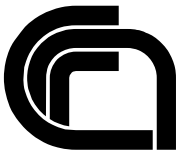

Consiglio Nazionale delle Ricerche

# A Fake Follower Story: improving fake accounts detection on Twitter

S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi

IIT TR-03/2014

## Technical report

**Febbraio 2014**

**Istituto di Informatica e Telematica**

# A Fake Follower Story: improving fake accounts detection on Twitter

Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi

**Abstract**—*Fake followers* are those Twitter accounts created to inflate the number of followers of a target account. Fake followers are dangerous to the social platform and beyond, since they may alter concepts like popularity and influence in the Twittersphere—hence impacting on economy, politics, and Society. In this paper, we contribute along different dimensions. First, we review some of the most relevant existing features and rules (proposed by Academia and Media) for anomalous Twitter accounts detection. Second, we create a gold standard of verified human and fake accounts. Then, we exploit the gold standard to train a set of machine-learning classifiers built over the reviewed rules and features. Most of the rules provided by Media provide unsatisfactory performance in revealing fake followers, while features provided by Academia for spam detection result in good performance. Building on the most promising features, we optimise the classifiers both in terms of reduction of overfitting and costs for gathering the data needed to compute the features. The final result is a "Class A" classifier, that is general enough to thwart overfitting and that uses the less costly features, while being able to correctly classify more than 95% of the accounts of the training set.
The findings reported in this paper, other than being supported by a thorough experimental methodology and being interesting on their own, also pave the way for further investigation.

**Index Terms**—Twitter, fake followers detection, gold standard, machine learning

## 1 Introduction

Originally started as a personal microblogging site, Twitter has been transformed by common use to an information publishing venue. As of December, 2013, statistics reported 645 million of Twitter subscribers, with some 300 billion ($> 2^{38}$) of tweets sent [16]. Twitter annual advertising revenue in 2013 has been estimated to $405,500,000 [19]. Popular public characters, such as actors and singers, as well as traditional mass media (radio, TV, and newspapers) use Twitter as a new media channel. Politicians commit a notable part of their campaigning to their Twitter home pages, as it happened for the last US presidential and Italian general election events [23]. As a consequence, the Twitter platform has raised the attention of Industry and Business as well, with some (if not all) of the most famous brands massively using this platform for business promotion [2].

Such a versatility and spread of use have made Twitter the ideal arena for proliferation of anomalous accounts, that behave in unconventional ways. Academia has focused its attention on *spammers*, that is those accounts actively putting their efforts in spreading malware, sending spam, and advertising activities of doubtful legality [4], [13], [22], [26]. Very often, to enhance the effectiveness of spammers, they are armed with automated twitting programs, known as bots. Such automated pieces of software could be designed and used to post legitimate tweets as well—such as news updates.

In the recent past, media have started reporting that the accounts of politicians, celebrities, and popular brands featured a suspicious inflation of followers [5], [6], [14]. So called *fake followers* correspond to Twitter accounts specifically exploited to increase the number of followers of a target account. As an example, during the 2012 US election campaign, the Twitter account of challenger Romney experienced a sudden jump in the number of followers. The great majority of them has been later claimed to be fake [14]. Similarly, before the last general Italian elections took place (on the 25th of February 2013), online blogs and newspapers had reported statistical data over a supposed percentage of fakes of major candidates [24].

At a first glance, acquiring fake followers could seem a practice limited to foster one's vanity—a maybe questionable, but harmless practice. However, a deeper analysis reveals that artificially inflating the number of followers can also be finalized to make an account more trustworthy and influential, in order to stand from the crowd and to attract other genuine followers. Recently, it seems that banks and financial institution are analyzing Twitter and Facebook accounts of loan applicants before granting the loan. In particular, they take in account the number of friends and how their interactions influence their decisions [12]. Indeed, the more the supposed influence, the more those accounts with lots of followers will likely interfere with the genuine followers. Then, having a dependable and popular profile could definitely help in obtaining credit from a bank or even successfully engage in social lending. Similarly, if the practice of buying fake

- *Roberto Di Pietro is with IIT-CNR, Pisa, Italy and Dept. of Maths and Physics, University of Roma Tre, Italy*
  *E-mail: dipietro@mat.uniroma3.it*
- *Stefano Cresci, Marinella Petrocchi, Angelo Spognardi and Maurizio Tesconi are with IIT-CNR, Pisa, Italy*
  *E-mail: name.surname@iit.cnr.it*

followers is adopted by spammers, it can act as a way to post more authoritative messages and launch more effective advertising campaigns. The outcome could be the alteration of the concepts of popularity and influence in the Twittersphere, leading to formation of fictitious public opinion and possible impact on real world economy and Society. That is why fake followers detection is an issue worth addressing.

Fake followers detection seems to be an easy task for many bloggers, that suggest their "golden rules" and provide a series of criteria, to be used as red flags to classify a twitter account behavior. However, such rules are usually paired neither with analytic algorithms to aggregate them, nor with validation mechanisms. As for Academia, researchers have focused mainly on spam and bot detection, with brilliant results characterizing Twitter accounts based on their (non)-human features. To the best of our knowledge, however, there is a lack of analysis on fake followers characterization and detection. Moreover, most of the scientific studies generates a classifier to discriminate twitter accounts. The classifier is built as follows: researchers manually test the nature of a set of accounts, that, upon testing, becomes the training set for a machine learning-based classifier. The intuitive drawback is that humans are not error-free, and, thus, the manual classification phase is both error prone and time consuming. In this paper, detection of fake followers follows a classifier-based approach too. However, we aim at overcoming the above drawbacks of a manual construction of the training set, as clarified in the following.

### *Contributions*

The goal of this work is to shed light on the phenomenon of fake followers, aiming at overcoming current limitations in their classification and detection. In particular, we provide the following contributions.

First, we provide a reference set of Twitter accounts, a so-called *gold standard*, where humans and fakes are known a priori.

Second, we test known methodologies for bot and spam detection on our gold standard. In particular, we apply to Twitter accounts in our reference set algorithms based on 1) single classification rules proposed by bloggers, and 2) feature sets proposed in the literature. The outcome of the analysis leads us to conclude that fake followers detection deserves specialized mechanisms: in particular, algorithms based on classification rules do not succeed in detecting the fakes in our reference dataset. Instead, classifiers based on features sets for spambot detection work quite well also for fake followers detection.

Third, we classify all the investigated rules and features based on the cost required for gathering the data needed to compute them.

Then, we define an optimized classifier that makes use of the the less costly features, while being able to correctly classify more than 95% of the accounts, on the training set.

### *Roadmap*

The remainder of this paper is organized as follows. Section 2 considers related work in the area of Twitter spam and bot detection. Section 3 describes our reference dataset. In Section 4, we concentrate on a set of criteria for fake followers detection promoted by social media firms and we present the results of the application of such criteria over the reference dataset. In Section 5, we examine features used in past work for spam detection of Twitter accounts and we assess the performance of a set of machine-learning classifiers that we have trained over the reference dataset with those features. In Section 6 we compute the cost for extracting the features our classifiers are based upon. An optimized classifier is provided yielding a good balance between fake detection capability and crawling cost. Finally, Section 7 concludes the paper.

## 2 Related Work

In this section, we revise recent work in the area of spam and automated detection of user behavior on Twitter.

The work in [22] presents an analysis on how spammers operate on Facebook, Twitter, and MySpace. For data gathering, the authors created a large set of honey profiles on three social platforms, logged the kind of contacts and messages that they received, and manually analyzed the collected data. The analysis reported that the suspicious accounts shared some common traits, formalized by the authors in a set of features. They served as input to a machine learning based classifier [8], to take automatic decisions over a large set of unknown accounts. Impressively, such an approach led to the detection of more than 15,000 spam profiles, that Twitter promptly deleted.

In [26], the authors observed that the more researchers and engineers make progress in keeping Twitter a spam-free online community, the more Twitter spammers are evolving to evade existing detection techniques. They also proposed a taxonomy of criteria for detecting Twitter spammers. A series of experiments showed how the newly designed criteria feature a detection rate higher than existing ones.

Authors of [4] classify Twitter accounts in three classes: humans, bot, and cyborgs. The latter class represents either a bot-assisted humans or an human-assisted bots. About six thousands accounts have been manually classified to create a training set and a test set, each one with 1,000 accounts for each of the three classes. The authors build their classifier based on four components: an entropy component that evaluates the timing regularity of an account tweets; a spam filter to detect spam tweets; an account property analyzer to extract additional information; and a decision maker component. This last one determines the class of a given account combining the outputs of the other three parts with a multiclass linear discriminant (LDA) analysis method.

Work in [21] makes an interesting analysis on the underground phenomenon of so called Twitter Account Markets, i.e., websites offering their subscribers to provide followers in exchange for a fee, and to spread promotional tweets on their behalf. The authors list a series of criteria that are helpful to detect Account Markets clients that pay for acquiring followers and spam with debatable tweets. In addition, criteria for detecting the spammer victims are also highlighted. Results of the analysis reveal a surprising and alarming business behind this phenomenon.

A series of reports published by the firm *digitalevaluations.com* [3] have attracted the attention of Italian and European newspapers and magazines, raising doubts on the Twitter popularity of politicians and leading international companies. A number of criteria, inspired by common sense and denoting *human* behavior, are listed in the reports and used to evaluate a sampling of the followers of selected accounts. For each criterion satisfied by a follower, a *human* score is assigned. For each not fulfilled criterion, either a *bot* or *neutral* score is assigned to the account. According to the total score achieved, Twitter followers are classified either as humans, as bots or as neutral (in this last case, there is no sufficient information to assess their nature), providing a quality score of the effective influence of the followed account. The results in [3] lack a validation phase.

Beside academic work, we assisted to the proliferation of online blogger and columnist posts, listing their own criteria for Twitter bots detection. As an example, a well-known blogger in [18] indicates as possible bots-like distinctive signals the fact that bots accounts: 1) have usually a huge amount of following and a small amount of followers; 2) tweet the same thing to everybody; and, 3) play the follow/unfollow game, meaning that they follow and then unfollow an account usually within 24 hours. Criteria advertised by online blogs are mainly based on common sense and the authors usually do not even suggest how to validate them.

Finally, some companies specialized in social media analysis, like [17], [20], offer online services to analyze how much a Twitter account is *genuine* in terms of its followers. However, the criteria used for the analysis are not publicly disclosed and just partially deducible from information available on their web sites.

In the next sections of the paper, we introduce a Twitter account dataset that we used to evaluate the performance on detecting fake accounts of five of the cited work, namely [3], [17], [18], [22], [26]. We are aware that this selection is not exhaustive. However, it considers a huge collection of criteria, that we further leverage for our reasoning on fake follower detection. It is worth noticing how other works for spam detection, like [13], [27], base their results on subsets, or on slightly modified versions, of the criteria considered by our selected set of works.

We distinguish the above 5 works in two main categories, considering the type of algorithm used for the detection: decision rule based or feature set based. The first type of algorithms relies on a list of rules each account has to be checked against: considering the output of each check, the algorithm distinguish between the possible classes. The second type of algorithms extracts from a set of pre-classified accounts some properties that uses to learn a model able to distinguish between the possible classes. The first type of algorithms has been proposed for fake/bot account detection, while the second type has been used for spam account detection. We detail and evaluate the first type of algorithms in Section 4 and the second type in Section 5.

## 3 Reference Datasets

In this section, we present the datasets of Twitter accounts (our "gold standard") that we used to conduct our empirical study and that will be used throughout the paper. We detail how we collected each of them and how we verified if they were genuine humans or fakes. Despite the final size of the gold standard, to perform our research, we altogether crawled 9 millions of Twitter accounts and about 3 millions of tweets.

### 3.1 The Fake Project

The Fake Project started its activities on December 12, 2012, with the creation of the Twitter account @TheFakeProject. Its profile reports the following motto: *Follow me only if you are NOT a fake* and explains that the initiative is linked with a research project owned by researchers at IIT-CNR, in Pisa-Italy. The account biography points to the project web page[1]. At that page, one may find instructions to join the initiative and an overall description of motivations and goals of the project. In a first phase, the owners contacted further researchers and journalists to advertise the initiative. The online version of a popular Italian newspaper and a famous Italian social media analyst promoted the project and invited people to join it (see [11], [15] for an Italian version of these pieces). Foreign journalists and bloggers also supported the initiative in their countries. In a twelve days period (Dec 12-24, 2012), the account has been followed by 574 followers. Through Twitter API v1.1, we crawled a series of public information from these followers, i.e., their profiles and timeline information, together with their followers and followings profiles. For this dataset, we crawled these 574 accounts, leading to the collection of 616,193 tweets and 971,649 relationships (namely, linked Twitter accounts).

All those followers voluntarily joined the project. To include them in our reference set of humans, we also launched a verification phase. Each follower received a direct message on Twitter from @TheFakeProject, containing an URL to a CAPTCHA, unique for each follower. We consider as "certified human" all the 469 accounts out of the 574 followers that successfully completed the CAPTCHA.

1. http://wafi.iit.cnr.it/TheFakeProject/

### 3.2 #elezioni2013 dataset

The #elezioni2013 dataset was born to support a research initiative for a sociological study carried out in collaboration with the University of Perugia and the Sapienza University of Rome, on the strategic changes in the Italian political panorama for the 3-year period 2013-2015. Researchers identified 84,033 unique Twitter accounts that used the hashtag #elezioni2013 in their tweets, during the period between January 9 and February 28, 2013. Identification of these accounts has been based on specific keyword-driven queries on the username and biography fields of the accounts' profiles. Keywords include blogger, journalist, social media strategist, congressperson, representative. Specific names of political parties have been also searched. In conclusion, all the accounts belonging to politicians and candidates, parties, journalists, bloggers, specific associations and groups, and whoever somehow was officially involved in politics, have been discarded. Accounts not having a biography have been discarded too. The remaining accounts (about 40k) have been classified as *citizens*. This last set has been sampled (with confidence level 95% and confidence interval 2.5), leading to a final set of 1488 accounts, that have been subject to a manual verification to determine the nature of their profiles and tweets. Finally, 1481 accounts became part of dataset #elezioni2013.

### 3.3 Gold standard of human accounts

The above introduced datasets form our final set of about 1950 verified human accounts. It is worth noticing how the two subsets differ from each other. The Fake Project consists of accounts that have been recruited on a volunteer base: people involved in the initiative aimed to be part of an academic study for discovering fake followers on Twitter, and are a mixture of researchers and social media experts and journalists, mostly from Italy, but also from US and other European countries. The #elezioni2013 set consists of particularly active Italian Twitter users, with different professional background and belonging to diverse social classes, sharing a common interest for politics, but that do not belong to the following categories: politicians, parties, journalists, bloggers.

### 3.4 Gold standard of fake followers

In April, 2013, we bought 3000 fake accounts from three different Twitter online markets. In particular, we bought 1000 fakes accounts from http://fastfollowerz.com, 1000 from http://intertwitter.com, and 1000 fake accounts from http://twittertechnology.com, at a price of $19, $14 and $13 respectively.

## 4 Algorithms based on classification rules

In this section, we detail three procedures explicitly conceived for fake and bot account detection, namely [3], [17], [18]. Coincidentally, all of them are proposed as algorithms relying on a list of rules, or criteria: each account to be classified is checked against all the rules and the output of the checks are combined together in order to make the final classification. For each of the procedures, we report the criteria as indicated by the original sources, and we further specify how we implemented them into a rule suitable to be applied over our datasets. We also detail the reason for our implementation choices.

In this section, we mainly focus on the application of each single rule over our dataset, to assess its strength to discriminate fake followers. Indeed, in many cases, the analyzed algorithm did not specify the final combination of the proposed rule outcomes. Details on how aggregation has been performed are provided in [3] only. Driven by the provided details, we implement the full algorithm and we present its detection performances in Section 4.5.

Throughout the sequel of the paper we use the term "friends" to denote the users followed by an account (i.e., if $A$ follows $B$, $B$ is a friend of $A$).

### 4.1 Followers of political candidates

Camisani-Calzolari [3] carried out a series of tests over samples of Twitter followers of Romney and Obama, for the last US presidential election candidates, as well as for popular Italian politicians. In [3] it is detailed an algorithm to evaluate the account nature based on some of its public features. The cited algorithm has enough details to be reproducible: it assigns *human/active* and *bot/inactive* scores and classifies an account considering the gap between the sum of two scores. In particular, the algorithm assigns to the examined accounts 1 (or more, where specified) *human* point for each of the following criteria:

1) the profile contains a name;
2) the profile contains an image;
3) the profile contains a physical address;
4) the profile contains a biography;
5) the account has at least 30 followers;
6) it has been inserted in a list by other Twitter users;
7) it has written at least 50 tweets;
8) the account has been geo-localized;
9) the profile contains a URL;
10) it has been included in another user's favorites;
11) it writes tweets that have punctuation;
12) it has used a hashtag in at least one tweet;
13) it has logged into Twitter using an iPhone;
14) it has logged into Twitter using an Android device;
15) it is connected with Foursquare;
16) it is connected with Instagram;
17) it has logged into *twitter.com* website;
18) it has written the userID of another user in at least one tweet, that is it posted a *@reply* or a *mention*;
19) (2*number followers) $\geq$ (number of friends);
20) it publishes content which does not just contain URLs;

21) at least one of its tweets has been *retwitted* by other accounts (it's worth 2 points);
22) it has logged into Twitter through different clients (it's worth 3 points).

Moreover, the account receives 2 *bot* points if it only uses APIs. Finally, for each criterion that fails to be verified, the account receives 1 *bot* point, with the exception of criteria 13, 14, 15, 16, 17 and 8: in this cases, no *bot* points are assigned. To verify those rules, we referred to the *source* metadata of the tweets, that contains a different value representing from which platform the user posted a tweet. In particular, concerning the above rules, we considered the *source* metadata with the values *iphone, android, foursquare, instagram* and *web*, respectively, and we assigned 1 *human* point for each of the values found at least once within the collected tweets of the account. For the criterion 21, 2 *bot* points are assigned if no tweets of the account have been retweeted by other users.

Considering rule 22, *geo-localization* is related to tweets. Consequently, we set this rule as satisfied when at least one tweet of the account has been geo-localized.

For the rule 11, *punctuation* has been searched in both the profile biography and in its timeline.

### 4.2 Stateofsearch.com

Among the several bloggers that propose their golden rules to identify suspicious Twitter accounts, we consider the *"7 signals to look out for recognizing Twitter bots"*, according to the founder of the social media website *stateofsearch.com* [18]. The "7 signals to look out for" to recognize Twitter bots are the following [18]:

1) the biography of the profile clearly specifies that it is a bot account;
2) the friends/followers ratio is in the order of 100:1;
3) the account tweets the same sentence to many other accounts;
4) different accounts with duplicate profile pictures are suspicious;
5) accounts that tweet from API are suspicious;
6) the response time (follow+reply) to tweets of other accounts is within milliseconds;
7) the account tends to follow/unfollow other accounts within a temporal arc of 24 hours.

The rule 3 has been implemented considering the tweet as a single unit . We consider the last 20 tweets of each timeline.

For the rule 4, we consider the existence of a duplicate profile picture when at least 3 accounts within the dataset have the same profile picture.

For the rule 5, we consider as tweets posted from API all those tweets not being posted from the website *twitter.com*.

We did not apply rules 6 and 7 to our datasets, since they require to actively interact with the account. This means that those rules cannot be used to support an automatic detection process.

### 4.3 Socialbakers Fake Follower Check

Several companies provide online tools to classify Twitter followers based on their *fakeness* degree. Here, we consider the FakeFollowerCheck tool, by Socialbakers [17]. While the company website provides eight criteria to evaluate the fakeness degree of the followers of a certain account, it omits details on how to combine such criteria to classify the account. We contacted their customer service, but we were answered that "how the respective criteria are measured is rather an internal information". The FakeFollowerCheck tool considers the followers of an account and consider them likely fake when the following criteria are satisfied:

1) the ratio $\frac{friends}{followers}$ of the account under investigation is 50:1, or more;
2) more than 30% of all the tweets of the account use spam phrases, such as "diet", "make money" and "work from home";
3) the same tweets are repeated more than three times, even when posted to different accounts;
4) more than 90% of the account tweets are retweets;
5) more than 90% of the account tweets are links;
6) the account has never tweeted;
7) the account is more than two months old and still has a default profile image;
8) the user did not fill in neither bio nor location and, at the same time, she is following more than 100 accounts.

For the rule 2, we consider as spam phrases expressions like "diet" or "make money" or "work from home" (both English and Italian translations), as suggested by the website of Socialbakers.

It is worth noticing that the website reports the Fake Follower Check as a beta version, adding the following: "We are currently tweaking the algorithm". Therefore, we consider the criteria published on the firm website in December 2013.

### 4.4 Evaluation methodology

All the criteria above detailed have been applied to our mixed dataset, composed of a priori known human accounts, belonging to The Fake Project (469 verified accounts) and to #elezioni2013 (1481 verified accounts), as well as the fake accounts bought from the Twitter account markets, as described in Section 3. To obtain a mixed and balanced dataset of accounts composed by 50% of humans and 50% of fakes, we randomly chose 1950 out of the 3000 fake accounts bought. This set of 3900 accounts (a subset of our gold standard) has been used as the reference dataset for all our experiments (where not otherwise specified).

We conducted one experiment for each rule, considering two classes of accounts, the fakes and the humans. To summarize the outcomes of each experiment, we introduce four standard indicators, namely:

- *True Positive (TP)*: the number of those fake followers recognized by the rule as fakes;

| dataset | real humans | outcome | | |
|---|---|---|---|---|
| | | humans | bots | neutral |
| @TheFakeProject | 469 | 456 | 3 | 10 |
| #elezioni2013 | 1481 | 1480 | 0 | 1 |
| 100% fake | 0 | 2889 | 185 | 277 |

TABLE 1
Camisani-Calzolari algorithm outcomes on gold standard

- *True Negative (TN)*: the number of those human followers recognized by the rule as humans;
- *False Positive (FP)*: the number of those human followers recognized by the rule as fakes;
- *False Negative (FN)*: the number of those fake followers recognized by the rule as humans.

The meaning of each indicator is graphically highlighted by the following matrix (called the *confusion matrix* [10]), where each column represents the instances in the predicted class, while each row represents the instances in the actual class:

| actual class | predicted class: *human* | predicted class: *fake* |
|---|---|---|
| *human* | *TN* | *FP* |
| *fake* | *FN* | *TP* |

In order to evaluate the application of each single rule to the accounts in the gold standard, we consider the following, standard, evaluation metrics:

- *Precision*: the proportion of predicted positive cases that are indeed real positive, that is $\frac{TP}{TP+FP}$;
- *Recall*: the proportion of real positive cases that are indeed predicted positive, that is $\frac{TP}{TP+FN}$;
- *F-Measure*: the harmonic mean of precision and recall, namely $\frac{2 \cdot precision \cdot recall}{precision+recall}$
- *Matthew Correlation Coefficient* (*MCC* from now on) [1]: the estimator of the correlation between the predicted class and the real class of the samples. This metric is considered the unbiased version of the F-Measure, since it uses all four elements of the confusion matrix:

$$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}}$$

A *MCC* value close to 1 means that the prediction is really accurate, a value close to 0 means that the prediction is no better than random and a value close to -1 means that the prediction is heavily in disagreement with the real class. Then, we consider as best rules those criteria whose application gives $MCC \geq 0.6$, since such rules have the strongest correlation with the typology of the accounts.

### 4.5 Evaluation of Camisani-Calzolari algorithm

The detection algorithm in [3] aggregates the twenty-two criteria for identifying human and bot behavior, above introduced in Section 4.1. The algorithm evaluates every single rule on the account under investigation, and it assigns a positive human score or a negative bot score, according to the output of the rule application. The final outcome depends on the global score obtained by the account: if the result is a score greater than 0, then the account is marked as *human*; if it is between 0 and -4, it is marked as *neutral*; otherwise, it is marked as *bot*.

Table 1 reports in detail the results of running the algorithm over the complete reference dataset, including all the bought fake accounts. Although obtaining very good results in detecting the real human accounts, the algorithm achieves a poor fake account detection. Most of the accounts have been erroneously tagged as humans too. The main motivation of this unsatisfactory result is that the fake followers in our dataset feature characteristics that easily make them obtaining a human score higher than the bot one.

### 4.6 Single rule evaluation

Table 2 summarizes the results obtained by the application to our dataset of each single rule in sections 4.1, 4.2, and 4.3. In the table, we have highlighted the rules whose application on our dataset results in higher *MCC* values. Visibly, only three rules obtained a value higher than 0.6, namely:

1) the threshold of at least 30 followers;
2) the threshold of at least 50 tweets;
3) the use of a userID in at least one tweet.

Noticeably, none of the criteria suggested by online blogs and those addressed by Socialbakers FakeFollowerCheck are successful in detecting the fakes in our dataset. Clearly, the rules proposed by State of Search are aimed at detecting bots and, actually, we did not expect they would have performed brilliantly on detecting fakes. However, we observe that the rule "tweet from API" has an *MCC* of -0.779, meaning that it is strictly related to the class of the account, but by an inverse factor: in our dataset, fake accounts almost never tweet from API (instead, they use Twitter.com to tweet), whereas human accounts have posted at least once from outside the website. This is exactly the opposite behavior than that suggested by the blogger for bots, that almost exclusively post tweets using API.

Another interesting observation is that many rules proposed by Socialbakers have *MCC* values close to 0, meaning that their outcomes are almost unrelated with the class of the accounts. Indeed, the large majority of the accounts are recognized as humans (remember that *TN* is the number of human accounts recognized as humans and *FN* is the number of fakes recognized as humans). In other words, independently from the typology of the account, the rules are almost always satisfied, making it severely flawed for fake detection purposes. Such an independence from the account type is also exhibited by many other rules of both [3] and [18], like, for example, "*bot* in biography", "profile has a name", or "profile has an image", as shown in Table 2.

| | | results | | | | evaluation metrics | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **rule description** | *TP* | *TN* | *FP* | *FN* | *precision* | *recall* | *F-M.* | *MCC* |
| *Camisani-Calzolari* [3] (satisfaction of rules means human behavior) | | | | | | | | | |
| 1 | profile has name | 0 | 1950 | 0 | 1950 | — | — | — | — |
| 2 | profile has image | 2 | 1931 | 19 | 1948 | 0.095 | 0.001 | 0.002 | -0.06 |
| 3 | profile has address | 323 | 1313 | 637 | 1627 | 0.336 | 0.166 | 0.222 | -0.187 |
| 4 | profile has biography | 617 | 1806 | 144 | 1333 | 0.811 | 0.316 | 0.455 | 0.306 |
| 5 | followers $\geq$ 30 | 1852 | 1582 | 368 | 98 | 0.834 | 0.95 | 0.888 | 0.768 |
| 6 | belongs to a list | 1893 | 1052 | 898 | 57 | 0.678 | 0.971 | 0.799 | 0.566 |
| 7 | tweets $\geq$ 50 | 1582 | 1792 | 158 | 368 | 0.909 | 0.811 | 0.857 | 0.735 |
| 8 | geo-localization | 1923 | 678 | 1272 | 27 | 0.602 | 0.986 | 0.748 | 0.434 |
| 9 | has URL in profile | 1895 | 697 | 1253 | 55 | 0.602 | 0.972 | 0.743 | 0.417 |
| 10 | in favorites | 1130 | 1748 | 202 | 820 | 0.848 | 0.579 | 0.689 | 0.502 |
| 11 | uses punctuation in tweets | 93 | 1948 | 2 | 1857 | 0.979 | 0.048 | 0.091 | 0.151 |
| 12 | uses hashtags | 437 | 1934 | 16 | 1513 | 0.965 | 0.224 | 0.364 | 0.337 |
| 13 | uses iPhone to log in | 1905 | 845 | 1105 | 45 | 0.633 | 0.977 | 0.768 | 0.489 |
| 14 | uses Android to log in | 1932 | 677 | 1273 | 18 | 0.603 | 0.991 | 0.75 | 0.442 |
| 15 | has connected with Foursquare | 1943 | 261 | 1689 | 7 | 0.535 | 0.996 | 0.696 | 0.257 |
| 16 | has connected with Instagram | 1890 | 772 | 1178 | 60 | 0.616 | 0.969 | 0.753 | 0.446 |
| 17 | uses the website Twitter.com | 131 | 1852 | 98 | 1819 | 0.572 | 0.067 | 0.12 | 0.036 |
| 18 | has tweeted a userID | 1071 | 1941 | 9 | 879 | 0.992 | 0.549 | 0.707 | 0.609 |
| 19 | 2*followers $\geq$ friends | 1947 | 863 | 1087 | 3 | 0.642 | 0.998 | 0.781 | 0.531 |
| 20 | tweets do not just contain URLs | 125 | 1943 | 7 | 1825 | 0.947 | 0.064 | 0.12 | 0.167 |
| 21 | retwitted tweets $\geq$ 1 | 1021 | 1915 | 35 | 929 | 0.967 | 0.524 | 0.679 | 0.569 |
| 22 | uses different clients to log in | 118 | 1924 | 26 | 1832 | 0.819 | 0.061 | 0.113 | 0.125 |
| *Van Den Beld (State of search)* [18] (satisfaction of rules means bot behavior) | | | | | | | | | |
| 1 | *bot* in biography | 0 | 1950 | 0 | 1950 | — | — | — | — |
| 2 | following:followers = 100:1 | 158 | 1950 | 0 | 1792 | 0.541 | 1 | 0.15 | 0.205 |
| 3 | same sentence to many accounts | 188 | 1521 | 429 | 1762 | 0.438 | 0.78 | 0.146 | -0.169 |
| 4 | duplicate profile pictures | 26 | 1809 | 141 | 1924 | 0.471 | 0.928 | 0.025 | -0.146 |
| 5 | tweet from API | 429 | 33 | 1917 | 1521 | 0.118 | 0.017 | 0.2 | -0.779 |
| *Socialbakers* [17] (satisfaction of rules means fake behavior) | | | | | | | | | |
| 1 | friends:followers $\geq$ 50:1 | 316 | 1949 | 1 | 1634 | 0.997 | 0.162 | 0.279 | 0.296 |
| 2 | tweets spam phrases | 5 | 1950 | 0 | 1945 | 1 | 0.003 | 0.005 | 0.036 |
| 3 | same tweet $\geq$ 3 | 30 | 1327 | 623 | 1920 | 0.046 | 0.015 | 0.023 | -0.407 |
| 4 | retweets $\geq$ 90% | 14 | 1933 | 17 | 1936 | 0.452 | 0.007 | 0.014 | -0.009 |
| 5 | tweet-links$\geq$ 90% | 58 | 1936 | 14 | 1892 | 0.806 | 0.03 | 0.057 | 0.084 |
| 6 | 0 tweets | 84 | 1949 | 1 | 1866 | 0.988 | 0.043 | 0.083 | 0.146 |
| 7 | default image after 2 months | 2 | 1931 | 19 | 1948 | 0.095 | 0.001 | 0.002 | -0.06 |
| 8 | no bio, no location, friends $\geq$100 | 255 | 1927 | 23 | 1695 | 0.917 | 0.131 | 0.229 | 0.231 |

TABLE 2
Rules evaluation

We acknowledge that our fake followers dataset could be illustrative, and not exhaustive, of all the possible existing sets of fakes. However, it is worth noticing that we found the Twitter accounts marketplaces by simply Web searching them on the most common search engines. Thus, we can argue that our dataset represents what is easily possible to find on the Web.

## 5 Evaluation of algorithms based on feature sets

In this section, we examine work in [22], [26] that address spam account detection on Twitter. Both of them propose a list of features to be extracted from manually classified datasets of accounts. Such features sets are then used to train and test machine learning classifiers that learn to distinguish among humans and spammers. Even if the proposed features have been originally designed for spam detection, here, for the first time, we consider them to spot another category of Twitter accounts, i.e., the fake followers.

### 5.1 Detecting spammers in social networks

The work presented in [22] focuses on detecting spambots exploiting five features, that can be gathered crawling an account's details, both from its profile and timeline. The features are:

1) the number of friends;
2) the number of tweets;
3) the content of tweets;
4) the URL ratio in tweets;
5) the relation between the number of friends and followers.

For each investigated account, such features are given as input to a Random Forest algorithm [7], [8], that outputs if the account is a spambot or not. In particular, results of the analysis in [22] show that, on average, the accounts under investigation show a spambot behavior if:

1) they do not have thousands of friends;
2) they have sent less than 20 tweets;
3) the content of their tweets exhibits the so called *message similarity*;

4) they a high $\frac{tweets\ containing\ URLs}{total\ tweets}$ ratio (*URL ratio*);
5) they have a high $\frac{friends}{(followers\hat{}2)}$ ratio value (i.e., lower ratio values mean legitimate users).

We briefly give some notes on how we use the features of [22] over our dataset. For feature 3, we implement the notion of message similarity by checking the existence of at least two tweets, in the last 15 tweets of the account timeline, in which 4 consecutive words are equal (words are consecutive characters separated by white spaces). This notion has been given in a latter work by the same authors, see [21].

Without the original training set, we were unable to reproduce the same classifier, but we picked the five features and used them to train a set of classifiers with our dataset. The results are reported in Table 3 of Section 5.3.

### 5.2 Fighting evolving Twitter spammers

The authors of [25], [26] observed that Twitter spammers often modify their behavior in order to evade existing spam detection techniques. Thus, they suggested to consider some new features, making evasion more difficult for spammers. Beyond the features directly available from the account profile lookup, the authors propose some graph-, automation-, and timing-based features. We detail nine of them:

1) *age* of the account (this feature also appears in [13]);
2) *bidirectional link ratio*, i.e., $\frac{bidirectional\ links}{friends}$, where a bidirectional link is when two accounts follow each other;
3) *average neighbors' followers*, i.e., the average number of followers of the account's friends. This feature aims at reflecting the quality of the choice of friends of an account;
4) *average neighbors' tweets*: the average number of tweets of the account's followers;
5) *followings to median neighbor's followers* of an account, defined as the ratio between the number of friends and the median of the followers of its friends;
6) *API ratio* (= number of tweets sent from API / total number of tweets);
7) *API URL ratio* (= number of tweets posted from API and containing URL / total number of tweets posted from API);
8) *API tweet similarity*: this metric considers only the number of similar tweets sent from API. The notion of tweet similarity is as in Section 5.1;
9) *following rate*: this metric reflects the speed at which an accounts follows other accounts.

The authors of [25], [26] combine their features in four different machine learning classifiers and compare their implementation with other existing approaches. Their results on the nine features were as follows:

1) *age*: the more an account is aged, the more it could be considered a good one;
2) *bidirectional link ratio* has been tested to be lower for spammer accounts than for legitimate accounts;
3) *average neighbors' followers* is commonly higher for legitimate accounts than for spammers;
4) *average neighbors' tweets* should be lower for spammers than for legitimate accounts;
5) *followings to median neighbor's followers* has been found higher for spammers than for legitimate accounts;
6) *API ratio*: work in [4], [26] reveal higher values for suspicious accounts;
7) *API URL ratio* is higher for suspicious accounts;
8) *API tweet similarity*: the idea is that a higher API tweet similarity of an account implies that this account is suspicious;
9) *following rate*: the idea is that higher values are related to spammers.

We were unable to completely reproduce the machine learning classifiers in [26], since we had a different dataset, but here we evaluate how those features, proved to be quite robust against evasion techniques adopted by spammers, perform in detecting fake followers.

Some notes follow on our implementation of the features above. Precisely evaluating rule 9 requires to know the evolution of the number of friends of an account. Actually, this kind of information is publicly unavailable and, as in [26], we approximate the rate as the ratio *friends/age*.

Interestingly, the authors of [26] also suggest two further graph-based features. The first feature is the *local clustering coefficient* and it quantifies how close the neighbors of a Twitter account are to be a clique. The intuitive idea behind this feature is that spammers blindly follow other accounts, that do not know each other and have a looser relationship among them, thus, they do not form a clique. Therefore, spammers have lower local clustering coefficients, compared to humans. The second feature is the *betweenness centrality* which reflects the position of a node in the graph (namely how much a node is involved in the shortest paths between all the possible pairs of vertices). The intuitive idea behind this feature is that, following unrelated accounts, the spammer will create new shortest paths between those who re-follow it, leading to a position in the graph more central for the spammer than for human accounts.

Although these features should be very effective to recognize spammers, unfortunately they are extremely computational expensive to evaluate and the same authors evaluated it using a simplified approach. This is the reason why we have not implemented them in our analysis, and, thus, how these features behave in discriminating fake followers is an open issue.

Finally, note that in [26] there are also other features, in addition to the above-mentioned; however, as claimed by the same authors, they are less robust against evasion techniques. For this reason, we decided not to include them in our evaluation.

| | classifier | results TP | TN | FP | FN | evaluation metrics precision | recall | F-M. | MCC |
|---|---|---|---|---|---|---|---|---|---|
| *Classifiers based on feature set of Yang et al. [26]* | | | | | | | | | |
| RF | Random Forest | 1933 | 17 | 17 | 1933 | 0.991 | 0.991 | 0.991 | 0.983 |
| D | Decorate | 1927 | 23 | 14 | 1936 | 0.988 | 0.993 | 0.991 | 0.983 |
| J48 | Decision Tree | 1933 | 17 | 21 | 1929 | 0.991 | 0.989 | 0.990 | 0.980 |
| AB | Adaptive Boosting | 1928 | 22 | 26 | 1924 | 0.989 | 0.987 | 0.988 | 0.974 |
| BN | Bayesian Network | 1939 | 11 | 81 | 1869 | 0.994 | 0.958 | 0.976 | 0.936 |
| *Classifiers based on feature set of Stringhini et al. [22]* | | | | | | | | | |
| RF | Random Forest | 1917 | 33 | 40 | 1910 | 0.983 | 0.979 | 0.981 | 0.961 |
| D | Decorate | 1919 | 31 | 41 | 1909 | 0.984 | 0.979 | 0.981 | 0.961 |
| J48 | Decision Tree | 1919 | 31 | 51 | 1899 | 0.984 | 0.974 | 0.979 | 0.953 |
| AB | Adaptive Boosting | 1882 | 68 | 58 | 1892 | 0.965 | 0.970 | 0.968 | 0.938 |
| BN | Bayesian Network | 1859 | 91 | 91 | 1859 | 0.953 | 0.953 | 0.953 | 0.907 |
| *Classifier based on Camisani-Calzolari algorithm [3]* | | | | | | | | | |
| CC | Camisani-Calzolari♯ | 1936 | 3 | 1687 | 111 | 0.534 | 0.998 | 0.696 | 0.178 |

TABLE 3
Performance comparison for 10-fold cross validation. Training set: 1950 humans and 1950 fake. (♯): the CC algorithm classified 163 accounts as *neutral*.

### 5.3 Evaluation

To evaluate the feature sets described in Sections 5.1 and 5.2, we used five classifiers obtained exploiting five different machine learning based algorithms, namely Decorate (D), Adaptive Boost (AB), Random Forest (RF), Decision Tree (J48) and Bayesian Network (BN), all implemented within the Weka framework [7]. Random Forest was the only used by the authors of [22] and all of them, but Adaptive Boost, were used by the authors of [26] to build spam detection classifiers. We also included AB since it is considered one of the more effective machine learning algorithm for classification tasks. For both the considered works, we built five classifiers adopting the suggested features, and training the models using our reference dataset. Then, we used a 10-fold cross validation [8] to estimate the performances of each obtained classifier. As for the rule-based algorithms in Section 4.4, we consider the *MCC* as the preferred metric to assess classifier performances. The obtained results are summarized in Table 3. The table also reports the results of the classification algorithm proposed in [3] and discussed in Section 4.5.

We observe that all the classifiers built with the feature set of [26] obtain better results, compared to the others. In particular, RF, J48 and D classifiers have a *MCC* above 0.98. Similarly, precision and recall are around 0.99 for all of them. The classifiers built with the feature set of Stringhini *et al.* [22], also, obtain extremely high detection levels: precision and recall are around 0.98 for RF and D, with an *MCC* of 0.96. Overall, even if some small differences can be observed in the number of false negatives and false positives, all the classifiers almost correctly distinguish between human and fake follower accounts, in our reference dataset. The feature-based classifiers are indisputably more accurate for fake detection when compared with the CC algorithm, that does not perform well within our dataset, as observed above in Section 4.5.

### 5.4 Discussion

By examining the internal structure of the classifiers, we get insights about the best features that contribute more to distinguish humans and fakes. In case of decision trees, the best features are the ones closer to the root and the classifier automatically finds the numeric thresholds characterizing, for a given feature, the borderline between humans and fakes. It is worth noticing also that the Decorate, AdaBoost, and Random Forest algorithms exploit, ultimately, combinations of simple decision tree classifiers. Despite their very good performance, they have the disadvantage of being difficult to analyze, since they can consist in tens of individual trees that interact together. Then, we only focus on the J48 classifier (a single decision tree) to examine how the features are applied during the classification process.

#### 5.4.1 Differences between fake followers and spam accounts

Looking at the tree structure, we observe some interesting differences between the fake followers in our dataset and the spam accounts characterized in [22] and [26]. For example, the feature *URL ratio* has been found to have a higher value for spammers than for legitimate users, as highlighted in [22] (Section 5.1). Observing the tree structure of our J48 classifier, instead, low values for this feature characterize fake followers, compared with higher values that indicate human accounts in our gold standard. More than 72% of the fake followers in our training dataset have a *URL ratio* lower than 0.05, oppositely to 14% of human accounts. Similarly, the *API ratio* feature has been found higher for spammers than for legitimate accounts ( [26], see also Section 5.2). In our dataset, the *API ratio* is lower than 0.0001 for 78% of fake followers. A similar behavior has been observed for the

| | tree details | | | evaluation metrics | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **pruning method** | *nodes* | *leaves* | *height* | *TP rate* | *FP rate* | *precision* | *recall* | *F-M.* | *MCC* |
| *Decision tree based on feature set of Stringhini et al. [22]* | | | | | | | | | |
| subtree raising 0.25 | 43 | 22 | 7 | 0.979 | 0.021 | 0.979 | 0.979 | 0.979 | 0.953 |
| reduced error 3 folds | 31 | 16 | 5 | 0.975 | 0.025 | 0.975 | 0.975 | 0.975 | 0.943 |
| reduced error 50 folds | 9 | 5 | 4 | 0.964 | 0.036 | 0.964 | 0.964 | 0.964 | 0.914 |
| *Decision tree based on feature set of Yang et al. [26]* | | | | | | | | | |
| subtree raising 0.25 | 33 | 17 | 8 | 0.99 | 0.01 | 0.991 | 0.989 | 0.99 | 0.980 |
| reduced error 3 folds | 19 | 10 | 5 | 0.988 | 0.012 | 0.988 | 0.988 | 0.988 | 0.976 |
| reduced error 50 folds | 11 | 6 | 3 | 0.982 | 0.018 | 0.982 | 0.982 | 0.982 | 0.966 |
| *Decision tree based on feature set of Yang et al. [26], without the bi-link ratio feature* | | | | | | | | | |
| subtree raising 0.25 | 101 | 51 | 10 | 0.96 | 0.04 | 0.96 | 0.96 | 0.96 | 0.917 |
| reduced error 3 folds | 53 | 27 | 8 | 0.961 | 0.039 | 0.961 | 0.961 | 0.961 | 0.914 |
| reduced error 50 folds | 37 | 19 | 9 | 0.933 | 0.067 | 0.933 | 0.933 | 0.933 | 0.866 |

TABLE 4
Performance comparison with increased pruning. 10-fold cross validation. Training set: 1950 humans and 1950 fakes.

*average neighbor's tweets* feature, that has been found to be lower for spammers in [26], but higher for our fakes.

These initial observations highlight a behavioral difference between a spam account and a fake follower. In particular, fake followers appear to be more passive compared to spammers and they do not make use of automatized mechanisms for posting their tweets, as spammers usually do.

#### 5.4.2 Reducing overfitting

It is well known that trained classifiers can be subject to "overfitting", namely the problem of being too specialized on the training dataset and unable to generalize the classification to new and unseen data [9]. In other words, the classifier could have worse predictive ability since its internal structure and reasoning are more complicated than required.

A simple way to avoid overfitting is to keep the classifier as simple as possible. In case of a decision tree algorithm, for example, one solution could be reducing the number of nodes and, possibly, the height of the tree. The decision tree obtained with the feature set of [22] has 22 leaves, 43 nodes, and a height of 7, whereas the best feature is the *friends/(followers^2)* ratio that places at the root. The decision tree with the feature set of [26] has 17 leaves, 33 nodes and a height of 8, with the *bi-directional link* ratio as the root.

A common practice to generalize the classifiers is the adoption of a more aggressive pruning strategy, e.g., by using the reduce-error pruning with small test sets [7], [8]. Adopting this strategy, we were able to obtain simpler trees with a lower number of nodes and a very reduced height. Such simpler trees only use subsets of the feature set, still maintaining very good performance on our dataset.

Table 4 reports the characteristics and the performance of the experiments we have carried out, varying the pruning strategy. It is worth noticing that the complexity of the tree is not directly responsible of the improvement in the detection capability: for example, for the feature set of [26], reducing the number of nodes from 33 to 11 decreases the *TP rate* of 0.018 and the *MCC* of 0.014, only. The results of this experiment show that, even reducing the features, it is possible to have a detection rate higher than 0.95 (as in the last line of Table 4, for [22] and [26], respectively. For example, in those two experiments, the features used by the pruned tree were only *bi-directional link ratio*, the *average neighbors' followers*, the *age*, and the *followings to median neighbors' followers* as a subset of the original feature set of [26], and the *friends/(followers^2)*, *URL ratio*, and *number of friends* as the subset for [22]'s original feature set.

#### 5.4.3 Bidirectional link ratio

To test if the bidirectional link ratio is the decisive feature to distinguish between humans and fake followers in our reference dataset and how much it influences the detection process, we compare the results of the previous experiments with a new one: we build a decision tree classifier leaving out the bi-link ratio from the feature set of [26].

This experiment is particularly interesting since, as detailed in next Section 6, this feature is the more expensive to evaluate, especially in terms of crawling. The results in Table 4 show a limited lowering of both *TP rate* and *FP rate* for the less pruned trees (subtree raising 0.25 and reduced error 3 folds), but a more evident reduction of the *MCC* measure. The reduced error pruning with 50 folds produces a classifier that has *MCC* dropping from 0.966 to 0.866. However, the detection level (*TP rate*) is still very good for all the three pruned trees (0.96, 0.961 and 0.933, respectively). The more interesting aspect is the increased complexity of the decision tree: without the bi-link ratio, the classifiers need to resort to a considerably larger number of nodes. For example, the tree that does not use that feature, pruned with subtree raising confidence of 0.25, requires 101 nodes, whereas the tree that uses it requires only 33.

From the results shown in Table 4, we conclude that the bidirectional link ratio is an important feature for

fake follower detection: even if not essential, it is extremely effective to the detection process.

## 6 Optimized classifier

As previously shown in Sections 4 and 5, the classifiers based on feature sets perform much better than those based on rules. Similarly, we have seen that the feature set proposed by Yang *et al.* seems to be slightly more effective than that proposed by Stringhini, when used in feature-based classifiers aiming at fake followers detection. Here, we look for an optimized classifier, exploiting the best features and the best rules, not only in terms of detection performance, but also considering their evaluation costs. In particular, we can distinguish between the computational cost and the crawling cost required to evaluate a feature (or a rule). Computational costs can be generally lowered with optimized algorithms and data representations and they are negligible when compared to the crawling costs. Thus, in this section we focus on the latter: we quantify the crawling cost of each feature and rule, and we build a set of optimized classifiers that make use of the more efficient features and rules, in terms of crawling cost and fake followers detection capability. For the sake of readability, in the following section with the term "feature" we intend all the rules and features presented in Sections 4 and 5.

### 6.1 Crawling cost analysis

Intuitively, some features require few data for their calculation, while others require the download of big amounts of data. For the sake of this analysis, we divide the features in three categories:

*A*) *profile*: features that require information present in the profile of the followers of the target account (like, e.g., *profile has name*);
*B*) *timeline*: features that require the tweets posted in the timeline of the followers of the target account (like, e.g., *tweet from API*);
*C*) *relationship*: features that require information of the accounts that are in a relationship (i.e., that are a friend, or a follower, or both) with the followers of the target account (like, e.g., *bidirectional link ratio*).

Each category, in turn, belongs to a crawling cost class directly related to the amount of data to be crawled from Twitter. Starting from the list of the followers of a target account, *Class A* features can be evaluated simply accessing to all the profiles of the followers; *Class B* features require to download all the tweets posted by each follower; *Class C* features need to crawl the friends and the followers of each follower of the target account. To evaluate the class of cost associated to each feature's category, we estimate the number of *API calls* needed to download data required for the calculation. Results are in Tables 5 and 6. The following parameters refer to the Twitter account for which the number of fake followers is being investigated:

| | *profile* | *timeline* | *relationship* |
|---|---|---|---|
| API calls | $\left\lceil \frac{f}{100} \right\rceil$ | $\sum_{i \in f} \left\lceil \frac{t_i}{200} \right\rceil$ | $\sum_{i \in f} (\left\lceil \frac{f_i}{5000} \right\rceil + \lceil \frac{\varphi_i}{5000} \rceil)$ |
| Best-case | $\left\lceil \frac{f}{100} \right\rceil$ | $f$ | $2 * f$ |
| Worst-case | $\left\lceil \frac{f}{100} \right\rceil$ | $16 * f$ | unpredictable |
| Calls/min. | 12 | 12 | 1 |

TABLE 5
Number of API calls needed to download data

$f$ : number of followers of the target account;
$t_i$ : number of tweets of the *i*-th follower of the target account;
$\varphi_i$ : number of friends of the *i*-th follower of the target account;
$f_i$ : number of followers of the *i*-th follower of the target account.

The number of API calls for each category depends on the maximum number of accounts (100), tweets (200) and friends/followers (5000) that can be fetched from Twitter with a single API request. For example, for the *profile* category, a single API call can return 100 follower profiles, leading to $\left\lceil \frac{f}{100} \right\rceil$ API calls in total. The detailed costs do not account for the initial download of the whole list of $f$ followers of the target account, that requires $\left\lceil \frac{f}{5000} \right\rceil$ API calls.

Table 5 also shows the minimum (*Best-case*) and maximum (*Worst-case*) number of API calls that could possibly be required, that depend on the length of the timelines and the number of relationships of the followers. The Best-case is when a single API call is sufficient to get all the data for a single follower. For the Worst-case we can precisely evaluate the number of API calls for the *timeline* category, since the number of tweets that can be accessed from a user timeline is limited to 3200, leading to a maximum of 16 calls for each follower. The number of friends and followers, instead, is not limited and, therefore, it is impossible to calculate a worst-case scenario for the *relationship* category. However, we can consider the account with the maximum number of followers on Twitter, which, at the time of writing, belongs to the pop star Lady Gaga (*@ladygaga*), with about 40 millions of followers. We can therefore consider as the worst-case scenario an account with 40 millions followers and 40 millions friends, which leads to a number of API calls equal to $16000 * f$.

Observing the values of Table 5, we have a clear idea of the order of magnitude of each class: features in *Class B* are 100 times more costly than features of *Class A*, while features of *Class C* could be several orders of magnitude more costly than features of *Class A*.

To protect Twitter from abuse, the number of API calls allowed per minute is limited. In Table 5, we also report the maximum number of calls allowed per minute (*Calls/min.*), which directly impacts on the time needed to complete the data acquisition.

| *Feature set* | *Class A (profile)* | *Class B (timeline)* | *Class C (relationships)* |
|---|---|---|---|
| *Camisani-Calzolari* [3] | has name, has image, has address, has biography, followers≥30, belongs to a list, tweets≥50, URL in profile, 2*followers≥friends | geo-localized, is favorite, uses punctuation, uses hashtag, uses iPhone, uses Android, uses Foursquare, uses Instagram, uses *Twitter.com*, userID in tweet, tweets with URLs, retweet≥1, uses different clients | |
| *State of search* [18] | *bot* in biography, following:followers = 100:1, duplicate profile pictures | same sentence to many accounts, tweet from API | |
| *Socialbakers* [17] | friends:followers ≥ 50:1, default image after 2 months, no bio, no location, friends ≥100, 0 tweets | tweets spam phrases, same tweet≥3, retweets ≥ 90%, tweet-links≥ 90% | |
| *Stringhini* [22] | number of friends, number of tweets, $\frac{friends}{(followers^2)}$ | tweet similarity, URL ratio | |
| *Yang* [26] | age, following rate | API ratio, API URL ratio, API tweet similarity | bi-link ratio, average neighbors' followers, average neighbors' tweets, followings to median neighbor's followers |

TABLE 6
Feature crawling cost classes

| | | **results** | | | | **evaluation metrics** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **classifier** | *TP* | *TN* | *FP* | *FN* | *precision* | *recall* | *F-M.* | *MCC* |
| *Class C classifiers that use all the features* | | | | | | | | | |
| RF | Random Forest | 1931 | 1944 | 6 | 19 | 0.997 | 0.990 | 0.994 | 0.987 |
| J48 | Decision Tree | 1935 | 1932 | 18 | 15 | 0.991 | 0.992 | 0.992 | 0.983 |
| D | Decorate | 1931 | 1934 | 16 | 19 | 0.992 | 0.990 | 0.991 | 0.982 |
| AB | AdaBoost | 1924 | 1927 | 23 | 26 | 0.988 | 0.987 | 0.987 | 0.975 |
| BN | BayesNet | 1860 | 1883 | 47 | 90 | 0.975 | 0.954 | 0.964 | 0.931 |
| *Class A classifiers that use only Class A features* | | | | | | | | | |
| RF | Random Forest | 1911 | 1937 | 13 | 39 | 0.993 | 0.980 | 0.987 | 0.967 |
| D | Decorate | 1912 | 1924 | 26 | 38 | 0.987 | 0.981 | 0.984 | 0.964 |
| J48 | Decision Tree | 1909 | 1909 | 41 | 41 | 0.979 | 0.979 | 0.979 | 0.958 |
| AB | AdaBoost | 1889 | 1901 | 49 | 61 | 0.975 | 0.969 | 0.972 | 0.941 |
| BN | BayesNet | 1877 | 1889 | 61 | 73 | 0.969 | 0.963 | 0.966 | 0.928 |

TABLE 7
Performance comparison for 10-fold cross validation.

Some further considerations follow. Firstly, data collected for a category can be used to evaluate all the features of that category. Secondly, Twitter limits the number of calls of the same API, but different APIs can be called in parallel. This means that data for all the three feature categories can be possibly acquired contemporary. The total time required to collect all data depends on the category that requires more time, i.e., the *relationship* one. In other words, to get the total time, one must not consider the sum of the time needed for each of the three cost classes, but just the most costly.

### 6.2 The *Class A* classifier

All the rules and features considered in this study fall into one of the three aforementioned categories, as reported in Table 6. Therefore, their crawling cost impacts on the final cost of the whole feature set and, ultimately, to the class of the classifier: a classifier that uses a certain feature set belongs to the class of the more expensive feature. This means that all the classifiers we have analyzed in the previous sections are classifiers of *Class B*, excepting for the classifier with the feature set of Yang *et al.* that belongs to *Class C*.

In the following, we consider an optimized classifier working only with features of *Class A*. The aim is verifying whether the *Class A* classifier reaches performances that are comparable to those of the more expensive *Class B* and *Class C* classifiers.

Table 7 reports the results of the classifiers built with two different feature sets: all the features and the *Class A* features. We start observing that the classifier built with all the features considered in our study performs better than all the others, including the classifiers using the feature set of Yang *et al.* and Stringhini *et al.* reported in Table 3. However, the increase of *MCC* between the best classifier and our optimized *Class A* classifier is very limited, i.e., around 0.03 for RF, D, J48 and AB.

*Class A* classifier with BN only decreases its precision, but increases its recall, obtaining a *MCC* very close to the best classifier. Concerning the complexity of the two decision trees obtained with the J48 algorithm, we also observe that they are comparable, since the best classifier is composed by 31 nodes, while our *Class A* classifier by 41 nodes, and they have a height of 6 and 8 respectively. Another interesting observation is that both the classifiers select features that belong to different feature sets considered above, including some rules like *has tweeted a userID* and *has URL in profile*, proposed by Camisani-Calzolari [3].

## 7 Conclusions

The motivations for this paper steamed from the lack of rigorous definitions and criteria for the identification of fake followers on Twitter. Our main research objective was to assess the performance of the features used to recognize Twitter spam accounts —suggested in both the grey and the Academic literature— when applied to recognize fake accounts. To reach this goal, we firstly created a gold standard of human and fake Twitter accounts. Then, we collected and precisely analyzed various spam detection proposals based on classification rules and feature sets proposed in both the Academic and the grey literature, to target spambots and inactive accounts. From this extensive state of the art, we extracted several sets of rules and features that were eventually tested on our dataset to understand their effectiveness in detecting fake follower accounts. A few selected features performed pretty well and were very effective in spotting fake accounts, as shown using classifiers relying on optimized Machine Learning algorithms.

We further analyzed the best performing features, and ranked them according to their crawling cost. This led us to identify three categories of features incurring increasing crawling cost. Finally, we built an optimized classifier for fake follower account detection that only leverages lightweight features; we showed that our proposal is able to achieve detection rates comparable with the best of breed classifiers, whereas these latter ones necessitate overhead-demanding features.

## References


[1] P. Baldi, S. Brunak, Y. Chauvin, C. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.

[2] Brandwatch.com. Analysis of global brands' Twitter activity. In *http://goo.gl/C6MeU*, Dec. 2012. Last checked December 27, 2013.

[3] M. Camisani-Calzolari. Analysis of Twitter followers of the US Presidential Election candidates: Barack Obama and Mitt Romney. In *http://digitalevaluations.com/*, August 2012.

[4] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Trans. Dependable Sec. Comput.*, 9(6):811–824, 2012.

[5] Corriere Della Sera (online ed.). Academic Claims 54% of Grillo's Twitter Followers are Bogus. In *http://goo.gl/qi7Hq*, July 2012. Last checked December 27, 2013.

[6] Financial Times – Tech blog (online ed.). Twitter bots are boosting brands - survey. In *http://goo.gl/Zt2l2*, June 2012. Last checked December 27, 2013.

[7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11, 2009.

[8] M. Hall, I. Witten, and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3 edition, 2011.

[9] D. M. Hawkins. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12, 2004.

[10] R. Kohavi and F. Provost. Glossary of Terms. *Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process*, 30(2-3), 1998.

[11] La Repubblica (online ed.). Twitter, quanti falsi profili: il CNR ora va a caccia dei fake. In *http://goo.gl/zNC2k*, December 2012. Last checked December 27, 2013.

[12] Le Monde (online ed.), Big Browser blog. BANQUE POPULAIRE – Dis-moi combien d'amis tu as sur Facebook, je te dirai si ta banque va t'accorder un prêt. In *http://goo.gl/zN3PJX*, November 2013. Last checked December 27, 2013.

[13] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: social honeypots + machine learning. In *SIGIR Conference on Research and Development in Information Retrieval*, pages 435–442. ACM, 2010.

[14] New York Times (online ed.). Buying Their Way to Twitter Fame. In *http://goo.gl/VLrVK*, August 2012. Last checked December 27, 2013.

[15] Skande.com. Twitter: un progetto del CNR cerca i Fake Followers #ImNotAFake. In *http://goo.gl/V3zX6*, December 2012. Last checked December 27, 2013.

[16] C. Smith. By The Numbers: 68 Amazing Twitter Stats. In *http://goo.gl/2Xr9X*, Dec 2013. Last checked December 27, 2013.

[17] SocialBakers. Fake follower check. In *http://goo.gl/chWn0*. Last checked December 27, 2013.

[18] Stateofsearch.com. How to recognize Twitterbots: 7 signals to look out for. In *http://goo.gl/YZbVf*, September 2012. Last checked December 27, 2013.

[19] Statistic Brain. Twitter statistics. In *http://goo.gl/XEXB1*, Jan 2014. Last checked January 3, 2014.

[20] Statuspeople.com. Status People Fakers. In *http://goo.gl/0Jpky*. Last checked December 27, 2013.

[21] G. Stringhini, M. Egele, C. Kruegel, and G. Vigna. Poultry markets: on the underground economy of Twitter followers. In *Workshop on online social networks*, WOSN '12, pages 1–6. ACM, 2012.

[22] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *26th Annual Computer Security Applications Conference*, ACSAC '10, pages 1–9. ACM, 2010.

[23] The Guardian (online ed.). Barack Obama tweets the start to his 2012 re-election campaign. In *http://goo.gl/Uk6Av*, April 2011. Last checked December 27, 2013.

[24] The Telegraph (online ed.). Human or 'bot'? Doubts over Italian comic Beppe Grillo's Twitter followers. In *http://goo.gl/2yEgT*, July 2012. Last checked December 27, 2013.

[25] C. Yang, R. Harkreader, and G. Gu. Empirical evaluation and new design for fighting evolving twitter spammers. *Information Forensics and Security, IEEE Transactions on*, 8(8):1280–1293, 2013.

[26] C. Yang, R. C. Harkreader, and G. Gu. Die free or live hard? Empirical evaluation and new design for fighting evolving Twitter spammers. In *14th International Conference on Recent Advances in Intrusion Detection*, RAID'11, pages 318–337. Springer-Verlag, 2011.

[27] S. Yardi, D. M. Romero, G. Schoenebeck, and D. Boyd. Detecting Spam in a Twitter Network. *First Monday*, 15(1), 2010.