# Supervised Classification of Social Spammers using a Similarity-based Markov Random Field Approach

Nour El-Mawass
Normandie Univ, UNIROUEN, LITIS
Rouen, France
nour.el-mawass@etu.univ-rouen.fr

Paul Honeine
Normandie Univ, UNIROUEN, LITIS
Rouen, France
paul.honeine@univ-rouen.fr

Laurent Vercouter
Normandie Univ, INSA Rouen, LITIS
Rouen, France
laurent.vercouter@insa-rouen.fr

## ABSTRACT

Social spam has been plaguing online social networks for years. Being the sites where online users spend most of their time, the battle to capitalize and monetize users' attention is actively fought by both spammers and legitimate sites operators. Social spam detection systems have been proposed as early as 2010. They commonly exploit users' content and behavioral characteristics to build supervised classifiers. Yet spam is an evolving concept, and developed supervised classifiers often become obsolete with the spam community continuously trying to evade detection. In this paper, we use similarity between users to correct evasion-induced errors in the predictions of spam filters. Specifically, we link similar accounts based on their shared applications and build a Markov Random Field model on top of the resulting similarity graph. We use this graphical model in conjunction with traditional supervised classifiers and test the proposed model on a dataset that we recently collected from Twitter. Results show that the proposed model improves the accuracy of classical classifiers by increasing both the precision and the recall of state-of-the-art systems.

## KEYWORDS

Social Spam detection, Online Social Networks, Twitter, Supervised Learning, Markov Random Field, Cybersecurity

## 1 INTRODUCTION

In October 2016, both Disney and Salesforce backed off from an awaited Twitter acquisition. Specialized press reported that the move was partially attributed to Twitter's long-lasting problem of abusive content and trolls [23]. Since its early days, social media has seen a striking proliferation of abusive behavior. The fact that content is user-generated can be as much a curse as a blessing. With 500 million tweets generated daily, monitoring and filtering abusive content is a tedious task. Additionally, automation of content generation and account management via developed applications is very pronounced on Twitter. And while some automated applications adhere to Twitter terms of service, many studies demonstrated that automation is now the main component in a myriad of abusive behavior, including popularity inflation, bot-created opinion manipulation, advertisement, phishing, and malware dissemination.


*MISNC 2018, July 2018, Saint-Etienne, France*

ACM ISBN 978-1-4503-6465-2…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

With the complex and ever-changing nature of abuse on social media, the race to identify and characterize abusers and subsequently quarantine their effects is becoming both more intense and more relevant. As hard a task as it may be, maintaining a virtual environment that is safe, enjoyable and trustworthy is proving to be critical to the image, growth and success of online social networks.

Early attempts at social spam detection were based on an assumption of smoothness. Informally stated, this assumption implies that users having similar characteristics are more likely to have similar labels. Accordingly, early approaches used supervised classification to characterize and detect social spammers. With the continuous evolution of spammers, however, researchers have upped their game by using additional assumptions about the nature and behavior of spammers on social media. These assumptions are generally related to the collective behavior of spammers and aim at exploiting the interdependency introduced by the spam-as-a-service economy [34], e.g. common tools and obfuscation techniques, and the fact that spammers tend to work in groups or communities to ensure coordinated efforts and reach a large audience [6].

Collective detection can be roughly categorized into two main approaches: (1) supervised classification via community-based features, and (2) graph-based detection. The first approach follows the traditional supervised classification model, but uses features defined over a community of similar users instead of using individual features of users [5]. The second form of collective spammers detection follows a graph-based approach [2]. The graph used in these models is usually based on the structure of the social network [37, 38], and the leading assumption is that links between users are based on a relationship of trust, an assumption that has been shown to be questionable on real online social networks including Twitter [17].

The model that we propose in this paper falls under a hybrid methodology that marries the two previously mentioned approaches. The graph-based approach is coupled with machine learning classifiers in a probabilistic graphical model framework. Specifically, we investigate the used applications, a part of the user-content that has been modestly explored in the literature to define similarity between users. We then construct a similarity-based graph over social accounts to supplement local features of supervised classifiers and allow structured prediction over social accounts. Unlike previous hybrid models such as [16], we do not build the graph on the social structure of the network, and thus avoid making assumptions on the prevalence of attack links between a spammers region and a legitimate region in the network. We equally avoid the pitfalls of models based on collective features by preserving an individual classification model over each account, thus maintaining the ability to capture individual nuances of spammers.

The content and behavioral features we use in this work are reproduced from state-of-the-art systems described in the literature [3, 21, 24, 30]. To evaluate the performance of these features and compare it to the performance of the system we propose, we collect and label a dataset from Twitter. Results demonstrate the effectiveness of our system in mitigating the effect of evasion techniques on social spam classifiers. Quantitatively, our system improves prediction accuracy and effectively increases the recall and precision of state-of-the-art classifiers.

## 2 BACKGROUND AND RELATED WORK

### 2.1 Spam on Twitter and Social Media

In 2009, Yardi et al. were the first to explore spam on Twitter [36]. They describe accounts posting unwanted URLs to a trending Twitter topic (*#robotpickuplines*). The following year, several attempts were made to formally address the rising problem of spam on Twitter [3, 8, 21, 30]. These early approaches were generally based on a supervised classification framework and used features extracted from users profiles, content, behavior and social network to characterize and identify spammers. As is expected of a situation where the opponent is adversarial, the parameters of the problem kept changing along the years. Subsequent research addressed new and more complex aspects of abuse on Twitter including popularity inflation, fake followers [9, 31], Sybil accounts, bot-created opinion manipulation and political propaganda [32], trends poisoning [20], advertisement [39], phishing and malware dissemination [1, 7], and even random acts of sabotage. Evidence of spam evolution can be traced back to the work of Yang et al. [35] and several recent studies have independently reached the same result [11, 13].

Along with detection systems, there was therefore a need to deeply understand the mechanisms that controlled the underground of malicious and abusive behavior on social media. The work of Thomas et al. [33, 34] and Stringhini et al. [29, 31] on the spam underground and communities are notable in this domain.

### 2.2 Collective Spam Detection on Social Media

The previously mentioned work lead the way to a new paradigm of spam detection based on collective evidence. Two distinct approaches fall under the hood of this new trend.

- Community-based classification via supervised classifiers. This approach follows the mainstream supervised classification methodology but uses community-based features [5].
- Graph-based detection. This approach uses graph theoretical techniques to detect communities on a social graph. The graph can represent the social structure of the network [37, 38] or it can be alternatively based on a measure of similarity or abnormally synchronous actions between users [4, 19].

There exists a third hybrid approach that generally uses a probabilistic graphical model to incorporate information from both the social graph and users labels or statistical features. SybilBelief [18] is a system that propagates known labels of users over the social structure using a Markov Random Field (MRF) model. The system is tested on synthetic and real-world social graphs including the social graph of Facebook. SybilFrame [16] is based on a similar idea but uses weak local classifiers instead of known labels and is evaluated on synthetic data and the social structure of Twitter. Our work differs from these works in that it totally avoids using the social structure of the network, and chooses instead to base the graph on the similarity between users applications, thus avoiding the notion of strong-trust that is assumed in structure-based contributions. Additionally, by constructing the graph with content-based similarity, our work is more easily reproducible than the one using proprietary social graph information. Moreover, SybilBelief and SybilFrame are compared to graph-based approaches and are reported to significantly improve the performance of these approaches in realistic settings by allowing the propagation of local information. Unlike these two systems, we compare our system to local classifiers that have been shown their merit empirically. We specifically endeavor to select and reproduce the most relevant features proposed in the literature, and thus believe that the empirical evaluation of our system offers unique evidence on the effectiveness of structured classification as a paradigm for detecting evolved social spammers.

## 3 PROPOSED SYSTEM

### 3.1 General Overview

In classical classification problems, the goal is to find a model that maps an input space, e.g. a set of features, to the set of possible labels. The problem we define here is similar: we want to construct a model that labels each social account as either a spammer or a legitimate account. The system we propose has two phases:

(1) The learning phase where we learn the model parameters over a groundtruth dataset of labeled users, and
(2) The deployment phase where the model is evaluated.

In both phases, we start by crawling content information of Twitter accounts. We then use the profile information and the crawled content in two separate ways:

(1) We extract a features vector $f$ that codes the distinguishing behavioral, social and content-based characteristics of the account (e.g. age, number of followers, proportion of replies in a user's posts).
(2) For each account, we extract the set of "applications" used by the account to post its tweets (e.g. Twitter for iPhone, TweetDeck). From the percentage of tweets posted by each app, we then compute the similarity between pairs of accounts. Finally, we use the resulting similarity measures to construct a graph $G(V, E)$ over users.

In the learning phase, the features vectors along with the true labels of users are used to train a supervised classification model. The graph on the other hand, is used as a structure for the MRF built over the accounts. MRF is a probabilistic graphical model that we use to tie labels of similar accounts. Its parameters are estimated by maximizing the likelihood of obtaining the labels of the groundtruth dataset. In the deployment phase, we link the two parts of the system together. We use the supervised classifier to output beliefs about an account's class based on its features. These beliefs are subsequently used as node priors in the MRF. We apply joint optimization using Loopy Belief Propagation over the MRF to get the most probable configuration of labels. Figure 1 summarizes the data flow and general architecture of the proposed system.

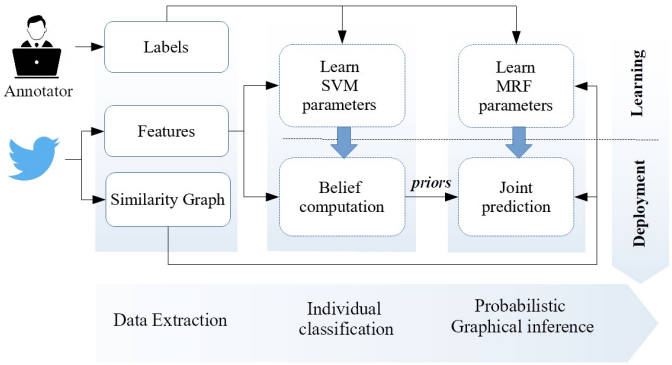


**Figure 1: General architecture of the proposed system.**

## 3.2 Supervised Classification

The social spam detection problem is often defined as a supervised classification problem. The goal is therefore to find a function $g : X \rightarrow Y$ that maps the input space $X$ (defined by features vectors) to the output space $Y$ (set of possible labels of an account: either legitimate or spammer).

For each user $u$, with features vector denoted $x_u$, the classifier predicts a label $y$ with a probability $p(y|x_u)$ These probabilities quantify the classifier confidence of its prediction and we consider them our prior belief of a user class.

To define and build a supervised classifier over our groundtruth dataset, we have to make two choices regarding the design of the classifier: selecting the set of features that will represent each user, and choosing the statistical model that will be used to represent and learn the classification function (a mapping from the features space to the space of labels).

*3.2.1 Choosing a suitable set of statistical features.* The goal of this work is not to propose a new set of features that is adapted to the task of detecting contemporary social spammers. Rather, we would like to simulate a set of relevant features and show that their performance on current Twitter's spammers population can be improved using our proposed model. For that, we opt for two strategies:

(1) First, we reproduce the work in [3] and [30] (denoted hereafter as *Benevenuto* and *Stringhini* respectively). Namely, we extract the sets of features proposed in these two works, and use them to train and evaluate classifiers over the groundtruth dataset. A list of these features along with their description is presented in Table 1. Note that we have chosen these two models based on self-reported performance, wide acceptance in the community, and reproducibility. The latter is defined by the possibility of reproducing the model with accessible account information and without the need for internal information such as IP addresses or the social graph[1].

[1]While it is theoretically possible to use Twitter's Rest API to obtain a user's social graph, the imposed API rate limit makes it prohibitive and impractical to require this information in a large-scale model. Models using such information (e.g. [35]) are hard to reproduce with a normal-level data access.

(2) Second, we carefully select 28 features from different previous works [3, 21, 24, 30] and compute their values for accounts in our dataset. Table 1 shows the list of selected features. This set captures a wide range of information including aspects related to the accounts behavior, social network, content and age. As will be shown in the results section, the extensiveness and domain relevance of this set of features results in a classification performance that is better than either sets of Stringhini or Benevenuto.

*3.2.2 Choosing a suitable classification model.* To choose a suitable classification model, we are constrained by two criteria:

- The model should be able to output not only a label $y_i$ for each configuration of features $x_i$, but also a probability $p(y_i|x_i)$ for each predicted label. These prediction probabilities are used as priors in our MRF model.
- In order to get a representative dataset, we include a variety of social profiles for both legitimate users and spammers. The methods used to collect labeled instances introduce a selection bias that represents a common problem across the literature [14]. To account for this characteristic of the collected dataset and to avoid bogus results, we are not to use generative learning models (e.g. Naive Bayes). These methods learn a joint distribution $p(x, y)$ over the input and output spaces and are therefore unsuitable for our purposes. Alternatively, since any groundtruth dataset does not offer a true distribution $p(x)$ over the input space, we only want to learn the conditional probability $p(y|x)$, and a discriminative learning model is more adapted to the task.

The Support Vector Machines (SVM) model performs classification by establishing a boundary of selected input points known as support vectors. Since it offers both required characteristics, we will use it in the remaining analysis as our baseline supervised classifier, without any loss of generality.

## 3.3 Constructing the Similarity Graph

We construct a graph where nodes are users and links represent similarity between users. We base our definition of similarity on the observation that accounts that use the same applications tend to belong to the same class. Specifically, spammers belonging to the same or similar spam campaigns tend to have similar applications usage profiles.

There are two prominent characteristics of applications on Twitter: The number of applications on Twitter is high (we counted $71k$ unique applications in our dataset) and the association between users and applications is dynamic. Encoding applications as a features vector results in a static, sparse, and high-dimensional representation that is not adapted to supervised classifiers. Instead, the concept of similarity offers an elegant and compact way to represent and exploit this vital characteristic of a user's profile.

To construct the similarity graph, we first crawl the most recent tweets posted by each user in the dataset and extract the applications used to post each of these tweets (e.g. Twitter for iPhone, TweetDeck). We then compute the proportion of tweets posted by each app. For example, if a user $u$ has 5 tweets posted by the following respective applications $(a_1, a_2, a_3, a_2, a_1)$, the normalized form of this user's applications profile is $\{a_1 : 2/5, a_2 : 2/5, a_3 : 1/5\}$.

**Table 1: Description of features used in this work**

| | Feature | Description | Our features | Benevenuto | Stringhini |
|---|---|---|---|---|---|
| Profile | age of the account | time since the account was created | ✓ | ✓ | |
| | statuses count | Total umber of tweets posted by the account | ✓ | | ✓ |
| S. Network | followers count | Number of users following the account | ✓ | ✓ | |
| | friends count | Number of users followed by the account | ✓ | ✓ | ✓ |
| | followers per followees | Ratio of followers to followees (friends) | ✓ | ✓ | |
| | followees per squared followers | Ratio of friends per the squared number of followers | | | ✓ |
| Content | replicates | Number of identical posts in the account timeline | ✓ | | |
| | similarity | Average similarity between the account's posts | ✓ | | ✓ |
| | fraction of tweets with urls | Fraction of tweets containing a link | ✓ | ✓ | ✓ |
| | fraction of tweets with hashtags | Fraction of tweets containing a hashtag | ✓ | | |
| | fraction of replies | Fraction of replies in the user's posts | ✓ | ✓ | |
| | fraction of retweets | Fraction of retweets in the user's posts | ✓ | | |
| | mean nb hashtags per tweet | Average number of hashtags in a tweet | ✓ | ✓ | |
| | mean nb urls per tweet | Average number of links in a tweet | ✓ | ✓ | |
| | urls used on average | Average number of times a single link is used | ✓ | | |
| Behavioral | avg intertweet interval | Average time interval between consecutive tweets | ✓ | | |
| | std intertweet interval | Standard deviation of time intervals between consecutive tweets | ✓ | | |
| | min intertweet interval | Minimum time interval between two consecutive tweets | ✓ | | |
| | nb followees per day | Average number of followees the account follows in a day | ✓ | | |
| | nb followers per day | Average number of users following the account in a day | ✓ | | |
| | active tweeting frequency per day | Average number of tweets posted by the account on a daily basis (computed based on the 200 most recent tweets) | ✓ | | |
| | distribution of tweets in temporal bins | 8 features corresponding to the proportion of the account's tweets contained in temporal bins of 3 hours each. | ✓ | | |

For $n$ unique applications, the normalized vector notation representing the applications profile of the previous example is a $n \times 1$ sparse vector of the form:

$$A_u = \left[0.4, 0.4, 0.2, \underbrace{0, \ldots, 0}_{(n\text{-}3) \text{ zeros}}\right]^T. \quad (1)$$

To obtain the similarity between two users $u$ and $v$ with normalized applications vectors $A_u$ and $A_v$, we compute the cosine similarity measure (normalized inner product of the two vectors) which outputs a value between 0 and 1:

$$Sim(u, v) = \cos(A_u, A_v) = \frac{A_u^T A_v}{\|A_u\| \, \|A_v\|} \quad (2)$$

Figure 2 illustrates the computation of similarity on a toy example.

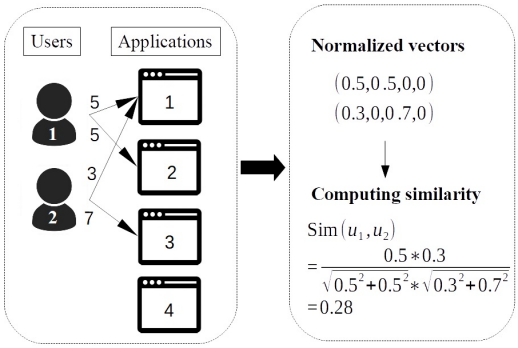


**Figure 2: A toy example showing the computation of similarity between two users based on their applications profile.**

## 3.4 Markov Random Field

A MRF is a probabilistic graphical model that allows joint inference over dependent random variables. It consists of a graph $G(V, E)$ where nodes are random variables and edges denote a dependency between two random variables. The Markov assumption in a MRF states that a node is independent from its non-neighboring nodes given its neighbors. Formally, given two variables $y_i$ and $y_j$ and the neighboring nodes $N_{y_i}$ of $y_i$, the following equation holds:

$$p(y_i|y_j, N_{y_i}) = p(y_i|N_{y_i}) \quad (3)$$

A set $\Psi$ of potential functions govern the relationships between random variables. Potentials are factors defined over cliques of nodes. In this work, we propose to use the pairwise MRF model (p-MRF), and define two types of potentials over nodes: edge (or pairwise) potentials defined over two connected nodes and node (or unary) potentials defined over individual nodes. Together, these potentials ensure that the model responds to the smoothness criteria between connected labels $y$ (pairwise potentials) while penalizing discrepancy between the observations $x$ and their corresponding labels in $y$ (unary potentials). They are constructed as follows:

(1) A unary potential $\phi_u$ quantifies how favorable a label $y_i$ is for node $Y_i$. We define it as a function that for each user $u \in U$ and label $y \in L^U$ associates a probability $p(y_i)$:

$$\phi_u : L^U \rightarrow [0, 1].$$

These probabilities are fixed priors, inferred from the previously built supervised classification model, and so the features information is indirectly incorporated into these priors as follows:

$$\phi_u(y_u) = \begin{cases} p_u & \text{if } y_u = 0 \\ 1 - p_u & \text{if } y_u = 1 \end{cases} \quad \text{where} \quad p_u \in [0, 1] \tag{4}$$

(2) An edge connects two nodes, $Y_u$ and $Y_v$, if the corresponding users $u$ and $v$ are similar, and is associated with a pairwise potential $\phi_{u,v}(Y_u, Y_v)$. In the current context, an edge potential defines correlation between labels, i.e., the relative likelihood that two nodes are labeled similarly. Formally, edge potentials are defined as functions that for every realization of a tuple of labels (in $L^U$) associates a real-valued factor quantifying its likelihood:

$$\phi_{u,v} : L^U \times L^U \rightarrow \mathbb{R}^+.$$

Specifically, we define the edge potentials as follows:

$$\phi_{u,v}(y_u, y_v) = \exp(f(y_u, y_v)) \tag{5}$$

$$f(y_u, y_v) = \begin{cases} w_0 & \text{if } y_u = y_v = 0 \\ w_1 & \text{if } y_u = y_v = 1 \\ 0 & \text{otherwise} \end{cases} \quad \text{where} \quad w_0, w_1 \in \mathbb{R}^+ \tag{6}$$

**Optimization Goal:** The goal is to maximize the probability of a joint configuration of labels $P(Y|\Theta)$ by optimizing the product of potentials defined over all nodes $v \in V$ and all edges $(u, v) \in E$:

$$P(Y|\Theta) = \frac{1}{Z}\widetilde{P}(Y|\Theta), \tag{7}$$

where $Z = \sum_Y \widetilde{P}(Y|\Theta)$ is the partition function and

$$\widetilde{P}(Y|\Theta) = \prod_{v \in V} \phi_v(Y_v) \prod_{(u,v) \in E} \phi_{(u,v)}(Y_u, Y_v). \tag{8}$$

This problem is usually solved as a non-convex multivariate optimization problem. The goal is to minimize the negative log likelihood value of the above equation over several iterations using e.g. a quasi-Newton method. We use the Loopy Belief Propagation (LBP) algorithm to estimate beliefs and infer the most likely configuration at each iteration. LBP is essentially an iterative message-passing algorithm with a time complexity that is linear in the number of edges. In graphs that contain loops, LBP is an approximate algorithm and is not guaranteed to converge. In practice, however, convergence is often reached after few iterations.

## 4 DATA COLLECTION AND LABELING

We collected data from Twitter, the well-known Online Social Network during the period between 5 and 21 October 2017. We used the Developer *Streaming API*[2] to get a random sample of $20M$ tweets from $12M$ active users, and the *Rest API*[2] to crawl tweets of selected users in the sample.

We also built a groundtruth dataset of labeled Twitter users. This dataset contained 767 users divided over four categories of users: verified accounts, normal users, hashtag hijackers and promoters.

[2]Twitter developers API https://developer.twitter.com/en/docs

**Table 2: Characteristics of the groundtruth dataset**

| Group Designation | Class | Users | Tweets |
|---|---|---|---|
| Verified Users | Legitimate | 500 | 100 108 |
| Human Users | Legitimate | 134 | 59 277 |
| Trends Hijackers | Spammer | 47 | 19 972 |
| Promotional Spambots | Spammer | 86 | 31 404 |
| Total | | 767 | 210 761 |

The first two categories were legitimate accounts and formed 83% of the dataset while the other two categories formed the remaining 17% and exhibited an abusive behavior that violates Twitter terms of service[3]. Table 2 summarizes the general characteristics of the groundtruth dataset. For each of these users, we extracted profile information and we used Twitter's Rest API in order to crawl at least the most recent 200 tweets of each user (or all the tweets if the user's timeline contains less than 200 tweets). Users profiles and tweets were subsequently used to extract relevant content and behavioral features.

### 4.1 Verified Accounts

Given the complex nature of accounts automation on Twitter, we found it important that the dataset comprised automated users on both ends of the spectrum, that is, automated profiles from both legitimate and abusive users. Unlike some recent work such as [25], we did not exclude verified users from the dataset. Verified users often belonged to companies, celebrities or public figures, and were often operated by dedicated or generic content management applications[4]. These accounts may also exhibit a mixed human-bot behavior where real persons use the verified account to interact with its followers. This type of behavior is typical of what has come to be known in the literature as a "cyborg" account. We acknowledge therefore that the characteristics of verified accounts are very different from those of normal human-based accounts. We chose to include these accounts in the dataset to prevent the classifier from learning that every automated behavior is abusive.

Note that these users are easy to identify (their profiles are marked with a blue tick mark and their crawled profiles include a "verified" flag). We randomly selected 500 users among 43k verified users appearing in the dataset and we included these 500 users in the groundtruth dataset.

### 4.2 Human users

The remaining 134 legitimate users in the groundtruth dataset were normal human-operated accounts. These users were identified by manually investigating a sample of active accounts from our initial dataset. Manual investigation required a careful examination of the account in question, its tweets, profile and behavioral characteristics, and has therefore a small throughput. We elaborated on the pitfalls and advantages of manual labeling in the next paragraphs.

[3]Twitter terms of service https://twitter.com/en/tos

[4]Examples of generic content management applications include TweetDeck and dlvr.


**Figure 3: A screenshot of a compromised verified account posting a tweet containing a phishing link.**

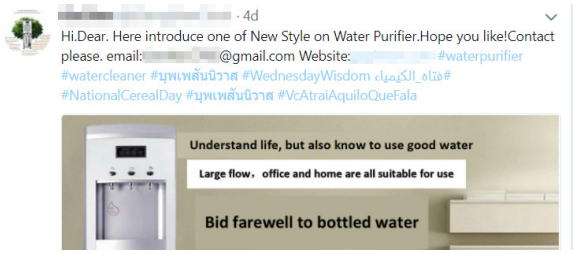


**Figure 4: An example of trend-hijacking spam on Twitter.**

## 4.3 Promoters

The blacklisted links heuristic is a well-known heuristic that is commonly used to identify spammers in email and social media [1, 22]. It consists of identifying users that posted links to malicious webpages by verifying links appearing on social media against a continuously updated database of malicious webpages such as Google Safe Browsing[5] and Phishtank[6].

We tried to directly apply this heuristic to our crawled dataset. For this, we first started by extracting all $3.8M$ links in the crawled tweets. We subsequently wrote a program that follows the redirection chain of each link and returns the final landing webpage. We then used Google Safe Browsing API to identify suspicious URLs. Only 156 URLs were identified as being phishing or malware URLs. We extracted all users IDs in our dataset that posted any of these malicious URLs and then proceeded to the manual verification of the resulting accounts. Surprisingly, a significant number of these accounts were actually legitimate accounts that were temporarily compromised[7] by a malicious posting mechanism[8]. Consequently, we could not rely on this labeling heuristic alone to obtain malicious accounts as it yielded a high false negative rate. Alternatively, for the users that were found to be genuinely malicious, we extracted the text associated with the blacklisted URLs. We then searched Twitter for users that posted the same text, and were able to identify several communities of spammers. We obtained 86 users in total, most of them engaged in promotional and site referral activity.

## 4.4 Trends hijackers

Trend hijacking is a type of spam that is particularly ubiquitous on social media. Trending topics and hashtags offer a high visibility and attract a large audience. Spam that targets trending topics is known as collective-attention spamming [20] because it tries to gain a higher visibility for abusive content by targeting topics and accounts that are popular on the attacked social network. This type of spam consists of poisoning the trending topic with unrelated posts, often to promote a particular product or service (see Figure 4 for an example). And while this particular instance of behavior is easy to spot, even for the untrained observer, there exist other types of trend poisoning that are difficult to identify. This occurs when the posts content is semantically aligned with the attacked topic, which is often the case when the goal is not direct promotion, but opinion manipulation and political propaganda [27, 32].

We used a dataset of trends hijackers that we presented in a previous work [13]. This dataset was obtained by reading the tweets of a trending sport-related hashtag and manually identifying suspect tweets. This was followed by a manual investigation consisting of reading the recent tweets of suspect profiles and cross-examining different profiles for similar patterns and content. This process is similar to what Cresci et al. describe in their paper [10]. Manual labeling is different from mainstream labeling techniques described in the literature in that it is time consuming and requires an annotator that is familiar with current spam techniques and tricks[9].

Since trends hijackers are particularly aggressive in their spamming activity, a substantial percentage of malicious accounts in the original dataset was suspended by the time we conducted the current analysis. Since we needed up-to-date evidence on spam for our results to be credible, we could not rely on information crawled from the suspended accounts two years ago. Hence, we used here the recently collected tweets from the remaining 47 accounts.

# 5 EXPERIMENTAL EVALUATION

## 5.1 Experimental setup and evaluation

We split the dataset into a training and a test datasets with a 70/30 ratio. We used the *sickit-learn* library [26] in Python to learn the parameters of the SVM classifier. The SVM classifier used the RBF kernel, and its parameters $C$ and $\gamma$ were obtained using a grid search. All features were normalized before training.

We chose a similarity threshold of 0.9 to construct the similarity graph and dropped links with lower similarity levels. This resulted in a graph with $32k$ edges, or about 5% of the number of edges in a fully connected graph with 767 nodes. We implemented the MRF using the *UGM* library [28] in Matlab. For inference and parameters estimation, we used the library's implementation of LBP and the multivariate functions optimization method *minfunc*, respectively.

We compared our model with the SVM classifier over the three previously discussed sets of features, namely our selected set of state-of-the-art features and the sets of features proposed in Benevenuto and Stringhini. We used information retrieval metrics to assess the performance of compared classifiers.

[5]Google Safe Browsing API: https://developers.google.com/safe-browsing/
[6]The Phishtank database https://www.phishtank.com/
[7]Compromise is fairly common on social media. Compa [12] is a system that builds statistical profiles for users and identifies compromise by comparing recent posts with the previous profile. We roughly used the system description as a guideline for identifying and excluding compromised accounts from our dataset.
[8]In one instance of these compromise campaigns, the "Rayban sale" scam, one verified account was found to retweet the same malicious URL dozens of times before the malicious behavior stops and the accounts restarts its normal behavior (see Figure 3).
[9]Previous work that used manual labeling such as [3] rely on crowdsourced annotation of individual hashtag tweets. While we think that this method could have yielded trustworthy annotation back when spam was less complicated and more straightforward, recent empirical evidence [11, 15] suggests that non-initiated human annotators fail to identify the new generation of spam on social media.

**Table 3: Classification results of SVM and our model on three different sets of features**

| | | **Our features** | | **Benevenuto features** | | **Stringhini features** | |
|---|---|---|---|---|---|---|---|
| | | Legitimate | Sybil | Legitimate | Sybil | Legitimate | Sybil |
| SVM | Precision | 0.947 | 0.795 | 0.902 | 0.703 | 0.843 | 0.733 |
| | Recall | 0.952 | 0.778 | 0.941 | 0.578 | 0.978 | 0.244 |
| | F-measure | 0.949 | 0.787 | 0.921 | 0.634 | 0.905 | 0.367 |
| | Accuracy | 0.918 | | 0.87 | | 0.835 | |
| SVM + MRF (this paper) | Precision | **0.968** | **0.878** | **0.91** | **0.774** | **0.877** | **0.8** |
| | Recall | **0.974** | **0.857** | **0.963** | 0.571 | **0.979** | **0.381** |
| | F-measure | **0.971** | **0.867** | **0.936** | **0.658** | **0.925** | **0.516** |
| | Accuracy | **0.952** | | **0.892** | | **0.87** | |

Results presented in Table 3 show a general improvement in the precision of classification for all sets of features considered for evaluation. There is also a strong tendency toward improving the recall of supervised classifiers.

## 5.2 Discussion

Results show that the prediction accuracy of traditional classifiers has indeed deteriorated (compared to the performance reported on datasets used in the original articles). This can arguably be explained by the detrimental effect of spam evolution on the predictive power of proposed features. In particular, the specific sets of features used by Benevenuto [3] and Stringhini [30] seem to be insufficient to effectively identify contemporary spammers (even with the SVM classifier being trained over recently collected data). The set of selected state-of-the-art features, on the other hand, yields an acceptable performance, a fact that encourages further contributions that incorporates this set of features as an active building block.

A notable observation that concerns the characteristics of the resulting similarity graph is that the graphical inference phase only alters beliefs on connected nodes. In the case of singleton nodes, that is nodes that are not connected to any other nodes, the system is equivalent to an SVM classifier. In the setting of our experimental setup, 75 nodes (or 10% of users) are singleton. While isolation is expected for profiles that use dedicated apps (such as some profiles that represent celebrities or companies), singleton spammers are perhaps an artifact of the limited sample size. Since spammers often follow a botnet-like model with a central entity controlling them, a similar application profile is to be expected and groups of connected spammers should be the norm. Further empirical evaluation on our large dataset is needed to confirm these assumptions.

A practical assumption of structured classification is that nodes degrees generally have the same order of magnitude. This turns out not to be the case on our Twitter-based similarity graph, where a small set of applications capitalizes most of the users. These are applications that are mostly used by human users (such as Twitter for Android and the web interface). This means that these applications figure in the profile of a sizable portion of users, resulting in the creation of similarity links between many of them. While the effect of this are manageable on a small-scale experiment, the number of connections grows exponentially with the number of users which can slow down LBP execution.

Additionally, most of the legitimate activity bundles together in a single cluster spanning various types of human-based activity profiles. This makes it easy for spammers to infiltrate the legitimate cluster by creating similarity links with users on the edges (e.g. imitating the profiles of legitimate cyborgs and using legitimate generic content management applications). A solution would be to pre-process users to filter out those that, by solely using human-based applications are likely to be human. For these users, a one-phase supervised classifier is more effective since MRF belief updating will inevitably mark them as legitimate.

Finally, results confirm that similarity can be used to improve the performance of a weak local classifier. The local information synthesized as a belief can be propagated throughout the graph to correct misclassified instances and mitigate the effect of spam evolution. Compared to local classifiers, our model consistently improves precision over several sets of features, and generally improves recall as well. The notion of a weak local classifier, however, is certainly a good candidate for exploration. Notably, the local classifier based on features from Stringhini, which seems to hold little classification power, was significantly improved by the introduction of joint prediction. Interestingly, the joint prediction based on Benevenuto features has a mixed result (a better precision and a slightly worse recall for spammers detection) which comes at odds with the apparently better performance of the local classifier (compared to that based on Stringhini's features). This suggests that the performance of a local classifier, measured in terms of precision and recall, is not enough to judge its effectiveness as a local belief estimator, and that the particular structure of the similarity graph may have an equally important role.

# 6 CONCLUSION AND FUTURE WORK

In this paper, we presented a structured classification approach for spammers detection on online social networks. The proposed system leverages similarity between users to propagate beliefs about their labels. We initiated beliefs using supervised classifiers trained with selected state-of-the-art features. We showed that optimizing the prediction over a Markov Random Field permits to correct misclassified labels, thus improving the performance of baseline supervised classifiers. This not only allows the detection system to be more sustainable but it could also be used to design adaptive classifiers.

The promising results of this study show that similarity can be leveraged for increased detection accuracy. And future work is likely to extend the evaluation by tackling aspects of similarity and design choices not discussed in this work. We especially plan to explore the effect of the local classifiers and the graph structure on successful belief propagation and to evaluate the system with other classification models (e.g. logistic regression).

## REFERENCES


[1] Anupama Aggarwal, Ashwin Rajadesingan, and Ponnurangam Kumaraguru. 2012. PhishAri: Automatic realtime phishing detection on twitter. In *eCrime Researchers Summit (eCrime), 2012*. IEEE, 1–12.

[2] Faraz Ahmed and Muhammad Abulaish. 2012. An MCL-based approach for spam profile detection in online social networks. *Proc. of the 11th IEEE Int. Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012 - 11th IEEE Int. Conference on Ubiquitous Computing and Communications, IUCC-2012* (2012), 602–608. https://doi.org/10.1109/TrustCom.2012.83

[3] Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. 2010. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, Vol. 6. 12.

[4] Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, and Christos Faloutsos. 2013. CopyCatch: stopping group attacks by spotting lockstep behavior in social networks. In *Proceedings of the 22nd international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 119–130.

[5] Sajid Yousuf Bhat and Muhammad Abulaish. 2013. Community-based features for identifying spammers in online social networks. *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining - ASONAM '13* (2013), 100–107. https://doi.org/10.1145/2492517.2492567

[6] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. 2014. Uncovering Large Groups of Active Malicious Accounts in Online Social Networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14*. ACM Press, New York, New York, USA, 477–488. https://doi.org/10.1145/2660267.2660269

[7] Sidharth Chhabra, Anupama Aggarwal, Fabricio Benevenuto, and Ponnurangam Kumaraguru. 2011. Phi.sh/$oCiaL: The Phishing Landscape through Short URLs. In *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference on - CEAS '11*. ACM Press, New York, New York, USA, 92–101. https://doi.org/10.1145/2030376.2030387

[8] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. 2010. Who is tweeting on Twitter: human, bot, or cyborg?. In *Proceedings of the 26th annual computer security applications conference*. ACM, 21–30.

[9] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2015. Fame for sale: Efficient detection of fake Twitter followers. *arXiv preprint* 80, July 2012 (2015), 1–34. http://arxiv.org/abs/1509.04098

[10] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2016. DNA-inspired online behavioral modeling and its application to spambot detection. *IEEE Intelligent Systems* 31, 5 (2016), 58–64.

[11] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2017. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. *arXiv preprint arXiv:1701.03017* (2017).

[12] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2013. COMPA: Detecting Compromised Accounts on Social Networks.. In *NDSS*. https://doi.org/10.1.1.363.6606 arXiv:1509.03531

[13] Nour El-Mawass and Saad Alaboodi. 2016. Detecting Arabic Spammers and Content Polluters on Twitter. In *6th International Conference on Digital Information Processing and Communications (ICDIPC'16)*. IEEE, Beirut, Lebanon.

[14] Nour El-Mawass and Saad Alaboodi. 2017. Data Quality Challenges in Social Spam Research. *ACM Journal on Data and Information Qulaity* (2017).

[15] David Mandell Freeman. 2017. Can You Spot the Fakes?: On the Limitations of User Feedback in Online Social Networks. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1093–1102.

[16] Peng Gao, Neil Zhenqiang Gong, Sanjeev Kulkarni, Kurt Thomas, and Prateek Mittal. 2015. Sybilframe: A defense-in-depth framework for structure-based sybil detection. *arXiv preprint arXiv:1503.02985* (2015), 17. arXiv:1503.02985 http://arxiv.org/abs/1503.02985

[17] Saptarshi Ghosh, Bimal Viswanath, Farshad Kooti, Naveen Kumar Sharma, Gautam Korlam, Fabricio Benevenuto, Niloy Ganguly, and Krishna Phani Gummadi. 2012. Understanding and combating link farming in the twitter social network. In *Proceedings of the 21st international conference on World Wide Web - WWW '12*. ACM Press, New York, New York, USA, 61. https://doi.org/10.1145/2187836.2187846

[18] Neil Zhenqiang Gong, Mario Frank, and Prateek Mittal. 2014. SybilBelief: A Semi-Supervised Learning Approach for Structure-Based Sybil Detection. *IEEE Transactions on Information Forensics and Security* 9, 6 (jun 2014), 976–987. https://doi.org/10.1109/TIFS.2014.2316975

[19] Meng Jiang, Bryan Hooi, Alex Beutel, Shiqiang Yang, Peng Cui, and Christos Faloutsos. 2015. A general suspiciousness metric for dense blocks in multimodal data. In *Proceedings of IEEE international conference on data mining. IEEE*.

[20] Kyumin Lee, James Caverlee, Krishna Y. Kamath, and Zhiyuan Cheng. 2012. Detecting collective attention spam. In *Proceedings of the 2nd Joint WICOW/AIRWeb Workshop on Web Quality - WebQuality '12*. ACM Press, New York, New York, USA, 48. https://doi.org/10.1145/2184305.2184316

[21] Kyumin Lee, James Caverlee, and Steve Webb. 2010. Uncovering social spammers: social honeypots+ machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 435–442.

[22] Sangho Lee and Jong Kim. 2012. WarningBird: Detecting Suspicious URLs in Twitter Stream.. In *NDSS*.

[23] Mathew Ingram. 2016. Disney, Salesforce Dropped Twitter Bids Because of Trolls | Fortune. (2016). http://fortune.com/2016/10/18/twitter-disney-salesforce/

[24] M. McCord and M. Chuah. 2011. Spam detection on twitter using traditional classifiers. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6906 LNCS (2011), 175–186. https://doi.org/10.1007/978-3-642-23496-5_13

[25] Fred Morstatter, Liang Wu, Tahora H Nazer, Kathleen M Carley, and Huan Liu. 2016. A new approach to bot detection: striking the balance between precision and recall. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*. IEEE, 533–540.

[26] F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[27] Jacob Ratkiewicz, Michael D Conover, Mark Meiss, B Gonc, Alessandro Flammini, Filippo Menczer, Bruno Gonçalves, Alessandro Flammini, and Filippo Menczer. 2011. Detecting and Tracking Political Abuse in Social Media.. In *ICWSM*. 297–304. arXiv:1011.3768 http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/viewFile/2850/3274

[28] M. Schmidt. 2007. UGM: A Matlab toolbox for probabilistic undirected graphical models. (2007). http://www.cs.ubc.ca/{~}schmidtm/Software/UGM.html

[29] Gianluca Stringhini, Manuel Egele, Christopher Kruegel, and Giovanni Vigna. 2012. Poultry markets: on the underground economy of twitter followers. In *Proceedings of WOSN'12*. 1–6. https://doi.org/10.1145/2377677.2377781

[30] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2010. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*. ACM, 1–9.

[31] Gianluca Stringhini, Gang Wang, Manuel Egele, Christopher Kruegel, Giovanni Vigna, Haitao Zheng, and Ben Y Zhao. 2013. Follow the green: growth and dynamics in twitter follower markets. In *Proceedings of the 2013 conference on Internet measurement conference*. 163–176. https://doi.org/10.1145/2504730.2504731

[32] Kurt Thomas, Chris Grier, and Vern Paxson. 2012. Adapting Social Spam Infrastructure for Political Censorship. In *5th USENIX Workshop on Large-Scale Exploits and Emergent Threats*. https://www.usenix.org/conference/leet12/workshop-program/presentation/thomas

[33] Kurt Thomas, Chris Grier, Dawn Song, and Vern Paxson. 2011. Suspended accounts in retrospect: an analysis of twitter spam. *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference* (2011), 243–258. https://doi.org/10.1145/2068816.2068840

[34] Kurt Thomas, Vern Paxson, Damon Mccoy, and Chris Grier. 2013. Trafficking Fraudulent Accounts : The Role of the Underground Market in Twitter Spam and Abuse. *USENIX Security Symposium* (2013), 195–210.

[35] Chao Yang, Robert Chandler Harkreader, and Guofei Gu. 2011. Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. In *Recent Advances in Intrusion Detection*. Springer, 318–337.

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

[37] Haifeng Yu, Phillip B. Gibbons, Michael Kaminsky, and Feng Xiao. 2008. SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE, 3–17. https://doi.org/10.1109/SP.2008.13

[38] Haifeng Yu, Michael Kaminsky, Philip B. Gibbons, and Abraham D. Flaxman. 2008. SybilGuard: Defending against sybil attacks via social networks. *IEEE/ACM Transactions on Networking* 16, 3 (2008), 576–589. https://doi.org/10.1109/TNET.2008.923723

[39] Yubao Zhang, Xin Ruan, Haining Wang, and Hui Wang. 2014. What scale of audience a campaign can reach in what price on Twitter? *INFOCOM, 2014 Proceedings IEEE* (2014), 1168–1176.