this claim by simulating a set of instantaneous handovers spread uniformly over time and applying our probing algorithm to calculate an estimated CDF. The estimated CDF is, indeed, a line and the slope of the line is very similar to what we have observed.

This analysis formed the basis of our data-driven detection process. In the data-driven detection process, we can only detect a handover if the pair of accounts tweet

something before and after the handover, and Twitter provides us the tweets. Under such stringent condition, the lags we calculate are weak upper bounds of the real lags.

We show the CDF of the handover lags detected by the data-driven technique in Figure 8 (IV). Although the data-driven process detects larger lags compared to the real lag, since we know from this analysis that the handovers are mostly instantaneous operations performed by automated programs, we trust that the handovers detected using the data-driven technique are highly suspicious. Also note that the lag for half of the handovers we find is less than 14 days. Therefore, if a URL is not claimed after few days of releasing, it (probably) will not ever be claimed.

5 Why Handovers?

Such a magnitude of automated URL changes must have good reasons behind. In this section, we discuss association of handovers with potential benefits such as obtaining human followers and avoiding suspension, and thus attempt to answer the question *why are handovers so frequent?*.

Mentions and External URLs: Although URL changes do not impact the internal connectivity among users (who follows whom), they have a direct impact on URLs linked from external web pages. It also affects the links created by *mentions* within Twitter. For example, when user1 mentions user2 as `@DavidW` (whose screen name is *DavidW*) in a tweet, Twitter creates the URL `twitter.com/DavidW` and embeds it in the tweet content. If user2 hands over this URL to user3, the mention DavidW would point to user3's profile page. Thus, thousands of mentions within Twitter are being abused by URL changes.

Our hypothesis is that the miscreants change URLs frequently to fool users in visiting different pages every time they follow the same mention. The motivation is to increase the chance of getting a human visitor or follower in the process.

To test this hypothesis, we use the Twitter Advanced Search page in which one can search for the mentions of a certain screen name (i.e. URL). We count the number of mentions a URL receives in the first fifteen pages of the search result. Figure 9 (left) shows the percentage of URLs based on the number of mentions for 1000 random URLs and 1000 handover URLs. The URLs that have been handed over have a higher number of mentions compared to random URLs. The average number of mentions for handover URLs is 80 compared to 22 for random URLs.

Suspension: Twitter suspends accounts that violate some of its rules [2]. Twitter rules says, *Creating multiple accounts with overlapping uses or in order to evade the temporary or permanent suspension of a separate account is not allowed.* Handovers are strong signals of overlapping uses, hence, handover accounts are violating the Twitter rules.

We have detected URL changes and handovers until 31 December 2015. In order to see whether or not doing the handover has any impact on the suspension of the involved users, we check the status of all handover users almost every week from 1 January 2016 to 8 February 8 2016. Each point in Figure 9 (right) shows the percentage of the handover accounts being suspended by Twitter until that day. The interesting point is

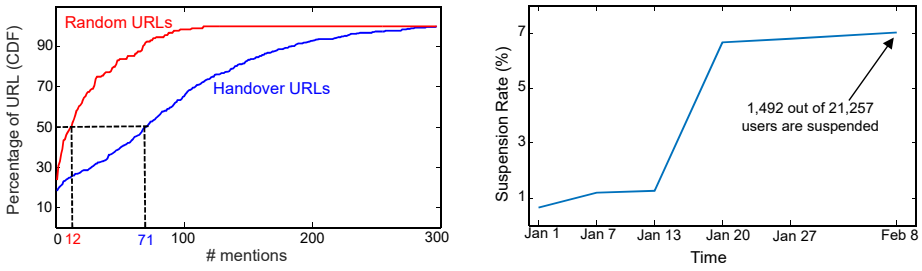

Fig. 9. (left) The percentage of URLs based on the number of mentions for random URLs and handover URLs. (right) Twitter suspension rate of handover users.

that although these users had done many URL handovers in our data collection period, just 0.6% of them were suspended by 1 January 2016. In Section 4.4, we mentioned that we have additional strong evidence of 31% of the handover users being suspicious, and still just 7% of these accounts are suspended by Twitter at the time of writing, while we had found these suspicious users weeks earlier.

6 Related Work

URL changes and handovers have been actively performed by users in social media. To the best of our knowledge, our work is the first to investigate the association between these activities and abuse in social media. Research has been done on other various aspects of abuse in social media including account hijacking [17], trolling [6], faking [4] and trafficking fraudulent accounts [18]. All of these works provide an important perspective on how fraudsters, merchants and abusers are manipulating social media for their own benefit. Our work considers URL handovers in the same manner. There are several works on bot and automated user account detection in social media using data mining techniques. In [8], the authors have modeled the inter-arrival time between tweets to understand bot behavior. In [7] and [10], supervised techniques are used to detect bots at registration time.

7 Conclusion

We develop methods to detect URL handovers between accounts in social media using publicly available data. We perform an in-depth analysis on the users who perform URL changes and handovers and identify several interesting characteristics. Collaborative abusers exploit this ability to change their URLs in social media to trick regular human users into following spam accounts. Our data analysis discovers automated and collaborative handovers in temporal and connectivity profiles of these users, and provides useful insights into how the abusers are operating. In future work we will develop active prevention based on these insights by predicting which users are going to do a URL handover.

References

1. Project repository. https://cs.unm.edu/~hamooni/handover.
2. The twitter rules. https://support.twitter.com/articles/18311.
3. Twitter streaming api. https://dev.twitter.com/tags/streaming-api.
4. L. Akoglu, R. Chandy, and C. Faloutsos. Opinion fraud detection in online reviews by network effects. In *Icwsm*, pages 2–11, 2013.
5. S. Asur and B. A. Huberman. Predicting the future with social media. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 492–499. IEEE, 8 2010.
6. J. Cheng, C. Danescu-Niculescu-Mizil, and J. Leskovec. Antisocial behavior in online discussion communities. In *Proceedings of ICWSM*, 2015.
7. Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Transactions on Dependable and Secure Computing*, 9(6):811–824, 11 2012.
8. A. F. Costa, Y. Yamaguchi, A. J. M. Traina, C. Traina Jr., and C. Faloutsos. Rsc: Mining and modeling temporal activity in social media. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15. ACM, 2015.
9. I. S. Dhillon. Co-clustering documents and words using bipartite co-clustering documents and words using bipartite spectral graph partitioning. In *Proc of 7th ACM SIGKDD Conf*, pages 269–274, 2001.
10. K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '10*, page 435. ACM Press, 7 2010.
11. H. Li, A. Mukherjee, B. Liu, R. Kornfield, and S. Emery. Detecting campaign promoters on twitter using markov random fields. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 290–299, 2014.
12. Y. Matsubara, Y. Sakurai, N. Ueda, and M. Yoshikawa. Fast and exact monitoring of co-evolving data streams. In *2014 IEEE International Conference on Data Mining*, pages 390–399. IEEE, 12 2014.
13. A. Mueen. Enumeration of time series motifs of all lengths. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, ICDM, pages 547–556, 2013.
14. J. Ratkiewicz, M. Conover, M. Meiss, B. Goncalves, A. Flammini, and F. Menczer. Detecting and tracking political abuse in social media, 2011.
15. V. Subrahmanian, A. Azaria, S. Durst, V. Kagan, A. Galstyan, K. Lerman, L. Zhu, E. Ferrara, A. Flammini, F. Menczer, R. Waltzman, A. Stevens, A. Dekhtyar, S. Gao, T. Hogg, F. Kooti, Y. Liu, O. Varol, P. Shiralkar, V. Vydiswaran, Q. Mei, and T. Huang. The darpa twitter bot challenge. IEEE Computer (In press), 2016.
16. K. Thomas, C. Grier, D. Song, and V. Paxson. Suspended accounts in retrospect: an analysis of twitter spam. In *Proceedings of the 2011 ACM* , IMC '11, pages 243–258, 2011.
17. K. Thomas, F. Li, C. Grier, and V. Paxson. Consequences of connectivity: Characterizing account hijacking on twitter. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14*, pages 489–500. ACM Press, 11 2014.
18. K. Thomas, V. Paxson, D. Mccoy, and C. Grier. Trafficking fraudulent accounts : The role of the underground market in twitter spam and abuse trafficking fraudulent accounts :. In *USENIX Security Symposium*, SEC'13, pages 195–210, 2013.
19. M. Vlachos, D. Gunopulos, and G. Das. Rotation invariant distance measures for trajectories. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, page 707. ACM Press, 8 2004.

20. Wikipedia. Jaccard index — wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Jaccard_index&oldid=688763411.