Fame for Sale: efficient detection of fake Twitter followers


IIT TR-05/2015

Technical Report

Maggio 2015
Fame for sale: efficient detection of fake Twitter followers

Stefano Cresci\textsuperscript{a}, Roberto Di Pietro\textsuperscript{a,b}, Marinella Petrocchi\textsuperscript{a}, Angelo Spognardi\textsuperscript{a,1,*}, Maurizio Tesconi\textsuperscript{a}

\textsuperscript{a}IIT-CNR, Via G. Moruzzi, 1 – 56124 Pisa, Italy
\textsuperscript{b}Bell Labs, Alcatel-Lucent, Paris, France

Abstract

Fake followers are those Twitter accounts specifically created to inflate the number of followers of a target account. Fake followers are dangerous for 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 baseline dataset of verified human and fake follower accounts. Such baseline dataset is publicly available to the scientific community. Then, we exploit the baseline dataset to train a set of machine-learning classifiers built over the reviewed rules and features. Our results show that most of the rules proposed by Media provide unsatisfactory performance in revealing fake followers, while features proposed in the past by Academia for spam detection provide good results. Building on the most promising features, we revise the classifiers both in terms of reduction of overfitting and cost for gathering the data needed to compute the features. The final result is a novel Class A classifier, general enough to thwart overfitting, lightweight thanks to the usage of the less costly features, and still able to correctly classify more than 95% of the accounts of the original training set.

The findings reported in this paper, other than being supported by a thorough experimental methodology and interesting on their own, also pave the way for further investigation on the novel issue of fake Twitter followers.

Keywords: Twitter, fake followers, anomalous account detection, baseline dataset, 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 October, 2014, statistics reported about a billion of Twitter subscribers,
with 284 million monthly active users, and around 180 billion of Twitter timeline views in Q3 2014 [1]. Twitter annual advertising revenue in 2013 has been estimated to $405,500,000 [2]. 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 [3]. 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 [4, 5].

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 mostly focused its attention on spammers, those accounts actively putting their efforts in spreading malware, sending spam, and advertising activities of doubtful legality, see, e.g. [6, 7, 8, 9]. To enhance their effectiveness, these malicious accounts are often armed with automated twitting programs, as stealthy as to mimic real users, known as bots. Even if such automated pieces of software could be designed and used to post legitimate messages as well, such as news updates, they can be turned to support malicious activity too, as shown, e.g., in a noticeable experiment carried out on Facebook [10].

In the recent past, media have started reporting that the accounts of politicians, celebrities, and popular brands featured a suspicious inflation of followers [11, 12, 13]. 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 [12]. 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 fake followers of major candidates [14].

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, banks and financial institutions in U.S. have started to analyze Twitter and Facebook accounts of loan applicants, before actually granting the loan. In particular, to take the decision, they consider the number of Twitter friends of the applicant and the frequency of his/her interactions on the social platform [15]. Thus, to have a “popular” profile can definitely help to augment the creditworthiness of the applicant. Similarly, if the practice of buying fake followers is adopted by malicious accounts, as 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 the formation of fictitious public opinion and severely impacting on the real world economy and society. These are keypoints why fake followers detection is an issue that is 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. Most of the scientific studies discriminate Twitter accounts by means of machine-learning classifiers trained over manually annotated sets of accounts.

To the best of our knowledge, however, despite fake followers constitute a spread phenomenon with both economical and social impacts, in the literature the topic has not been deeply investigated yet. There are (few) interesting works addressing the subject, thoroughly described in the Related Work section: they analyze how Twitter followers markets operate, concentrating on which are the market victims and customers [16][17] and on which services are exploited by merchants to massively register the fake accounts [18].

Instead, in this paper, we aim at addressing the characterization and consequent detection of fake Twitter followers using a classifier-based approach. We also try to limit the drawbacks of a manual construction of the training sets, as clarified in the following.

**Contributions**

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

First, we build a baseline dataset of Twitter accounts where humans and fake followers are known a priori.

Second, we test known methodologies for bot and spam detection on our baseline dataset. In particular, we test the Twitter accounts in our reference set against algorithms based on 1) single classification rules proposed by bloggers, and 2) feature sets proposed in the literature for detecting spammers. The outcome of the analysis suggest that fake followers detection deserves specialized mechanisms: specifically, algorithms based on classification rules do not succeed in detecting the fake followers in our baseline 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. Building on theoretical calculations and empirical evaluations, we show how the best performing features are also the most costly ones. The novel results of our analysis show that data acquisition cost often poses a serious limitation to the practical applicability of such features.

Finally, building on the crawling cost analysis, we design and implement lightweight classifiers that make use of the less costly features, while still being able to correctly classify more than 95% of the accounts of our training dataset. In addition, we also validated the detection performances of our classifiers over two other sets of human and fake follower accounts, disjoint from the original training dataset.
Roadmap

The remainder of this paper is structured as follows. Section 2 considers and compares related work in the area of Twitter spam and bot detection. Section 3 describes our baseline dataset. In Section 4 we evaluated a set of criteria for fake Twitter followers detection promoted by Social Media analysts using our baseline dataset. In Section 5 we examine features used in previous works for spam detection of Twitter accounts. In Section 6 we compute the cost for extracting the features our classifiers are based on. A lightweight and efficient classifier is also provided, attaining a good balance between fake followers detection capability and crawling cost. Finally, Section 7 concludes the paper.

2. Related Work

Quoting from [19], “A fake Twitter account is considered as one form of deception (i.e., deception in both the content and the personal information of the profiles as well as deception in having the profile follow others not because of personal interest but because they get paid to do so).” The second characterisation for deception is exactly the one we deal with in our paper. We specifically consider fake followers as those Twitter accounts appropriately created and sold to customers, which aim at magnifying their influence and engagement to the eyes of the world, with the illusion of a big number of followers.

So defined fake followers are only an example of anomalous accounts which are spreading over Twitter. Anomalies have been indeed identified in the literature as either spammers (i.e., accounts that advertise unsolicited and often harmful content, containing links to malicious pages [8]), or bots (i.e., computer programs that control social accounts, as stealthy as to mimic real users [10]), or cyborgs (i.e., accounts that interweave characteristics of both manual and automated behaviour [20]). Finally, there are fake followers, i.e., those accounts massively created to follow a target account and that can be bought from online accounts markets, as Intertwitter.com [17]. We would like to remark that fake followers could be seen as a macro category in the scenario of Twitter anomalous accounts, since subsets of fake followers could include bots or even stolen accounts of real users, as clarified in Section 2.3.

Hereafter, we illustrate methods and approaches that have been proposed in the academic literature to analyze the different phenomena, and we highlight similarities and differences with our approach.

2.1. Spam detection

In recent years, spam detection on Twitter has been the matter of several investigations, approaching the issue from several points of view. As an example, a branch of research focused on the textual content of tweets, by, e.g., mining it, as in [21], studying the redirection of embedded URLs in tweets, as in [22], and classifying the URLs landing pages, as in [23]. Work in [24] moves beyond the incapability of labeling those tweets without URLs as spam tweets, by proposing a composite tool, able to match incoming tweets with underlying templates commonly used by spammers.
Instead of considering the content of tweets, work in [19] tries to classify if an account can be trusted or not based on inconsistent information originating from the profile of the account only. In particular, the focus is on the use of a language-independent algorithm that classifies a Twitter profile as belonging either to a male or a female, only regarding the data in the user profile. As reference dataset, the authors used the gender information that the crawled accounts stated in other social platforms (such as in Facebook profiles).

A series of works investigate spammers on microblogging platforms through a multi-feature approach, including features on the profile, the behaviour, and the timeline of an account. Within these research line, we cite here [8], [6], and [9]. The work in [8] 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 the three social platforms, logged the kind of contacts and messages that they attracted, and manually analyzed the collected data. The analysis reported that the suspicious accounts shared some common traits, obtained by the authors leveraging a set of features. Those served as input to a machine learning-based classifier [25], 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 [6], 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 have higher detection rates, when compared to the existing ones.

In [9], the authors leverage a combination of behavioural features (such as tweeting and retwitting activities), network features (such as the number of an account’s followers and friends), and content-based features to develop a hybrid mathematical model for spammer detection in Weibo, the Chinese microblogging site resembling Twitter. To build a reference dataset to test their model, they crawled data of accounts that were friends of randomly selected users. This crawled set is supposedly made of legitimate accounts only, based on the observation that “normal” users do not usually follow spammers. Furthermore, Weibo provides a public list of real spammers, that was leveraged by the authors. The mathematical functions representing the “spam level” of an account have been tested on the reference dataset, obtaining high values for precision, recall, and F-measure metrics (metrics that will be detailed in the following).

2.2. Cyborgs and bots detection

The authors of [20] classify Twitter accounts in three classes: humans, bot, and cyborgs. The latter class represents either bot-assisted humans or 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 built 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 analysis (LDA) method.

The algorithm proposed in [26] aims at spotting groups of automated-malicious accounts, as quickly as possible, hopefully around their creation time. The quest was born from the observation that detection schemes relying on behavioral and timeline information “take a considerable time to collect such information before running detection algorithms, so criminals utilize their accounts until suspension and exploit others again”. Thus, the authors apply a clustering algorithm to group accounts created within a short period of time, considering some name-based features. Then, they adopt a classification algorithm to classify malicious account clusters. The process exploits the difference between algorithmically generated account names and human-made account names. As reference dataset, the authors collected the names of a set of verified Twitter accounts (namely the accounts with the Twitter ✓ mark), and considered them as the ground truth of human-made account names.

2.3. Fake followers and Account Markets Analysis

So called Twitter Account Markets are online services offering their subscribers to provide followers in exchange for a fee, and to spread promotional tweets on their behalf. In [16], the authors list several criteria to detect clients and victims of such markets. The results of the analysis reveal a surprising and alarming business behind this phenomenon. In a recent work [17], the same research team provides more details about the Account Markets, analyze additional properties and characteristics of their customers (e.g., the dynamics of followers and friends and the ability of generating engagement), and provide a classifier for the detection of both markets and market customers. As reference datasets for training the classifier, the authors have first subscribed to a set of markets, and then analyzed the followers of the subscribed accounts, to get a set of market victims. Studying the relationships among victims and their friends is the key in [17] to build the reference set for customers. It is worth noting that the selection of legitimate accounts was carried out by picking 1 million randomly-sampled users from the general Twitter population (imposing a threshold over a minimum number of followers).

The authors of [18] monitor prices, availability, and fraud perpetrated by a set of merchants of Twitter accounts over the course of a ten-months period. Such a research is a spotlight on techniques and methodologies that accounts markets exploit to create and register fraudulent accounts, from CAPTCHA solving services, to deceitful email credentials and a diverse pool of IP addresses to evade blacklisting. In collaboration with Twitter itself, the authors developed a classifier to detect such fraudulent accounts, which were consequently suspended.

2.4. Differences and similarities with our approach

The goal of our research is the automatic detection of those Twitter accounts specifically created to inflate the number of followers of some target account (so called fake Twitter followers). A priori, both
spammers, bots, and genuine users’ accounts could fall in the macro-category of fake followers, and specific features already proved effective in the literature for spotting spammers and bots could work also in the case of fake followers. It was indeed this observation that initially drove the authors of this paper towards the direction of testing rules and features from past works on a reference dataset of genuine accounts and fake followers. This contributed to prune those rules and features that behaved worst in detecting fake followers, and leave the ones that well behave.

From a technical point of view, in our experiments we rely on machine learning-based classifiers exploiting features of 1) profile, 2) activity, and 3) relationships of the accounts, as done, e.g., in [8, 6]. Instead, we do not rely on features inherent to specific contents of tweets, like the presence of URLs, the semantics of the text, etc. (as done in, e.g., [21, 23].

We move beyond the mere application of already tested features to a new dataset, since we revise our classifiers to reduce overfitting and cost for data gathering, as illustrated in sections 5 and 6.

Finally, similar to [17], we bought fake Twitter followers from different markets available on the Web. We conducted such an exercise independently from [17] and, moreover, goals of the two works are quite different, being ours concentrated on accounts sold by these markets, while the other concentrates more on customers. As for the genuine part of our baseline dataset, we recruit accounts of people that have voluntarily adhered to our campaign, and leverage a dataset of annotated accounts, belonging to people active on Twitter within a particular period of time on a specific domain, and whose authenticity has been verified (see Section 3). However, willing to test our classifiers over a representative sample of the entire Twitter population, we also approached the construction of a test set by randomly picking: 1) a sample of Barack Obama followers, and 2) a sample of the Twitter population (as in [17]). This is illustrated in Section 6.3. We do not rely on suspended Twitter accounts for verifying their authenticity (as done, e.g., in [27]) nor on correlations between accounts on different social platforms (as done, e.g., in [19]).

2.5. Grey literature and Online Blogs

In this subsection, we briefly report on documentation that, although not directly attributable to academic work, presents a series of intuitive detection criteria, though not proved to be effective in a scientific way. The reason why we cite this work is twofold: on the one hand, online articles and posts testify the quest for a correct discrimination between genuine and fake Twitter followers; on the other hand, some of the experiments in this paper prove the (in)effectiveness of such criteria, highlighting how even some of the most intuitive criteria result unsatisfactory to correctly discriminate between genuine and fake accounts.

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 [28] 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, i.e.
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.

A series of reports published by the firm Digital evaluations [29] have attracted the attention of Italian and European newspapers and magazines, raising doubts on the Twitter popularity of politicians and leading international companies [13, 11, 14]. 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 the latter case, there is not enough information to assess their nature), providing a quality score of the effective influence of the followed account. The results in [29], however, lack a validation phase.

Finally, some companies specialized in social media analysis, like [30, 31, 32], 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. Moreover, as demonstrated in our previous work [33], these analyses are affected by biases like small and statistically unsound follower samples.

In the next sections of the paper, we introduce a Twitter account dataset that we used to evaluate the performance on detecting fake followers accounts of five of the cited work, namely [8, 6, 29, 28, 31]. 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 noting how other works for spam detection, like [7, 34], base their results on subsets, or on only slightly modified versions, of the criteria considered by our selected set of works.

We distinguish the investigated five 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 distinguishes between the possible classes. The second type of algorithms extract properties from a set of pre-classified accounts and exploit them to train a model to distinguish among the possible classes. The first type of algorithms has been proposed for fake follower/bot account detection, while the second type has only 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. Baseline datasets

In this section, we present the datasets of Twitter accounts 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 fake followers. Despite the final size of the baseline dataset, to perform our
research, we altogether crawled 9 millions of Twitter accounts and about 3 millions of tweets. To foster investigation on the novel issue of fake Twitter followers, our baseline dataset has been made publicly available for research purposes. 

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 [wafi.iit.cnr.it/TheFakeProject](http://wafi.iit.cnr.it/TheFakeProject), where one may find instructions on how 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 promoted the project and invited people to join it (see [36] for an Italian version of the piece). 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 the Twitter APIs v1.1, we crawled a series of public information from these followers, i.e., their profiles and timeline information, together with that of their followers and followings. For this dataset, we crawled the 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 humans” all the 469 accounts out of the 574 followers that successfully completed the CAPTCHA. In the remainder of this section this dataset is referred to as TFP.

3.2. #elezioni2013 dataset

The #elezioni2013 dataset, henceforth E13, 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. The study focused 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/analyst, and congressperson. 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. 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. The manual verification process has been carried out by two sociologists from the University of Perugia, Italy. It involved the analysis of profile pictures, biographies, and timeline of the accounts under investigation. Accounts not having a biography or a profile picture have been discarded. URLs in biographies have also been manually checked to allow for a deeper analysis of the subject. Only accounts labeled as humans by both the sociologists have been included in the E13 dataset. Overall, the manual verification phase lasted roughly two months. As a result, 1481 accounts became part of dataset E13.

3.3. Baseline dataset of human accounts

The above introduced datasets form our final set, labeled HUM, of 1950 verified human accounts. It is worth noting how the two subsets differ from each other. The TFP set 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 E13 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. Baseline dataset of fake followers

In April, 2013, we bought 3000 fake accounts from three different Twitter online markets. In particular, we bought 1000 fake 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. Surprisingly, fastfollowerz and intertwitter gave us a few more accounts than what we paid for, respectively 1169 and 1337 instead of 1000. We crawled all these accounts to build a fastfollowerz dataset, labeled FSF, and an intertwitter dataset labeled INT. Instead, we were unable to crawl all the 1000 fake followers bought from twittertechnology since 155 of them got suspended almost immediately. The remaining 845 accounts constitute the twittertechnology dataset, which is labeled TWT.

We acknowledge that our fake followers dataset is just illustrative, and not exhaustive, of all the possible existing sets of fake followers. However, it is worth noting 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 was easily possible to find on the Web at the time of searching.

3.5. Baseline dataset of fake followers and human accounts

The final baseline dataset exploited in our experiments is composed of both fake and human profiles. In the following, we briefly discuss the distribution between fake and human accounts that has been chosen for this dataset. Many machine-learning techniques are affected by the imbalance of the natural distributions
of the minority and majority classes. This is why, for example, works in the literature have studied how the
decision tree-based techniques perform when varying the distribution of the training set. In particular,
Weiss and Provost in [37] have considered the performances of decision-tree based classifiers to predict the
samples of 26 different datasets, with different distributions between the minority and majority classes. The
conclusions of their investigation have shown that the metric used to evaluate the performance of the dif-
ferent classifiers changes the optimal distribution of the classes for the training set. For example, after their
empirical analysis, using accuracy as evaluation metric, 9 out of the 26 datasets have the optimal distribution
very different from the natural one, while, when using the AUC as evaluation metric, this number grows to
14 out of the 26. Moreover, the optimal distribution has an oversampling of the minority class (there are also
cases when the best classifier is obtained with an oversampling up to 90% of the minority class samples).

Here, we face another fundamental issue: we do not precisely know the real (natural) distribution of fake
followers and human accounts. The Twitter staff conjectures that the number of “false or spam accounts
should represent less than 5% of our MAUs” [38] (where MAUs refer to monthly active users, i.e., “Twitter
users who logged in and accessed Twitter (...) in the 30-day period ending on the date of measurement”).
However, MAUs can be assimilated neither to a random sample of Twitter accounts nor to the followers of a
given account. Moreover, if an account has bought fake followers, then its distribution of fake followers and
human followers can vary dramatically from the natural distribution that one can find, either among MAUs,
or among all the Twitter accounts in the Twittersphere. In conclusion, the estimation of 5% as false or spam
accounts, in the whole Twitter, can not be directly extended to the fake followers of a given account.

Although Twitter has never disclosed the total number of registered users, unofficial sources claim that
the Twitter accounts created up to date are many more than MAUs (see, e.g. statistics reported by analytics
companies like Statistic Brain, Twopcharts, and Semiocast). This is why, to operate under a conservative as-
sumption, we consider a balanced distribution of fake followers and human followers as our baseline dataset.

To obtain a balanced dataset, we randomly undersampled the total set of fake accounts (i.e., 3351) to
match the size of the HUM dataset of verified human accounts. Thus, we built a baseline dataset of 1950 fake
followers, labeled FAK. The final baseline dataset for this work includes both the HUM dataset and the FAK
dataset for a total of 3900 Twitter accounts. This balanced dataset is labeled BAS in the remainder of the
paper and has been exploited for all the experiments described in this work (where not otherwise specified).

Table I shows the number of accounts, tweets and relationships contained in the datasets described in this
section.

4. Algorithms based on classification rules

In this section, we detail three procedures, originally proposed by bloggers and social media analysts,
explicitly conceived for fake followers and bot accounts detection. These proposals were introduced in [26]
[28, 31]. The work we focus on in this section is not directly attributable to academic work. However, it
is an example of the spreading interest on the phenomenon of fake Twitter followers by Media and Social Marketing companies. Although we do not expect these proposals to satisfactorily perform for the complex task of fake followers detection, we believe a thorough analysis of the proposed criteria could still provide some useful insights. Coincidentally, all the procedures 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 outputs of the checks must be combined together in order to obtain the final classification. Unfortunately, in many cases, how to combine the criteria to obtain the final classification of an account is not publicly available. Details on how aggregation has been performed are provided in [29] only. Driven by the provided details, we implement the full algorithm described in [29] and we present its detection performances in Section 4.5. In addition, for each of the procedures, we report the criteria as indicated by the original sources and we further specify how we have implemented them into rules suitable to be applied over our datasets. We also detail the reasons for our implementation choices.

In this section, we mainly focus on the application of each single rule over our dataset to assess its strength (or weakness) in discriminating fake followers. In Section 6 we combine all the rules together with the features analyzed in Section 5 to assess their collective classification power. This is because a single rule – or feature – alone may not perform well in classifying fake and human accounts, but it may improve the detection if used in combination with other ones. Indeed, it is worth noting that some of the criteria analyzed in this section have been actually exploited by the classifiers built in Section 6.

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 [29] carried out a series of tests over samples of Twitter followers of Romney and Obama, for the last US presidential elections, as well as for popular Italian politicians. In [29] 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

<table>
<thead>
<tr>
<th>dataset</th>
<th>accounts</th>
<th>tweets</th>
<th>followers</th>
<th>friends</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFP (@TheFakeProject)</td>
<td>469</td>
<td>563,693</td>
<td>258,494</td>
<td>241,710</td>
<td>500,204</td>
</tr>
<tr>
<td>E13 (#elezioni2013)</td>
<td>1481</td>
<td>2,068,037</td>
<td>1,526,944</td>
<td>667,225</td>
<td>2,194,169</td>
</tr>
<tr>
<td>FSP (fastfollowerz)</td>
<td>1169</td>
<td>22,910</td>
<td>11,893</td>
<td>253,026</td>
<td>264,919</td>
</tr>
<tr>
<td>INT (intertwitter)</td>
<td>1337</td>
<td>58,925</td>
<td>23,173</td>
<td>517,485</td>
<td>540,658</td>
</tr>
<tr>
<td>TWT (twittertechnology)</td>
<td>845</td>
<td>114,192</td>
<td>28,588</td>
<td>729,839</td>
<td>758,427</td>
</tr>
<tr>
<td>HUM (human dataset)</td>
<td>1950</td>
<td>2,631,730</td>
<td>1,785,438</td>
<td>908,935</td>
<td>2,694,373</td>
</tr>
<tr>
<td>FAK (fake dataset)</td>
<td>1950</td>
<td>118,327</td>
<td>34,553</td>
<td>879,580</td>
<td>914,133</td>
</tr>
<tr>
<td>BAS (baseline dataset: HUM ∪ FAK)</td>
<td>3900</td>
<td>2,750,057</td>
<td>1,819,991</td>
<td>1,788,515</td>
<td>3,608,506</td>
</tr>
</tbody>
</table>

Table 1: Statistics about total collected data [35].
considering the gap between the sum of the two scores. In particular, the algorithm assigns to the examined accounts 1 (or more, where specified) human point for each of the criteria in Table 2. 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 8, 13, 14, 15, 16 and 17: 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 the platform used to post 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 8, 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 [28]. The “7 signals to look out for” to recognize Twitter bots are listed in Table 3.

| 1. the biography of the profile clearly specifies that it is a bot account; | 5. accounts that tweet from API are suspicious; |
| 2. the friends/followers ratio is in the order of 100:1; | 6. the response time (follow → reply) to tweets of other accounts is within milliseconds; |
| 3. the account tweets the same sentence to many other accounts; | 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. For rules 6 and 7, when looking for an account’s friends or followers list, Twitter only gives information about the current list, with no details about past friends or followers. Moreover, Twitter does not disclose any temporal data related to the moment a user stared following, or got followed by, another user. This means that the only way to check a user’s follow/unfollow behavior (rule 7) is to continuously monitor full friends and followers complete lists. The same applies with respect to the measurement of the delay experienced when a user follows (and replies to) other users (rule 6). As further detailed in Section 6, the Twitter rate limits in the use of the APIs makes it practically infeasible to monitor friends and followers lists of even a small group of users. Therefore, we did not apply rules 6 and 7 to our datasets, since that would require to continually monitor those accounts. This also means that those rules cannot be used to support an automatic detection process, since they require an interactive process to be evaluated.

4.3. Socialbakers’ FakeFollowerCheck

Several companies provide online tools to classify Twitter followers based on their fakeness degree. Here, we consider the “FakeFollowerCheck tool”, by Socialbakers [31]. 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 accounts. We contacted their customer service, but we were answered that “how the respective criteria are measured is rather an internal information”. The FakeFollowerCheck tool analyzes the followers of an account and considers them likely fake when the criteria listed in Table 4 are satisfied.

1. the ratio $\frac{\text{friends}}{\text{followers}}$ 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, he/she is following more than 100 accounts.

Table 4: Socialbakers rule set.

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 noting that the website reports the FakeFollowerCheck as a beta version, adding the following: “We are currently tweaking the algorithm”. Therefore, we stuck to the criteria published on the firm website.
in December 2013.

4.4. Evaluation methodology

All the criteria detailed above have been applied to the 2 verified human accounts datasets (TFP and E13) as well as to all the 3351 fake followers accounts bought from the Twitter account markets (FSF ∪ INT ∪ TWT), as described in Section 3.

We conducted one experiment for each rule, considering two classes of accounts, the fake followers and the human ones. To summarize the outcomes of each experiment, we considered some evaluation metrics based on four standard indicators, namely:

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

The meaning of each indicator is graphically highlighted by the matrix in Table 5 (called the confusion matrix [39]), where each column represents the instances in the predicted class, while each row represents the instances in the actual class:

<table>
<thead>
<tr>
<th>actual class</th>
<th>human</th>
<th>fake</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TN</td>
<td>FP</td>
</tr>
<tr>
<td></td>
<td>FN</td>
<td>TP</td>
</tr>
</tbody>
</table>

Table 5: Confusion matrix.

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

- **Accuracy:** the proportion of predicted true results (both true positives and true negatives) in the population, that is $\frac{TP + TN}{TP + TN + FP + FN}$;
- **Precision:** the proportion of predicted positive cases that are indeed real positive, that is $\frac{TP}{TP + FP}$;
- **Recall** (or also **sensitivity**): 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 $\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$;
- **Matthew Correlation Coefficient** (MCC from now on) [40]: the estimator of the correlation between the predicted class and the real class of the samples, defined as:

$$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}}$$
Each of the above measures captures a different aspect of the prediction quality of the samples that belong to the relevant class (the fake followers, in our dataset). Accuracy measures how many samples are correctly identified in both of the classes, but it does not express if the relevant class is better recognized than the other one. Moreover, there are situations where some predictive models perform better than others, even having a lower accuracy [41]. A high precision indicates that many of the samples identified as relevant are correctly recognized, but it does not give any information about the relevant samples which have not been identified. This information is provided by the recall metric (also known as sensitivity), that expresses how many samples, in the whole set of relevant samples, have been correctly recognized: a low recall means that many relevant samples are left unidentified. F-Measure and MCC try to convey in one single value the quality of a prediction, combining the other metrics. Furthermore, MCC is considered the unbiased version of the F-Measure, since it uses all the four elements of the confusion matrix [41]. An 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 guess, 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. For completeness, when available, we also report the area-under-the-curve metric ($AUC$), that is the area under the Receiver Operating Characteristic (ROC) curve. The latter is the curve that depicts the performance of a classifier considering the percentage of true positive samples compared with the percentage of false positive samples. The $AUC$ is used to summarize ROC curves in a single value: the more the area approaches to 1, the more performant is the classifier.

Finally, we also report the Information Gain ($I\ gain$) and the Pearson Correlation Coefficient ($Pcc$). While the Pearson correlation coefficient can detect linear dependencies between a feature and the target class, the information gain considers a more general dependence, leveraging probability densities (or frequencies, in case of discrete variables). More precisely, the information gain is a measure about the informativeness of a feature with respect to the predicting class and it is typically adopted to train machine learning classifiers. It can be informally defined as the expected reduction in entropy caused by the knowledge of the value of a given attribute [42]. In particular, given an attribute $A$ and the whole dataset $S$ of samples, the information gain $I(S, A)$ is defined as $I = H(S) - \sum_{a \in \text{Values}(A)} \frac{|S_a|}{|S|} H(S_a)$, where $\text{Values}(A)$ is the set of all possible values for attribute $A$, and $S_a$ is the subset of samples that have $a$ in the attribute $A$. We compute two information gains: $I\ gain$ about the outcome of the rule and $I\ gain^*$ about the attribute used by the rule. For $I\ gain$, a rule based on attribute $A$ can only assume the values 0 (not satisfied) and 1 (satisfied), while for $I\ gain^*$, the attribute $A$ can assume much heterogeneous values. For example, when evaluating the information gain of the rule “followers $\geq 30$”, a sample with 234 followers contributes with value 1 when we compute $I\ gain$, and with value 234 when we compute $I\ gain^*$.

The Pearson correlation coefficient, instead, is a measure of the strength of the linear relationship between two random variables $X$ and $Y$ [43], that in case of a sample set can be expressed as $r = \ldots$
### Table 6: Camisani-Calzolari algorithm prediction outcome.

<table>
<thead>
<tr>
<th>dataset</th>
<th>real humans</th>
<th>bots</th>
<th>neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFP (@TheFakeProject)</td>
<td>469</td>
<td>456</td>
<td>3</td>
</tr>
<tr>
<td>E13 (#eletzioni2013)</td>
<td>1481</td>
<td>1480</td>
<td>0</td>
</tr>
<tr>
<td>FSF ∪ INT ∪ TWT (100% fake foll.)</td>
<td>0</td>
<td>2889</td>
<td>185</td>
</tr>
</tbody>
</table>

### Table 7: Evaluation of Camisani-Calzolari algorithm (CC algorithm) [29]. Training set: BAS (1950 humans and 1950 fake followers). \(^\dagger\): the CC algorithm classified 163 accounts as neutral.

<table>
<thead>
<tr>
<th>evaluation metrics</th>
<th>accuracy</th>
<th>precision</th>
<th>recall</th>
<th>F-M.</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC algorithm(^\dagger)</td>
<td>0.548</td>
<td>0.974</td>
<td>0.062</td>
<td>0.116</td>
<td>0.175</td>
</tr>
</tbody>
</table>

\[
\frac{1}{n-1} \sum_{i=1}^{n} \left( \frac{X_i - \bar{X}}{\sigma_X} \right) \left( \frac{Y_i - \bar{Y}}{\sigma_Y} \right),
\]

where \( \bar{X} \) and \( \sigma_X \) are the sample mean and the standard deviation of \( X \), respectively [44]. Again, we compute \( Pec \), considering the outcome of the satisfaction of the rule (namely: \( true=1 \) or \( false=0 \)) and \( Pec^\ast \), based on the value assumed by the attribute used to evaluate the rule. Our experiments in the following sections will show that, generally, a rule and the corresponding attribute assume very different values for the information gain and the Pearson correlation coefficient.

#### 4.5. Evaluation of Camisani-Calzolari algorithm

The detection algorithm in [29] aggregates the twenty-two criteria for identifying human and bot behavior, 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 6 details the results of running the algorithm over the complete dataset, including the FAK set, namely all the bought fake followers accounts. Although obtaining very good results in detecting the real human accounts, the algorithm achieves a poor fake follower account detection. Most of the accounts have been erroneously tagged as humans too, mainly because the fake followers in our dataset have characteristics that easily make them achieve a human score higher than the bot one.

The above inability to detect the fake accounts is evident in the results of our second experiment. To evaluate the algorithm, we used it to predict the class of the accounts of our baseline dataset (BAS), reporting the evaluation of the final prediction in Table 7. As expected, the algorithm has a poor accuracy (very close to 0.5) and a high precision, meaning that the (few) accounts identified as fake are effectively fake. However, it also has a very low recall, meaning that many of the other fake accounts were unidentified as fake. This poor performance is also expressed by a F-Measure close to 0.1 and by the low MCC value.
4.6. Single rule evaluation

In this section, we analyze the effectiveness of each single rule, as designed by the original authors, in order to evaluate which rule can be considered as a good criterion for the detection of fake Twitter followers.

Table 8 summarizes the results obtained by the application of each rule introduced in sections 4.1, 4.2, and 4.3 to our BAS dataset. In Table 8, we highlighted the rules whose application results in high MCC values. As shown, 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, and (3) the use of a userID in at least one tweet.

As expected by the definition of MCC, such rules also exhibit a combination of high accuracy, precision, and recall. However, it is worth observing the values for the information gain and the Pearson correlation coefficient. The information gain of the rules (I gain) is always lower than the evaluation of the related attribute I gain*, while this is not true for the Pearson correlation coefficient (Pcc and Pcc*). Actually, this happens because Pcc evaluates the linear dependency between two variables that assume very similar values, namely the output of the rule and the class, while the Pcc* considers variables with more heterogeneous
values. In the first case, indeed, both the variables class and the output can assume only the values 0 and 1: the class can be either 0 (human) or 1 (fake), the rules can output either 0 (false, for example, \textit{account does not have more than 50 tweets}) or 1 (true, for example, \textit{account has more than 50 tweets}). Instead, for the $P_{cc^*}$, the attribute of a rule (in the example, the number of tweets) can assume much higher values (\textit{account has 234 tweets}). This is clearly not linearly dependent on the class values, see [43], with the effect to achieve lower values on the $P_{cc^*}$ with respect to the $P_{cc}$.

Thus, for each rule listed in Section 4.1 (top part of Table 8), it is meaningless to compare the $P_{cc}$ and $P_{cc^*}$ values. Instead, we need to focus only on the same type of metric, namely by column, to compare the linear dependency of the feature with the class. Then, directing our attention to the information gain, we notice that many of the rules take into account attributes that are effectively able to perform the discrimination between the two classes. If we consider as useful the rules and features that have an information gain value higher than 0.5, we observe that, even if many rules exhibit a very low $I \text{ gain}$, their “feature” version becomes much more interesting: for example, rules 18, 20, 21 and 22 have an evident increase in their information gain when used as features. Thus, we can derive that the rule is based on a right assumption (for example, the use of hashtags), but the rule definition is too simple to be effective: the algorithm proposed by [29] is simply too naive for the complex task of fake accounts detection in Twitter. Coincidentally, we have that the best performing rules also show the highest $P_{cc}$ values, namely their satisfaction is more strongly related to the belonging class. Concerning the features underlying the rules, we find that the $P_{cc^*}$ is strongly reduced because, as above noticed, they can (and indeed do) assume very high values and this severely affects the linear correlation with the class.

Observing the other rules of Table 8, we can notice how none of the criteria suggested by online blogs and by Socialbakers’ FakeFollowerCheck are successful in detecting the fake followers in our dataset. As an example, all rules by Van Den Beld have accuracy and precision close to 0.5 or a very low recall. Also, we observe that “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 followers 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 (are supposed to) almost exclusively post tweets using API. The relevance to the prediction task is also confirmed by both the $I \text{ gain}/I \text{ gain^*}$ and the $P_{cc}/P_{cc^*}$ values.

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, resulting in a high precision, accuracy around 0.5 and very low recall. The exception is rule 6, “0 tweets”: as a rule, it has an information gain value of 0.02, but when considered as a feature (i.e., number of tweets) it obtains 0.621. Similarly, rules 4 and 5 are much more useful for the detection process when considering their underlying features (namely, number of retweets and
number of tweets with URLs). Summarising, independently from the typology of the account, the rules are almost always satisfied, leading to a severe flaw when dealing with fake followers detection.

5. Evaluation of algorithms based on feature sets

In this section, we examine works in [8, 6] 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 feature 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.

Differently from the rule-based algorithms in Section 4, features are here presented as quantifications of properties of the considered samples. Therefore, they are introduced without any prior knowledge about the values for the features that will characterize the considered classes. Only after the training phase, it will be possible to observe which are the most frequent values for the features within the different classes.

For our evaluation analysis, we adopt classifiers that produce “glass-box” models, i.e., their inner structure can be understood by humans, also providing insights on how the models identify fake accounts. We leave further analyses that make use of “black-box” models (like Artificial Neural Networks or Support Vector Machines) as future work.

5.1. Detecting spammers in social networks

The study presented in [8] focuses on spambot detection. The authors exploit five characteristics that can be gathered crawling an account’s details, both from its profile and timeline. The characteristics 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 characteristics are exploited in a Random Forest algorithm [25, 45], that outputs if the account is a spambot or not. The results of the analysis in [8] depicted some interesting features of the spambot accounts under investigation, as reported in Table 9.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>spambots do not have thousands of friends;</td>
</tr>
<tr>
<td>2.</td>
<td>spambots have sent less than 20 tweets;</td>
</tr>
<tr>
<td>3.</td>
<td>the content of spambots’ tweets exhibits the so-called message similarity;</td>
</tr>
<tr>
<td>4.</td>
<td>spambots have a high ( \frac{\text{tweets containing URLs}}{\text{total tweets}} ) ratio value (i.e., lower ratio values mean legitimate users);</td>
</tr>
<tr>
<td>5.</td>
<td>spambots have a high ( \frac{\text{followers}}{\text{friends}^2} ) ratio value (i.e., lower ratio values mean legitimate users).</td>
</tr>
</tbody>
</table>

Table 9: Feature set proposed by Stringhini et al. [8].
We briefly give some notes on how we use the features of [8] over our dataset. To evaluate 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 later work by the same authors [10].

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 BAS dataset. The results are reported in Table 12 of Section 5.3.

5.2. Fighting evolving Twitter spammers

The authors of [6] 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. In Table 10 we detail nine of them, together with the outcome of their analysis in [6].

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Observation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. age</td>
<td>Age of the account: the more an account is aged, the more it could be considered a good one;</td>
<td>The feature has been found to be commonly higher for legitimate accounts than for spammers;</td>
</tr>
<tr>
<td>2. bidirectional link ratio</td>
<td>Bidirectional link ratio ((\frac{\text{bidirectional links}}{\text{friends}})), where a bidirectional link occurs when two accounts follow each other: this feature has been tested to be lower for spammer accounts than for legitimate accounts;</td>
<td>This feature has been found to be lower for spammer accounts than for legitimate accounts;</td>
</tr>
<tr>
<td>3. average neighbors’ followers</td>
<td>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. The feature has been found to be higher for suspicious accounts;</td>
<td>This feature has been found to be higher for legitimate accounts than for spammers;</td>
</tr>
<tr>
<td>4. average neighbors’ tweets</td>
<td>The average number of tweets of the account’s followers. This feature is lower for spammers than for legitimate accounts;</td>
<td>The feature has been found to be lower for spammers than for legitimate accounts;</td>
</tr>
<tr>
<td>5. followings to median neighbor’s followers</td>
<td>Defined as the ratio between the number of friends and the median of the followers of its friends. This feature has been found higher for spammers than for legitimate accounts;</td>
<td>This feature has been found higher for spammers than for legitimate accounts;</td>
</tr>
<tr>
<td>6. API ratio</td>
<td>API ratio (tweets sent from API / total number of tweets): higher values for suspicious accounts;</td>
<td>The notion of API ratio is as in Section 5.1. This metric is higher for suspicious accounts;</td>
</tr>
<tr>
<td>7. API URL ratio</td>
<td>API URL ratio (tweets sent from API and containing URLs / total number of tweets sent from API): such ratio is higher for suspicious accounts;</td>
<td>The notion of API URL ratio is as in Section 5.1. This metric is higher for suspicious accounts;</td>
</tr>
<tr>
<td>8. API tweet similarity</td>
<td>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. This metric is higher for suspicious accounts;</td>
<td>The notion of API tweet similarity is as in Section 5.1. This metric is higher for suspicious accounts;</td>
</tr>
<tr>
<td>9. following rate</td>
<td>Following rate: this metric reflects the speed at which an account follows other accounts. Spammers usually feature high values of this rate.</td>
<td></td>
</tr>
</tbody>
</table>

Table 10: Feature set proposed by Yang et al. [6].

The authors of [6] combine their features in four different machine learning classifiers and compare their implementation with other existing approaches. We were unable to completely reproduce the machine learning classifiers in [6], since we had a different dataset. Instead, here we evaluate how those features, which proved to be quite robust against evasion techniques adopted by spammers, perform in detecting fake Twitter followers. As in [6], the following rate (feature 9) has been approximated with the ratio \(\frac{\text{friends}}{\text{age}}\), since a precise evaluation would require to know the evolution of the number of friends of an account, but this is, indeed, publicly unavailable.

Interestingly, the authors of [6] 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.

21
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 them using a simplified approach. For this reason, we have not implemented them in our analysis.

Finally, note that in [6] 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.

5.3. Evaluation

As done for the rule set in Section 4, we report in Table 11 the evaluation of the information gain and the Pearson correlation coefficient for all the features within the BAS dataset. Also in this case, since the Pcc evaluates the linear dependence between the considered feature and the class (that can only be 0 or 1), it produces results that are slightly different when compared to the information gain. Observing the results in Table 11, we can identify several promising features: “number of tweets” (already noticed in Section 4), “ratio between friends and followers”², “bidirectional links ratio” and “API ratio”. The beneficial effect of the bi-link ratio will be further confirmed by the experiments of Section 5.4.3.

<table>
<thead>
<tr>
<th>feature description</th>
<th>evaluation metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stringhini et al. [5]</td>
<td></td>
</tr>
<tr>
<td>1 number of friends</td>
<td>0.263</td>
</tr>
<tr>
<td>2 number of tweets</td>
<td>0.621</td>
</tr>
<tr>
<td>3 content of tweets</td>
<td>0.444</td>
</tr>
<tr>
<td>4 URL ratio in tweets</td>
<td>0.401</td>
</tr>
<tr>
<td>5 friends/followers^2</td>
<td>0.733</td>
</tr>
<tr>
<td>Yang et al. [6]</td>
<td></td>
</tr>
<tr>
<td>1 age</td>
<td>0.539</td>
</tr>
<tr>
<td>2 bidirectional links ratio</td>
<td>0.905</td>
</tr>
<tr>
<td>3 avg. followers of friends</td>
<td>0.327</td>
</tr>
<tr>
<td>4 avg. tweets of friends</td>
<td>0.203</td>
</tr>
<tr>
<td>5 friends / med. foll. of friends</td>
<td>0.336</td>
</tr>
<tr>
<td>6 api ratio</td>
<td>0.544</td>
</tr>
<tr>
<td>7 api url ratio</td>
<td>0.058</td>
</tr>
<tr>
<td>8 api tweet similarity</td>
<td>0.46</td>
</tr>
<tr>
<td>9 following rate</td>
<td>0.355</td>
</tr>
</tbody>
</table>

Table 11: Evaluation of the single feature.
To evaluate the combined effectiveness of the feature sets described in sections 5.1 and 5.2 on detecting fake follower accounts, we used five classifiers obtained with 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 [45]. Random Forest was the only used by the authors of [8] and all of them, but Adaptive Boost, were used by the authors of [6] to build spam detection classifiers. We also included AB since it is considered one of the most effective machine learning algorithms for classification tasks. For both the considered works, we built five classifiers adopting the suggested features, and training the models using our baseline BAS dataset. Then, we used a 10-fold cross validation [25] to estimate the performances of each obtained classifier. As for the rule-based algorithms in Section 4.4, we look at the MCC as the preferred metric to assess classifier performances. The obtained results are summarized in Table 12.

We observe that all the classifiers have an excellent prediction capability. The ones built over the feature set by [6] obtain slightly better results. In particular, RF, J48 and D classifiers have an MCC above 0.98. Similarly, precision and recall are around 0.99 for all of them. Only the accuracy and recall for BN are a bit smaller when compared to precision, meaning that BN misclassified some fake followers accounts as humans at a rate higher than other ones. Moreover, all the classifiers based on the feature set by [6] have a higher AUC, when compared with the ones built with the feature set by Stringhini et al. [8]. However, the latter one also obtains extremely high detection levels: accuracy, precision, and recall are around 0.98 for RF, D and J48, with an MCC of around 0.96. The lower precision and recall with respect to the ones obtained using the set of Yang et al. [6] show that the features of Stringhini et al. [8] exhibit the tendency to consider as fake followers some human accounts. Again, BN achieves the lowest performances among all and, as all the others, it has a reduced precision w.r.t. the BN built with the features of Yang et al. [6].

Overall, even if some small differences can be observed in the evaluation metrics, all the classifiers almost correctly distinguish between human and fake follower accounts, in our baseline BAS dataset. The feature-
based classifiers are indisputably more accurate for fake follower 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 between human and fake followers. In the 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 human and fake followers. It is worth noting that also 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 BAS dataset and the spam accounts characterized in [8] and [6]. For example, the feature \( URL \) ratio has been found to have a higher value for spammers than for legitimate users, as highlighted in [8] (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 baseline dataset. 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 ([6], 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 average neighbor’s tweets feature, that has been found to be lower for spammers in [6], but higher for our fake followers.

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 automated 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 [46]. 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 Stringhini et al. [8] has 22 leaves, 43 nodes, and a height of 7, whereas the best feature is the \( \text{friends}/(\text{followers}^2) \) ratio that places at the root. The decision
tree with the feature set of Yang et al. [6] 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 [25, 45]. Adopting this strategy, we were able to obtain simpler trees with a lower number of nodes and a very reduced height. Such simpler trees generally use subsets of the feature set, still maintaining very good performance on our BAS dataset.

Table 13 reports the characteristics and the performance of the experiments we have carried out, varying the pruning strategy. It is worth noting that the complexity of the tree is not always directly connected to an improvement in the detection capability: for example, for the feature set of Yang et al. [6], reducing the number of nodes from 33 to 11 decreases the accuracy of 0.007 and the MCC of 0.014, only. Similarly, the values for AUC remain almost the same between the pruned and the not pruned versions of the tree. Even if precision decreases a bit more, it always remains very high (above 0.95 for [8] and above 0.97 for [6]). This means that only a small fraction of humans were erroneously considered as fakes. Moreover, we clearly observe that the pruned version of Stringhini et al. [8] reduces its recall of 0.017, while the one of Yang et al. [6] only drops of 0.004, meaning that the latter is able to miss fewer fakes than the former one after pruning. This is also evident from the higher reduction of both F-Measure and MCC. We think that this increased effectiveness is a direct consequence of the quality of the used features. Overall, 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 lines of Table 13 for [8] and [6], respectively). For instance, 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 Yang et al. [6], and the friends/(followers^2), URL ratio, and number of friends as the subset for the Stringhini et al. [8] original feature set.
5.4.3. Bidirectional link ratio

In Section 5.3 we observed that the *bidirectional link ratio* had the highest information gain among all the considered features. In order to test if this 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 those of a new one. We build a decision tree classifier leaving out the bi-link ratio from the feature set of Yang et al. [6] and compare its effectiveness against the classifier built with the complete set. The results are reported in the last rows of Table 13.

This experiment is particularly interesting since, as detailed in next Section 6, this feature is the most expensive to evaluate, especially in terms of crawling. The results in Table 13 show an evident decrease in accuracy, precision and recall for the less pruned trees (subtree raising 0.25 and reduced error 3 folds), as well as for both $F$-Measure and $AUC$, and an even noticeable decrease of the $MCC$ measure. The reduced error pruning with 50 folds produces a classifier that has $MCC$ dropping from 0.966 to 0.866. Its detection level ($accuracy$) is still very good for all the three pruned trees (0.964, 0.982 and 0.933, respectively), but we can clearly observe a remarkable drop in both precision and recall, compared to the version with the whole feature set. This suggests that the highest effectiveness we noticed in the above experiments after pruning (Section 5.4.2) is considerably lost when we do not consider the bi-link ratio feature. The most 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 nodes.

From the results shown in Table 13 we conclude that the bidirectional link ratio is an important feature for fake follower detection: even if not essential, it is extremely effective for the detection process. By capturing the nature of the social ties between an account and its neighbors, this feature is intrinsically harder to beat than those based on simpler characteristics, like, e.g., other information in the account’s profile. On the one hand, features based on accounts’ relationships are robust with respect to evading techniques that can be applied by evolving fake follower originators [6]. On the other hand, such robust relationships-based features are usually more demanding in terms of data to be downloaded and, consequently, of time needed to perform the analysis, making them less suitable for massive fake follower detection. This point will be explored in greater detail in the next section.

6. An efficient and lightweight 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. [6] seems to be slightly more effective than the one proposed by Stringhini et al. [8], when used in feature-based classifiers aiming at fake followers detection. Here, we look for an efficient and lightweight classifier, exploiting the best features and the best rules, not only in terms of detection performance, but also considering their evaluation.
<table>
<thead>
<tr>
<th>Feature set</th>
<th>Class A (profile)</th>
<th>Class B (timeline)</th>
<th>Class C (relationships)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camisani-Calzolari</td>
<td>has name, has image, has address, has biography, followers ≥ 30, belongs to a list, tweets ≥ 50, URL in profile, 2*followers ≥ friends</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>State of search</td>
<td>bot in biography,</td>
<td></td>
<td>same sentence to many accounts,</td>
</tr>
<tr>
<td>Socialbakers</td>
<td>friends ≃ 100, duplicate profile pictures</td>
<td></td>
<td>tweet from API</td>
</tr>
<tr>
<td>Stringhini et al. [8]</td>
<td>number of friends, number of tweets,</td>
<td>tweet similarity, URL ratio</td>
<td></td>
</tr>
<tr>
<td>Yang et al. [6]</td>
<td>age, following rate</td>
<td>API ratio, API URL ratio, API tweet similarity</td>
<td>bi-link ratio, average neighbors’ followers, average neighbors’ tweets, followings to median neighbor’s followers</td>
</tr>
</tbody>
</table>

Table 14: Feature crawling cost classes.

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 about 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 Twitter API calls needed to download data required for the calculation. Results are in Tables 15 and 14. The following parameters refer to the Twitter account for which the number of fake followers is being investigated:

- \( f \): 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 15 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 one single API call is sufficient to get all the data for a single follower. For the Worst-case we can only 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, just to provide an estimation, we can consider the account with the highest number of followers on Twitter, which, at the time of writing, belongs to the pop star Katy Perry (@katyperry), with about 60 millions of followers. We can therefore consider as the worst-case scenario an account with 60 millions followers and 60 millions friends, which leads to a number of API calls equal to \( 22000 \times f \).

Observing the values of Table 15 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.

<table>
<thead>
<tr>
<th>profile</th>
<th>timeline</th>
<th>relationships</th>
</tr>
</thead>
<tbody>
<tr>
<td>API calls</td>
<td>( \sum_{i \in f} \frac{t_i}{200} )</td>
<td>( \sum_{i \in f} \left( \left\lceil \frac{f_i}{5000} \right\rceil + \left\lceil \frac{\varphi_i}{5000} \right\rceil \right) )</td>
</tr>
<tr>
<td>Best-case</td>
<td>( \left\lceil \frac{f}{100} \right\rceil )</td>
<td>( \left\lfloor \frac{f}{100} \right\rfloor )</td>
</tr>
<tr>
<td>Worst-case</td>
<td>( \left\lfloor \frac{f}{100} \right\rfloor )</td>
<td>( \left\lceil \frac{f}{5000} \right\rceil )</td>
</tr>
<tr>
<td>Calls/min.</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 15: Number of API calls needed to download data.
Furthermore, to protect Twitter from abuse, the number of API calls allowed per minute is limited. In Table 15, 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.

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 concurrently. 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 should not consider the sum of the time needed for each of the three cost classes, but just the most costly one.

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 14. 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. Then, all the classifiers of the previous sections are classifiers of Class B, with the exception of the classifier with the feature set of Yang et al. [6], that belongs to Class C.

In the following, we consider a lightweight classifier working only with features of Class A. We aim at verifying whether the Class A classifier reaches performances that are comparable to those of the most expensive Class B and Class C classifiers.

Table 16 reports the results of the classifiers built on our BAS dataset, using two different feature sets: all the features (independently from their cost) and the Class A features. We start observing that the classifiers built over all the features considered in our study perform better than all the others, including the classifiers using the feature sets of Yang et al. [6] and Stringhini et al. [8] in Table 12, with an AUC of 0.999, for three of them. However, the increase of MCC between the best classifier and our lightweight Class A classifier is very limited, i.e., around 0.02 for RF, D and J48. The AUC reduction is even smaller, only 0.004 for the

<table>
<thead>
<tr>
<th>algorithm</th>
<th>accuracy</th>
<th>precision</th>
<th>recall</th>
<th>F-M.</th>
<th>MCC</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class C classifiers that use all the features</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RF Random Forest</td>
<td>0.994</td>
<td>0.997</td>
<td>0.990</td>
<td>0.994</td>
<td>0.987</td>
<td>0.999</td>
</tr>
<tr>
<td>D Decorate</td>
<td>0.993</td>
<td>0.993</td>
<td>0.993</td>
<td>0.993</td>
<td>0.987</td>
<td>0.999</td>
</tr>
<tr>
<td>J48 Decision Tree</td>
<td>0.992</td>
<td>0.991</td>
<td>0.992</td>
<td>0.992</td>
<td>0.983</td>
<td>0.993</td>
</tr>
<tr>
<td>AB AdaBoost</td>
<td>0.987</td>
<td>0.988</td>
<td>0.987</td>
<td>0.987</td>
<td>0.975</td>
<td>0.999</td>
</tr>
<tr>
<td>BN BayesNet</td>
<td>0.960</td>
<td>0.965</td>
<td>0.954</td>
<td>0.960</td>
<td>0.921</td>
<td>0.991</td>
</tr>
</tbody>
</table>

| Class A classifiers that use only Class A (profile) features |
| RF Random Forest     | 0.987    | 0.993     | 0.980  | 0.987    | 0.967| 0.995|
| D Decorate           | 0.984    | 0.987     | 0.981  | 0.984    | 0.964| 0.995|
| J48 Decision Tree    | 0.983    | 0.987     | 0.979  | 0.983    | 0.962| 0.983|
| AB AdaBoost          | 0.972    | 0.975     | 0.969  | 0.972    | 0.941| 0.995|
| BN BayesNet          | 0.966    | 0.969     | 0.963  | 0.966    | 0.928| 0.991|

Table 16: Performance comparison for 10-fold cross validation. Training set: BAS.
same classifiers, even 0 for BN. The reduction of both accuracy and recall is slightly higher than that of precision, which still remains very high (always above 0.969). It is worth noting that the Class A classifier with BN outperforms the Class C competitor, noticeably obtaining an increase in all the metrics, but the AUC. Concerning the complexity of the two decision trees obtained with the J48 algorithm, we report that they are comparable, since both of them are composed by 31 nodes, 16 leaves and they have a height of 7. Another interesting observation is that both the classifiers select features that belong to other feature sets than the ones considered in the above Section 5 for the spam detection. For example, they include some features derived from the rules, like has biography, proposed in [29] for fake followers detection.

The analysis carried out in Section 6.1 highlights the effort and time needed to compute many of the features commonly proposed for the detection of spam and fake accounts. As shown by the Katy Perry example, crawling costs for some of the proposed features are totally infeasible for accounts with hundreds of thousands or millions of followers. For this reason, we trained and evaluated the proposed Class A classifiers, which achieve overall good performances, while only exploiting cost-efficient features. The Class A classifiers thus represent a feasible solution for the investigation of fake followers on a large scale.

We have to point out, however, that countermeasures could be taken to evade some of the simplest features our classifiers are built upon [6]. This would require to continually monitor and update the choice of such features, to keep pace with the fake follower generators. While contemporary fake Twitter followers are effectively and efficiently spotted by our Class A classifier, we can consider the use of the most expensive Class B or even Class C features to have stronger evidences about the more suspicious followers.

6.3. Validation of the Class A classifier

In this section, we propose a validation of our Class A classifier, built with our baseline BAS dataset. In particular, we set up two different experiments based on a random sampling of Twitter accounts.

For the first experiment, we built a set of 1000 Twitter accounts, randomly selecting numeric Twitter user IDs, ranging from user ID 12 (the very first valid Twitter account – @jack – belonging to Jack Dorsey, founder
of Twitter) to the user ID representing the last Twitter account created at the time of our experiment. This represents an unbiased sample of all the currently available (i.e., not closed, banned or suspended) Twitter accounts. This test set therefore comprises a broad range of accounts created during the eight years since Twitter’s advent. For the second experiment, instead, we consider a random sample of 1500 accounts among the followers of the US President Barack Obama – @BarackObama. This experiment resembles the practical application scenario of the proposed classifier, since it investigates a sample of a single account’s followers, in this case a major politician.

All the accounts acquired with the two aforementioned approaches have been labeled as humans, following the same approach used in [17]. Twitter officially reports that fake and spam profiles together are less than 5% of all registered accounts [38], therefore we are confident that just a few among the accounts labeled as humans might actually be fake ones. Automatically labelling the sampled accounts as humans would result in an error of at most 5%, still allowing an overall correct validation of our classifiers. In addition, many of the accounts randomly acquired for the first and second experiment show few signs of activity on Twitter (more than 70% of them did not post a tweet in the 3 months prior to our data acquisition). Thus, more reliable checks like CAPTCHA-based verifications would result in very sparse answers. Furthermore, we believe that also including less active accounts in our test sets, allows to validate our classifiers with Twitter accounts having different characteristics than those of our baseline dataset of humans in Section 3.3 (e.g., showing fewer “human” features). Together with the human accounts, the test set also includes the 1401 fake followers we bought, but not included in the BAS dataset that we used as training set (Section 3.5).

We report the results of the experiments on the test sets in Table 17. Observing the validation on the random sampled accounts, we can see that the three Class A classifiers based on RF, D and J48 obtain an accuracy above 0.9, a precision of 0.9 and a recall above 0.94: this means that they are able to spot almost all the fake followers of the test set. This is particularly true for the best performing classifier, RF, that reaches 0.975 for both accuracy and recall, with a precision of 0.982. The highest performances are also shown by both F-measure and MCC values, that are noticeably higher for RF when compared with the others. AB and BN obtain lower results for both accuracy and precision, but still a noticeably high recall, meaning that only few fake followers are left behind, but a considerably higher number of human accounts are classified as fake followers.

Since the accounts labeled as fake in both the test sets are the same, the values of the recall for the second experiment on the random sampled set of Obama followers are exactly the same as above. However, accuracy and precision are very close to the results obtained with the other test set for both D and J48: the results are comparable, showing an accuracy greater than 0.9 and a precision close to 0.87, with a reduction of 0.025 on average. This is also confirmed by both the F-measure and MCC that are very close to the results above. AB and BN noticeably switch their performances: the Adaptive Boosting algorithm raises its accuracy of 0.1 and its precision of around 0.08, outperforming the Bayesian Network-based one. This
latter loses 0.105 of accuracy, meaning that many of the accounts we labeled as humans were recognized as fake followers. Also the RF classifier loses 0.4 of accuracy and a considerable 0.1 in precision, meaning that it considers as fake followers many of the sampled Obama’s followers. We can finally observe that the AUC metric is always very high and that does not consistently reflect the real performances obtained by the five classifiers, as the F-measure and the MCC actually do.

7. Conclusions

In this paper, we focused on efficient techniques for fake Twitter followers detection.

To reach the goal, we firstly created a baseline dataset of human and fake follower accounts, the latter being bought from available online markets. Then, we surveyed various proposals for spammer and bot detection, based on classification rules and feature sets. Such proposals come part from Academia and part from the grey literature. Such rules and features, coming from this extensive state of the art, were eventually tested on our dataset to understand their effectiveness in detecting fake Twitter followers. A few features were selected and used by a set of classifiers that we have trained on the baseline dataset. Going further, we ranked the best performing features according to their crawling cost. This led us to identify three categories of features leading to three different, increasing, cost classes. Finally, we built a series of classifiers that only leverage cost-effective (Class A) features. With this final outcome, we were able to achieve detection rates comparable with the best of breed classifiers, whereas these latter necessitate overhead-demanding features.

Among all the analyzed features, we have seen that those yielding the best results are the ones based on the friends and followers of the account under investigation, such as the bidirectional link ratio and the friends/(followers^2) ratio: evaluating the social ties between an account and its neighbours, these features are more effective than those based on simpler characteristics, like, e.g., other information in the account’s profile. However, relationships-based features are more demanding in terms of data to be downloaded and, consequently, they require a significant analysis time, making them unsuitable for analyses on massive amounts of followers. Timeline-based (Class B) features have been shown to be less time-demanding, while still effective; hence, these might represent a promising trade-off between efficient and accurate detections. Eventually, as shown by the proposed Class A classifiers, the detection of currently available fake Twitter followers is possible even without leveraging resource-demanding features, by means of efficient algorithms and accurate selection and combination of less-demanding features.

Among the results of this study, there is the construction of a baseline dataset of verified human and fake follower accounts. To foster research on the novel issue of fake Twitter followers, we have publicly released our dataset to the scientific community.

We mainly resorted to “glass-box” classifiers which produce interpretable models. An analysis of the inner structure of such models allows to get insights into the mechanisms exploited to perform the detection. Nonetheless, we believe that the adoption of more complex (“black-box”) classification techniques,
such as Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs), could provide interesting results. Although not allowing an understanding of their inner working, ANNs and SVMs have been recently employed in a broad range of diverse classification tasks, achieving excellent results. Therefore, we consider this an open issue requiring further investigation.

As future work, we aim at designing and testing other advanced features that could be added to our lightweight fake detector classifier, leveraging additional characteristics of Twitter accounts. Data that could be further exploited for the classification task are the content of tweets and the accounts’ behavior. In particular, bot development forums represent a fruitful source of information to know more about bot/fake/spam accounts design, in terms, e.g., of similarities and differences in their behaviour. As highlighted by our work, the difficulty of the detection task and the massive numbers of accounts to analyze ask for the adoption of features that are not only effective, but also efficient, with regard to their crawling costs. Therefore, we believe that future works along this line of research should consider the balance between the predictive power of new features and their cost. This would allow to improve the detection of fake Twitter followers, while still retaining a scalable approach, making en masse analysis practically feasible.

It is foreseeable that countermeasures will be taken to masquerade a fake account with respect to some of the simplest features which our classifiers are built upon. This would require to continually monitor and update the choice of such features, conceivably studying (or directly interacting with) the same generators of fakes, to keep their pace. We believe that the features renovation process can be considered as another interesting direction for future research.

Acknowledgements

The authors would like to thank the anonymous reviewers who helped improving the quality of the manuscript.

References

Fame for sale: efficient detection of fake Twitter followers

Supporting material

S1. Structure of the decision trees

In this supporting section we report the structure of the decision tree models obtained within our study and reported in the other sections of the original article.

S1.1. Yang et al., subtree raising pruning, confidence factor 0.25

Tree obtained from the experiment with the Yang et al. [6] feature set of Section 5.3 with performance reported in Table 12 and in the first row of Table 13.

Number of leaves: 17
Size of the tree: 33
Height of the tree: 8

<table>
<thead>
<tr>
<th>bi_links_ratio &lt;= 0.059968</th>
<th>avg_foll_of_fri &lt;= 447559</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>following_rate &lt;= 0.173313</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bi_links_ratio &gt; 0.059968</th>
</tr>
</thead>
<tbody>
<tr>
<td>age &lt;= 264</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

S1.2. Yang et al., reduced error pruning, 3 folds

Tree obtained from the experiment with the Yang et al. [6] feature set of Section 5.4.2 with performance reported in second row of Table 13.

35
Number of leaves: 10
Size of the tree: 19
Height of the tree: 5

bi_links_ratio <= 0.05877
| following_rate <= 0.151116
| | age <= 266: fake (28.85)
| | age > 266: human (16.45/8.0)
| following_rate > 0.151116: fake (1230.91/1.0)
bi_links_ratio > 0.05877
| age <= 266
| | api_tweet_similarity <= 0
| | | api_ratio <= 0.444444: fake (16.98/3.0)
| | | api_ratio > 0.444444: human (2.24/0.24)
| | api_tweet_similarity > 0: human (6.0/1.0)
| age > 266
| | bi_links_ratio <= 0.174777
| | | following_rate <= 0.326308: human (64.17/2.0)
| | | following_rate > 0.326308
| | | | age <= 887: human (10.07/2.07)
| | | | age > 887: fake (14.0/1.0)
| | | bi_links_ratio > 0.174777: human (1210.32/0.95)

S1.3. Yang et al., reduced error pruning, 50 folds

Tree obtained from the experiment with the Yang et al. [6] feature set of Section 5.4.2 with performance reported in third row of Table 13.

Number of leaves: 6
Size of the tree: 11
Height of the tree: 3

bi_links_ratio <= 0.059968
| avg_foll_of_fri > 447559
| | fri_med_foll_of_fri_ratio <= 0.012182: human (9.01/1.01)
| | fri_med_foll_of_fri_ratio > 0.012182: fake (17.03/1.01)
bi_links_ratio > 0.059968
| age <= 255
| | fri_med_foll_of_fri_ratio <= 0.004951: fake (17.71/1.0)
| | fri_med_foll_of_fri_ratio > 0.004951: human (7.51/1.51)
| age > 255: human (1913.09/26.55)

S1.4. Stringhini et al., subtree raising pruning, confidence factor 0.25

Tree obtained from the experiment with the Stringhini et al. [8] feature set of Section 5.1 with performance reported in Table 12 and in the first row of Table 13.

Number of leaves: 22
Size of the tree: 43
Height of the tree: 7

fri_foll2 <= 0.511719
| url_ratio <= 0.013333
| | fri_foll2 <= 0.194444
| | | statuses_count <= 5: fake (12.75/2.36)
| | | statuses_count > 5
| | | | friends_count <= 324: human (23.31)
| | | | friends_count > 324
| | | | | friends_count <= 975: fake (10.0/1.0)
| | | | | friends_count > 975: human (4.0)
| | | fri_foll2 > 0.194444: fake (30.96/3.47)
| url_ratio > 0.013333
S1.5. Stringhini et al., reduced error pruning, 3 folds

Tree obtained from the experiment with the Stringhini et al. [8] feature set of Section 5.4.2 with performance reported in the second row of Table 13.

Number of leaves: 16

Size of the tree: 31

Height of the tree: 5

```csharp
fri_foll2 <= 0.493827
  | url_ratio <= 0.013333
  |   | fri_foll2 <= 0.16568
  |   |   | tweet_similarity <= 0
  |   |   |   | statuses_count <= 7: fake (10.17/2.38)
  |   |   |   | statuses_count > 7: human (3.34)
  |   |   | tweet_similarity > 0: human (21.0/7.0)
  |   | fri_foll2 > 0.16568: fake (17.36/1.38)
  | url_ratio > 0.013333
  |   | fri_foll2 <= 0.138013
  |   |   | tweet_similarity <= 0
  |   |   |   | statuses_count <= 1: fake (4.08/0.96)
  |   |   |   | statuses_count > 1: human (48.76/1.32)
  |   |   | tweet_similarity > 0: human (1045.32/12.88)
  |   | fri_foll2 > 0.138013
  |   |   | friends_count <= 419: human (136.54/8.45)
  |   |   | friends_count > 419: fake (20.79)
  |friends_count <= 146
  |   | statuses_count <= 92: fake (1717.47/2.0)
  |   | statuses_count > 92
  |   |   | friends_count <= 382
  |   |   |   | statuses_count <= 247: fake (2.1)
  |   |   |   | statuses_count > 247: human (3.1/0.1)
  |   |   | friends_count > 419: fake (26.11/5.0)
  |friends_count > 146
  |   | statuses_count <= 156: fake (1155.48/1.0)
  |   | statuses_count > 156
  |   |   | friends_count <= 247: fake (2.1)
  |   |   | statuses_count > 247: human (3.1/0.1)
```

```csharp
37
```
S1.6. Stringhini et al., reduced error pruning, 50 folds

Tree obtained from the experiment with the Stringhini et al. [8] feature set of Section 5.4.2 with performance reported in the third row of Table 13.

Number of leaves: 5
Size of the tree: 9
Height of the tree: 4

fri_foll2 <= 0.5092
| url_ratio <= 0.009009: fake (60.41/20.1)
| url_ratio > 0.009009
| fri_foll2 <= 0.145062: human (1627.05/27.76)
| fri_foll2 > 0.145062
| | friends_count <= 366: human (199.65/7.52)
| | friends_count > 366: fake (43.13/8.0)
| fri_foll2 > 0.5092: fake (1891.76/91.47)

S1.7. Yang et al. without the bi-link ratio feature, subtree raising pruning, confidence factor 0.25

Tree obtained from the experiment with the Yang et al. [6] feature set without the bi-link ratio feature, as in Section 5.4.3 with performance reported in the first row of Table 13.

Number of leaves: 51
Size of the tree: 101
Height of the tree: 10

api_tweet_similarity <= 0
| avg_status_of_fri <= 4913.1
| | age <= 279
| | | api_ratio <= 0.025677: fake (13.13/2.0)
| | | api_ratio > 0.025677: human (7.88/1.88)
| | age > 279
| | | following_rate <= 0.377176: human (142.0/1.0)
| | | following_rate > 0.377176
| | | | api_url_ratio <= 0.019231: fake (5.33/0.53)
| | | | api_url_ratio > 0.019231: human (14.67/3.2)
| | avg_status_of_fri > 4913.1
| | avg_foll_of_fri <= 407698
| | | fri_med_foll_of_fri_ratio <= 0.078615
| | | | following_rate <= 0.145098
| | | | | following_rate <= 0.005501
| | | | | age > 1047: human (2.0)
| | | | | following_rate > 0.005501: human (41.0)
| | | | | following_rate > 0.145098
| | | | | | age <= 1372
| | | | | | | api_ratio <= 0.993988: human (15.0)
| | | | | | | following_rate > 0.005501: human (41.0)
| | | | | | following_rate > 0.145098
| | | | | | | age <= 1372
| | | | | | | | age <= 291: fake (4.0)
| | | | | | | | age > 291: human (2.0)
| | | | | | | age > 313: fake (66.0)
| | | | | | | | age <= 313
| | | | | | | | | age <= 145098
| | | | | | | | | | following_rate <= 0.005501
| | | | | | | | | | age <= 1047: human (2.0)
| | | | | | | | | | following_rate > 0.005501: human (41.0)
| | | | | | | | | | following_rate > 0.145098
| | | | | | | | | | | age <= 1372
| | | | | | | | | | | | api_ratio <= 0.993988: human (15.0)
| | | | | | | | | | | | following_rate > 0.005501: human (41.0)
| | | | | | | | | | | | following_rate > 0.145098
| | | | | | | | | | | | | age <= 1372
| | | | | | | | | | | | | | age <= 291: fake (4.0)
| | | | | | | | | | | | | | age > 291: human (2.0)
| | | | | | | | | | | | | | age > 403: fake (40.69)
S1.8. Yang et al. without the bi-link ratio feature, reduced error pruning, 3 folds

Tree obtained from the experiment with the Yang et al. [6] feature set without the bi-link ratio feature, as in Section 5.4.3 with performance reported in the second row of Table 13.

Number of leaves: 53

Size of the tree: 27

Height of the tree: 8
S1.9. Yang et al. without the bi-link ratio feature, reduced error pruning, 50 folds

Tree obtained from the experiment with the Yang et al. [6] feature set without the bi-link ratio feature, as in Section 5.4.3, with performance reported in the third row of Table 13.

Number of leaves: 19

Size of the tree: 37

Height of the tree: 9
S1.10. Class C classifier

Tree obtained from the experiment with all the considered features, as detailed in Section 6.2, with performance reported in the third row of Table 16.

Number of leaves: 16
Size of the tree: 31
Height of the tree: 7

S1.11. Class A classifier

Tree obtained from the experiment with all the Class A features, as detailed in Section 6.2, with performance reported in the third row of Table 16.
Number of leaves: 16
Size of the tree: 31
Height of the tree: 7

fri_foll2 <= 0.511719
  | followers_count <= 4: fake (25.8/3.53)
  | followers_count > 4
    | fri_foll2 <= 0.145062
    |   | has_bio <= 0
    |   |   | friends_count <= 446: human (64.0)
    |   |   | friends_count > 446
    |   |   |   | statuses_count <= 672: fake (9.0/1.0)
    |   |   |   | statuses_count > 672: human (12.0/1.0)
    |   |   | has_bio > 0: human (1590.0/21.0)
    |   | fri_foll2 > 0.145062
    |   |   | friends_count <= 213: human (187.0/2.0)
    |   |   | friends_count > 213
    |   |   |   | following_rate <= 0.387931
    |   |   |   |   | has_bio <= 0: fake (5.0/1.0)
    |   |   |   |   | has_bio > 0: human (13.0/1.0)
    |   |   |   | following_rate > 0.387931
    |   |   |   | statuses_count <= 208: fake (43.0/2.0)
    |   |   |   | statuses_count > 208
    |   |   |   |   | age <= 1131: human (12.0/3.0)
    |   |   |   |   | age > 1131: fake (13.0/1.0)
  | fri_foll2 > 0.511719
    | friends_count <= 146
    |   | age <= 260: fake (59.8/2.0)
    |   | age > 260
    |   |   | nobio_noloc_friends_count_sup_100 <= 0: human (81.45/1.98)
    |   |   | nobio_noloc_friends_count_sup_100 > 0
    |   |   |   | has_bio <= 0: fake (3.49)
    |   |   |   | has_bio > 0: human (3.0)
    |   | friends_count > 146: fake (1778.46/7.0)