Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 112 (2017) 833–843

Procedia Computer Science

www.elsevier.com/locate/procedia

21st International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2017, 6-8 September 2017, Marseille, France

A Topic-Based Hidden Markov Model for Real-Time Spam Tweets Filtering

Mahdi Washha[a,*], Aziz Qaroush[b], Manel Mezghani[a], Florence Sedes[a]

[a] *IRIT Laboratory, University of Toulouse, Toulouse, France*
[b] *Department of Electrical and Computer Engineering, Birzeit University, Ramallah, Palestine*

Abstract

Online social networks (OSNs) have become an important source of information for a tremendous range of applications and researches such as search engines, and summarization systems. However, the high usability and accessibility of OSNs have exposed many information quality (IQ) problems which consequently decrease the performance of the OSNs dependent applications. Social spammers are a particular kind of ill-intentioned users who degrade the quality of OSNs information through misusing all possible services provided by OSNs. Social spammers spread many intensive posts/tweets to lure legitimate users to malicious or commercial sites containing malware downloads, phishing, and drug sales. Given the fact that Twitter is not immune towards the social spam problem, different researchers have designed various detection methods which inspect individual tweets or accounts for the existence of spam contents. However, although of the high detection rates of the account-based spam detection methods, these methods are not suitable for filtering tweets in the real-time detection because of the need for information from Twitter's servers. At tweet spam detection level, many light features have been proposed for real-time filtering; however, the existing classification models separately classify a tweet without considering the state of previous handled tweets associated with a topic. Also, these models periodically require retraining using a ground-truth data to make them up-to-date. Hence, in this paper, we formalize a Hidden Markov Model (HMM) as a time-dependent model for real-time topical spam tweets filtering. More precisely, our method only leverages the available and accessible meta-data in the tweet object to detect spam tweets exiting in a stream of tweets related to a topic (e.g., #Trump), with considering the state of previously handled tweets associated to the same topic. Compared to the classical time-independent classification methods such as Random Forest, the experimental evaluation demonstrates the efficiency of increasing the quality of topics in terms of precision, recall, and F-measure performance metrics.

Keywords: Hidden Markov Model, Social Spam, Real-Time, Twitter.

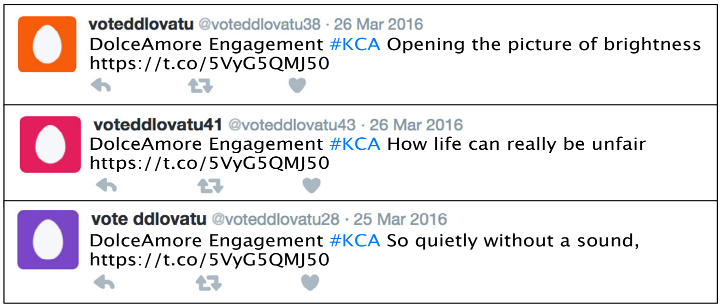
1. Introduction

Online Social Networks (OSNs) have an enormous popularity over the Internet because of the wide range of services that they provide for their users. For example, the most popular OSNs such as Twitter, and Facebook have exceeded billions of registered users and millions of daily active users[1]. The key point of the OSNs is their dependency on users as primary contributors in generating and posting information. Users' contributions might be exploited in different positive ways such as understanding users' needs, and analyzing users' opinions for election

* Corresponding author
E-mail addresses: mahdi.washha@irit.fr (Mahdi Washha)., aqaroush@birzeit.edu (Aziz Qaroush)., Mezghani.Manel@gmail.com (Manel Mezghani)., florence.sedes@irit.fr (Florence Sedes).

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of KES International
10.1016/j.procs.2017.08.075

purposes. However, the simplicity and flexibility of using OSNs in addition to the absence of an effective restrictions on content posting action have exposed different information quality problems such as social spam, and information overload[2]. Indeed, these characteristics have subjected OSNs to different attacks by ill-intentioned users, so-called social spammers, to post spam content. Social spammers intensively post nonsensical content in different contexts (e.g, topics) and in an automated way[3]. For example, posting a tweet talking about ”how to gain 100$ in 5 minutes” under the ”#BBC” topic is a spam tweet since such a tweet has no relation to the given topic. Thus, social spammers have a wide range of goals when publish a spam content in OSNs, summarized in[4]: (i) spreading advertisements to generate sales; (ii) disseminating porn materials; (iii) publishing viruses and malware; (iv) and creating phishing websites.


voteddlovatu @voteddlovatu38 · 26 Mar 2016
DolceAmore Engagement #KCA Opening the picture of brightness https://t.co/5VyG5QMJ50

voteddlovatu41 @voteddlovatu43 · 26 Mar 2016
DolceAmore Engagement #KCA How life can really be unfair https://t.co/5VyG5QMJ50

vote ddlovatu @voteddlovatu28 · 25 Mar 2016
DolceAmore Engagement #KCA So quietly without a sound, https://t.co/5VyG5QMJ50

Figure 1. An example of three correlated spam tweets posted in a consecutive way by three different spam accounts.

Motivation and Problem. Since OSNs have many information quality problems, in this work, a particular issue related to the social spam problem in Twitter platform has been addressed. More precisely, we address the problem of filtering out the spam tweets that might exist in a stream of tweets related to a Twitter topic to increase the topics content quality, with taking into consideration the real-time aspect of the filtration process. The proposed solution, has been integrated with our team researches on social networks. The research interests of our team addressing many issues related to OSNs such as tweet summarization, event detection[5], social profiling[6], profiles enrichment[7], and socio-semantic communities detection[8], where Twitter platform has been adopted as a source of information in most of them. Thus, experimenting and working on a high quality of Twitter data (Tweet content) is an indispensable step to achieve high performance results in our team researches. Besides the information quality requirement, some research topics, such as tweets summarization and events detection, require real-time spam tweets filtering.

As Twitter is not immune towards the social spam problem[1], a set of methods has been introduced in the literature for detecting spam campaigns and individual spam accounts[4,3,9,10,11,12,13,14], with little effort spent for individual spam tweets detection[4,15,16]. These efforts mainly exploit supervised machine learning methods combined with the features extraction concept to produce binary classifiers using annotated data-sets. However, some of these methods such as campaign and account based methods are ”not” suitable for real-time filtration because their features require an additional information from Twitter’s servers. The only legal way to retrieve these additional information is by using REST APIs[1] which are provided by Twitter for developers and researchers. However, Twitter imposes limitations and constraints (e.g., limiting the number of calls to a time slot) on using REST APIs, decreasing the applicability of such methods in the a real-time way. Moreover, exploiting graph-based features such as node betweenness, and sender-receiver distance require an exponential number of REST APIs calls to extract their values.

Most of the features that are used at tweet-level detection such as number of words are light in computation and thus they are suitable for real-time spam tweets detection. However, given the fact that social spammers are dynamic in their content and strategies[3], these light features are not strongly discriminant among spam and non-spam tweets. Also, the combination of these weak features is not necessary to produce robust binary classification models, since social spammers are easily manipulate in these features value. The straightforward and trivial solution to address such a problem is designing new light features having enough discriminant power among spam and non-spam tweets. However, this solution is not possible since the tweet content is limited to 140 characters and the simple available meta-data about its user (e.g., username attribute) increases the difficulty to design new robust light features. Beyond the feature design level, the approaches followed in building spam classification models are time-independent learning algorithms (e.g., Random-Forest, Support Vector Machine) in which the learning and classification steps are performed without considering the state of previous classified instances. Also, updating and tunning the classification models (i.e., Model parameters) that use those learning methods require a wide range of training and validation to

[1] https://dev.twitter.com/rest/public

obtain the desired results. Furthermore, performing these steps are time consuming, especially when treating large-scale collections of tweets, rather than the need for new ground-truth data-sets.

Time Dependent Model. When social spammers launch their spam campaigns through creating thousands of spam accounts, in an automated way, they might intensively attack the current trending topics at that moment. Thus, the probability of the trending topics to have a stream of spam tweets is higher than those topics that are not trending ones. For example, Figure 1 shows a series of spam tweets attacked the ”KCA” topic by three correlated spam accounts (social spammers). Besides the prior probability of trending topics being attacked, the state (spam or non-spam) of last streamed tweets to a particular topic can contribute in detecting spam tweets when treating new incoming tweets. Hence, instead of classifying a tweet independently, we hypothesize that the involvement of the prior probability to receive spam tweets and the probability of the handled tweets for being a spam can increase the performance of filtering spam tweets.

Contributions. In this paper, we introduce the design of a topic based Hidden Markov Model (HMM) for detecting spam tweets in a real-time way. The proposed model allows to perform an inference about the state of the incoming stream of tweets through: (i) taking and updating the prior probability to have a spam tweet using previous handled tweets; (ii) and by considering the transition probability between spam and non-spam states. We demonstrate the effectiveness of proposed method through a series of experiments conducted on a crawled and an annotated dataset containing more than six million tweets belonging to 100 topics. The experimental results show that our method has a superior performance in detecting spam tweets, compared to the time-independent supervised machine learning methods. The main strength of the proposed method is in tuning the method's parameters in an online fashion without requiring any kind of retraining phases. Moreover, our method might be leveraged in different directions: (i) applications that exploit the trending topics of Twitter can use the proposed method to increase their performance through filtering out the spam tweets to increase data quality; (ii) and Twitter might utilize the proposed method to improve its anti-spam mechanism.

The remainder of the paper is organized as follows. Section 2 presents the Twitter's anti-spam mechanism as well as the Twitter social spam detection methods proposed in the literature. Section 3 presents notations, and the design of the proposed method. Section 4 describes the dataset used in experimenting and validating the proposed method. The experimental setup and a series of experiments evaluating the proposed approach are described in section 5. At last, section 6 concludes the work with giving directions as a future work.

2. Background and Related Work

Social Spam Definition. Social spam is defined as a nonsensical or a gibberish text content appearing on OSNs and any website dealing with user-generated content such as chats and comments[2]. Social spam may take tremendous range of forms, including profanity, insults, hate speeches, fraudulent reviews, personally identifiable information, fake friends, bulk messages, phishing and malicious links, and porn materials. One might view the social spam as an irrelevant information; however, this interpretation is quite not accurate. We justify this misinterpretation through the definition of information retrieval (IR) systems[17] in which the relevancy of documents in IR systems is dependent on the input search query. Thus, irrelevant documents with respect to an input query are ”not” necessary to be a spam content. Hence, as an additional definition, social spam might be defined as irrelevant information that doesn't have an interpretation in any context as long as the input query is not a spam in content.

Topic Definition. The concept of ”Topic”[2] can be defined as a representation of hidden semantic structures in a text collection (e.g., textual documents, or tweets). On the other hand, ”Trending Topic” is a key word or phrase (e.g., #Trump, #KCA, and TopChef) that is mentioned at a greater rate than the others. Trending topics become popular either because of a concerted effort by users, or due to an event that encourages users to talk about a particular topic. The main purpose of trending topics is to help users in understanding what is occurring in the world and what users' opinions are about it in a real-time manner. Trending topics are automatically identified by Twitter through an algorithm that identifies topics that are massively circulated among users more than other topics.

Twitter's Anti-Spam Mechanism. Twitter provides an option for its users to report spam accounts through clicking on ”Report: they are posting spam” option that is available in all accounts. Once an account is reported, Twitter's administrators manually review that account to make a suspension decision. However, combating social spammers

[2] https://support.twitter.com/articles/101125#

using this reporting mechanism is inefficient because of the need for great efforts from both users and administrators. Moreover, not all reports are trustworthy, meaning that some reported accounts might be for legitimate users, not for social spammers. In addition to this manual reporting mechanism, Twitter has defined some general rules (e.g., not allowed to post porn materials) for public in order to reduce the social spam problem as much as possible with suspending permanently the accounts that violate those rules[18]. However, social spammers can bypass Twitter's rules. For instance, social spammers may coordinate multiple accounts and distributing the desired workload among them to mislead the detection process. These accounts tend to exhibit an invisible spam behavior. Thus, these shortcomings have motivated researchers to introduce more robust methods to increase data quality which can be used for the applications that use Twitter as source of information such as tweet summarization. We categorize the Twitter social spam detection approaches into two different types based on the automation detection level: (i) machine learning level as a fully automated approach; (ii) and social honeypot as a manual approach requiring human interactions.

Machine Learning Approach. In this approach, researchers have built their methods through employing three levels of detection, distributed between tweet-level detection, account-level detection, and campaign-level detection.

Tweet-Level. Martinez-Romo and Araujo[19] have designed a language model based method to detect spam tweets existing in topics. The method compute the kullback-leibler divergence between the language model of each tweet and the language model of the topic itself. However, this method is not suitable for real-time filtering because of the need for the tweets that have been posted in the same topic from Twitter's servers. The works introduced in[4,15,16] have proposed a set of light statistical features such as number of words with a set of time-independent machine learning algorithms such as support vector machine (SVM), and Random Forest, to build a binary classifier. Although of suitability of these works for real-time filtering; however, they have a major drawback in detecting efficiently spam tweets (i.e., low spam-recall values) due to the high evolving of spam content overtime.

Account-Level. The works introduced in[4,3,9,10,11,12,13,14] have focused on extracting feature from users' accounts, including the number of friends, number of followers, similarity between tweets, and ratio of URLs in tweets. In more dedicated studies, the works proposed in[20,21] have identified the spam URLs through analyzing the shorten URLs behavior like the number of clicks and the length of redirection chain. However, the ease of manipulation in this type of features by spammers has given a motivation to extract more complex features by employing graph theory. For instance, the studies presented in[22,23,24] have examined the relation among users using some graph metrics to measure three features, including the node betweenness, local clustering, and bi-directional relation ratio. Leveraging such complex features gives high spam accounts detection rate; however, they are not suitable for real-time Twitter-based applications because of the huge volume of data that must be retrieved from Twitter's servers.

Campaign-Level. Chu et al.[25] have treated the spam problem from the collective perspective view. They have clustered a set of desired accounts according to the URLs available in the posted tweets, and then a defined set of features which are extracted from the clustered accounts to be incorporated in identifying spam campaign using machine learning algorithms. Chu et al.[26] have proposed a classification model to capture the difference among bot, human, and cyborg with considering the content of tweets, and the tweeting behavior. Indeed, the methods belonging to this detection level have a major drawback that the methods use features requiring a great number of REST API calls to obtain information like users' tweets and followers. Consequently, exploiting the current version of campaign level methods is not appropriate for real-time filtering due to the high volume of data required from Twitter's servers.

Beyond the features design level, the works introduced in[27,28] have proposed two optimization frameworks which use the content of tweets and basic network information to detect spam accounts using an efficient online learning approach. However, the major limitation in these works is the need for information about the network from Twitter, making them unsuitable for real-time filtering at all.

Honeypot Approach. Social honeypot is viewed as an information system resource that can monitor social spammers' behavior through logging their information such as the information of accounts and any available content[29]. In fact, there is no significant difference between Twitter's anti-spam mechanism and the social honeypot approach. Both of them need an administrative control to produce a decision about the accounts that have fallen into the honeypot trap. The necessity of administrative control is to reduce the false positive rate, as an alternative solution to blindly classifying all users dropped in the trap as spam users.

3. Hidden Markov Model: Definitions, Learning, and Inference

Classical supervised machine learning methods perform training to build a classifier that can correctly predict the classes of new unseen samples. However, these methods mainly assume that the training samples are independent. Although of the applicability of such an assumption in a wide range of applications, this assumption is not strong in the context of social spam. Intuitively, social spammers might intensively attack topics to propagate their spammy contents as fast as possible. Thus, a high dependency might exist among successive tweets when they are streamed into a particular topic. This means that we can improve the detection of spam tweets by applying prior and transition state-based prediction models.

Hence, we propose the use of Hidden Markov Model (HMM) as a state prediction model to detect spam tweets in real-time topic-based streaming. HMMs are statistical models used to represent the probability distributions of discrete hidden states (e.g., spam or non-spam, and robot coordinates) over sequences (time series) of observations (e.g., number of words in tweet, and distance from object taken by sensor)[30]. HMM works based on two assumptions: (i) the observations at time *t* had been generated by some hidden state S_t where state S_t is hidden from the observer; (ii) and the hidden state must satisfies the Markov property in which the current state S_t is dependent only on the value (output) of the previous state S_{t-1} . In other words, the Markov conditional property assumes that the prediction of the next state at some time depends only upon the present state, not on the sequence of events that preceded it. .

Since HMM models are controlled and tunned by different important parameters, first, we introduce the design and notations used in building topic-based HMM model. Then, we mathematically illustrate a supervised learning approach to learn the HMM's parameters. At last, we describe how the inference step takes place to handle incoming tweets.

3.1. HMM Model Design and Notations

Each topic has a stream of tweets posted by one or more users at different times. Formally, we model each topic as a time series of tweets ordered based on their streaming time (i.e., posting date), defined as $Topic = \{T_1, \ldots, T_t\}$, where $t \in \mathbb{Z}^+$ represents the time of the most recent tweet streamed into the topic. Twitter provides a defined set of attributes in the tweet object which represents information about the user who has posted the tweet as well as its textual content. Hence, we model each tweet, $T_\bullet \in Topic$, by 5-tuple $T_\bullet = \langle Age, \#Followers, \#Followees, \#Retweets, Text\rangle$ where Age is the age of the user's account, $\#Followers$ is the number of accounts that follow user tweet $T_\bullet$, $\#Followees$ is the number of accounts that the used of the tweet $T_\bullet$ follows, $\#Retweets$ is the number of retweets that the tweet $T_\bullet$ has gained by other Twitter users, and $Text = \{w_1, w_2, \ldots\}$ is a finite set of ordered words representing the textual content of the tweet.

In the context of spam tweets detection, the states of the hidden process are "spam" and "non-spam". Since these states are not directly observable, we adopt *M* random independent variables forming a vector of observations $\mathbf{O}_t$ extracted from a tweet T_t streamed at time *t*. In order to predict the state (S_t) of the process at time *t* for a given sequence of observations $\{\mathbf{O}_1, \ldots, \mathbf{O}_t\}$, a prior knowledge is required about (i) the state transition matrix (*A*), (ii) initial state probability distribution (π), (iii) and the emission probability distributions ($\{b_{spam,1}, \ldots b_{spam,M}, b_{non-spam,1}, \ldots b_{non-spam,M}\}$). The state transition matrix contains probability values reflecting the chance of transitioning from state S_i to state S_j at any time *t*. For instance, $a_{spam,non-spam}$ is the probability of posting "spam" tweet stream given that the previous tweet streamed is "non-spam" (i.e., $P(S_t = "spam" | S_{t-1} = "non-spam")$). The π probability distribution contains *N* probability values reflecting the chance to start the hidden process in a particular state (e.g., probability of streaming a spam tweet). The third important parameter is the emission (observation) probability distribution $b_{i,v}$ of each adopted feature with respect to each state, requiring $N \times M$ distributions to perform the inference process. For instance, $b_{spam,1}$ is a probability distribution of observing the values of feature "1" (e.g., number of words in the tweet) when the process be in the "spam" state. Table 1, provides a summary of the notations used in formalizing the proposed HMM model beside their potential values.

3.2. Learning and Tuning HMM Parameters

Two approaches (supervised and unsupervised) are widely used in learning HMM parameters, denoted by $\theta = \{\pi, A, b_{spam,1}, \ldots b_{spam,M}, b_{non-spam,1}, \ldots b_{non-spam,M}\}$. Expectation Maximization (EM)[31] is widely used as an unsupervised learning approach to find the optimal HMM parameters. This approach assumes that the true hidden state of

Table 1. A description of the HMM notations with their potential values in the social spam detection problem.

Notation	**Description**	**Values in our model**
N	Number of possible states in the model.	2
M	Number of independent observation variables (e.g., features).	16
$Q = \{q_0, ..., q_{N-1}\}$	Set of distinct N hidden states of the Markov process.	Q={Spam, Non-Spam}
S_t	Hidden state value of the process at time t.	$S_t \in Q, t \in \mathbb{Z}^+$
A	Time invariant square matrix of state transition probabilities with size of NxN.	$A \in \mathbb{R}^{2\times2}$, $\sum_{i,j\in Q} a_{i,j} = N$
$a_{i,j}$	Transition probability from $j \in Q$ state to $i \in Q$ state.	$a_{i,j} \in \mathbb{R}$
π	Initial probability distribution of the possible states Q.	$\pi \in \mathbb{R}^2$
π_i	Initial probability of the state $i \in Q$.	$\pi_i \in \mathbb{R}$, $\sum_{i\in Q} \pi_i = 1$
$O_{t,v}$	Observation value of the v^{th} independent random variable (feature) at time t.	$O_{t,v} \in \mathbb{R}$, $v \in \{1, \dots, M\}$
$\mathbf{O}_t$	Vector of M observations,$[O_{t,1}, ..., O_{t,v}, ..., O_{t,M}]^T$, observed at time t,	$\mathbf{O}_t \in \mathbb{R}^M$
$b_{i,v}(o)$	Emission probability distribution of the v^{th} random independent variable of the state $i \in Q$.	$b_{i,v}(o) \in [0, 1]$, $o \in \mathbb{R}$

each tweet (tweet label) is *not* given in the training data. However, in this work, we leverage the supervised learning approach since it is possible to predict the true state of a tweet but *not* in real time. For instance, inspecting and analyzing deeply the user of a tweet using account-based features is a possible way to predict the true state of a tweet with an acceptable accuracy. Also, checking whether the user of a tweet has been suspended by Twitter is an alternative way.

Let $Traning\_Tweets \subset Topic$ be a set of tweets already streamed into a particular topic. Computing π probability distribution requires independent sequences of observations with their true states. Thus, we divide the $Traning\_Tweets$ set into m sequences based on their posting date "day" of the tweets (e.g., tweets posted in "4/May/2016" forming a single sequence). More formally, let $L = \{(z_1, x_1, y_1), ..., (z_m, x_m, y_m)\}$ be a set of m sequences of tweets streamed into $Traning\_Tweets$, where $(z_\bullet, x_\bullet, y_\bullet) = (\{T_{t_1}, \dots, T_{t_2}\}, \{\mathbf{O}_{t_1}, ..., \mathbf{O}_{t_2}\}, \{S_{t_1}, ..., S_{t_2}\})$, $1 \leq t_1 < t_2$, $z_\bullet$ is a sequence of tweets having same posting date day, $x_\bullet$ is a sequence of observation (feature) vectors corresponding to the tweets sequence in $z_\bullet$, and $y_\bullet$ is a sequence of true hidden states corresponding to the tweets sequence in $z_\bullet$.

By leveraging the training set of tweets, we can estimate the HMM parameters, θ, using the Maximum Likelihood Estimation (MLE) method[32] for a given set of observations. More precisely, the MLE finds the parameter values that maximize the likelihood of generating the given observations. We formalize the searching for the optimal HMM parameters as a constraint maximization optimization problem, defined as:

$$\theta_{mle} = \underset{\theta\in\Theta}{\arg\max} \quad P(L|\theta) \quad \text{subject to} \quad \sum_{i\in Q} \pi_i = 1, \prod_{j\in Q} \sum_{i\in Q} a_{i,j} = 1, \prod_{i\in Q} \prod_{v\in\{1,...,M\}} \int_{\mathbb{R}} b_{i,v}(x)dx = 1 \tag{1}$$

where Θ is the search space of HMM parameters. In HMM model, the sum of the discrete π distribution probabilities must equal 1. Also, the sum of each column in the transition matrix A must equal 1 and thus the multiplication of the column summation must equal 1. The last constraint is associated to $N \times M$ emission probability distributions where the area under each distribution must equal 1 and thus the multiplication of the $N \times M$ distributions area must equal 1.

The probability of generating any sequence of observation can be written using the HMM parameters θ as follows:

$$P((z_\bullet, x_\bullet, y_\bullet)|\theta) = \prod_{i\in Q} \pi_i^{f(i,y_\bullet)} . \prod_{i\in Q} \prod_{j\in Q} a_{i,j}^{f(i,j,y_\bullet)} . \prod_{i\in Q} \prod_{\mathbf{O}_t \in x_\bullet} \prod_{o_{t,v}\in\mathbf{O}_t} b_{i,v}(o_{t,v})^{f(i,v,o_{t,v},x_\bullet,y_\bullet)} \tag{2}$$

where $f(i, y_\bullet)$ is an integer (0 or 1) value representing the occurrence of the state i in the start of a given sequence $y_\bullet$, $f(i, j, y_\bullet)$ is the number of occurrence of having a transition from the state j to the state i in the given true states sequence $y_\bullet$, and $f(i, v, o_{t,v}, x_\bullet, y_\bullet)$ is the number of observing the value $o_{t,v}$ of feature v in the given sequence of observations $x_\bullet$ when the process be in a the state i.

Thus a closed form solution to find the optimal HMM parameters θ can be obtained by employing the assumption regarding each sequence of the training m sequences, which states that the probability of generating the L set equals to the probability multiplication of generating each sequence individually. This assumption formally defined using HMM parameters as:

$$P(L|\theta) = \prod_{l=1}^{m} P((x_l, y_l)|\theta) = \prod_{l=1}^{m} \left(\prod_{i\in Q} \pi_i^{f(i,y_\bullet)} . \prod_{i\in Q} \prod_{j\in Q} a_{i,j}^{f(i,j,y_\bullet)} . \prod_{i\in Q} \prod_{\mathbf{O}_t \in x_\bullet} \prod_{o_{t,v}\in\mathbf{O}_t} b_{i,v}(o_{t,v})^{f(i,v,o_{t,v},x_\bullet,y_\bullet)} \right) \tag{3}$$

Since the $P(L|\theta)$ which can be viewed as a function of θ, $P(L|\theta)$ is a concave function having only one global maximum value. Hence, we compute the partial derivatives of $P(L|\theta)$ with respect to each free parameter in, θ, $(a_{spam,spam}, \pi_{spam}, b_{spam,1}, \dots)$. The task now is turned to solve and find the value of each free parameter that makes each partial derivatives equal to zero ($\frac{\partial P(L|\theta)}{\partial a_{spam,spam}} = 0$, $\frac{\partial P(L|\theta)}{\partial \pi_{spam}} = 0$, $\frac{\partial P(L|\theta)}{\partial b_{spam,1}} = 0, \dots$). Thus, the optimal value of the HMM parameters can be computed using in a closed form way using the following three equations:

$$\pi_i = \frac{\sum_{l=1}^{m} f(i, y_l)}{\sum_{k \in Q} \sum_{l=1}^{m} f(k, y_l)} \qquad a_{i,j} = \frac{\sum_{l=1}^{m} f(i, j, y_l)}{\sum_{k \in Q} \sum_{l=1}^{m} f(i, k, y_l)} \qquad b_{i,v}(o) = \frac{\sum_{l=1}^{m} f(i, v, o, x_l, y_l)}{\sum_{l=1}^{m} \int_{\mathbb{R}} f(i, v, o', x_l, y_l) do'} \tag{4}$$

These optimal values implicitly satisfy the added constraints in equation 1 since the denominator of each value is a normalization factor for the numerator.

3.3. Inference

After computing the optimal HMM model parameters values θ, the next step is to use the tuned model as a time-dependent tweet spam prediction model to handle new incoming tweets related to a given topic. Let T_t be the first tweet streamed into the considered topic and $\mathbf{O}_t$ be the corresponding feature vector (observation) of the tweet. The likelihood process state when observing tweet T_t is computed as:

$$S_t^* = \arg\max_{i \in Q} P(S_t = i|\mathbf{O}_t) = \arg\max_{i \in Q} \sum_{j \in Q} P(S_{t-1} = j)\, a_{i,j} \prod_{o_v \in \mathbf{O}_t} b_{i,v}(o_v) \tag{5}$$

where $P(S_0 = j) = \pi_j$. The class label of the streamed tweet is the likelihood inferred state S_t.

4. Data-set Description and Ground Truth

The data-sets used at tweet level detection[15,4,19] are not publicly available for research use. Also, for privacy reasons, social networks researchers provide only the interested object IDs (e.g., tweets, and accounts) to retrieve them from servers of the target social network. However, inspired by the nature of spam problem, providing IDs of spam tweets or accounts is not enough because Twitter might already have suspended the corresponding accounts.

Table 2. Statistics of annotated users and tweets of 100 trending topics.

Number of Trending Topics	100
Number of Users	2,088,131
Number of Tweets	6,470,809

	Spam	non-Spam
Number of Tweets	763,555 (11.8%)	5,707,254 (88,2%)
Number of Users	185,843(8.9%)	1,902,288(91.1%)

Crawling Method. Hence, we exploit our research team crawler to collect accounts and tweets, launched since 1/Jan/2015. The streaming method is used to get an access for 1% of global tweets, as an unbiased crawling way. Such a method is commonly exploited in the literature to collect and create data-set in social networks researches.

Data-set Description. Using our team Twitter data-set, we have clustered the collected tweets based on the topic available in the tweet while ignoring the tweets that do not contain any topic. Then, we have selected the tweets of 100 trending topics (e.g. #Trump) which are randomly sampled to conduct our experiments. Table 2 shows the main statistics of the selected topics. We have exploited this way in crawling and sampling to remove any possible biasing in the data, and to draw unbiased conclusions.

Ground Truth Data-set. We have created an annotated data-set through labeling each account (user) as a spam or non-spam. However, with the huge amount of accounts, using manual annotation approach to have labeled data-sets is an impractical solution. Hence, we leverage a widely followed annotation process in the social spam detection researches. The process checks whether the user of each tweet was suspended by Twitter. In case of suspension, both the user and his tweets are labeled as a spam; otherwise we assign non-spam for both of them. In total, as reported in Table 2, we have found that more than 760,000 tweets were classified as spam, posted by almost 185,800 spam accounts.

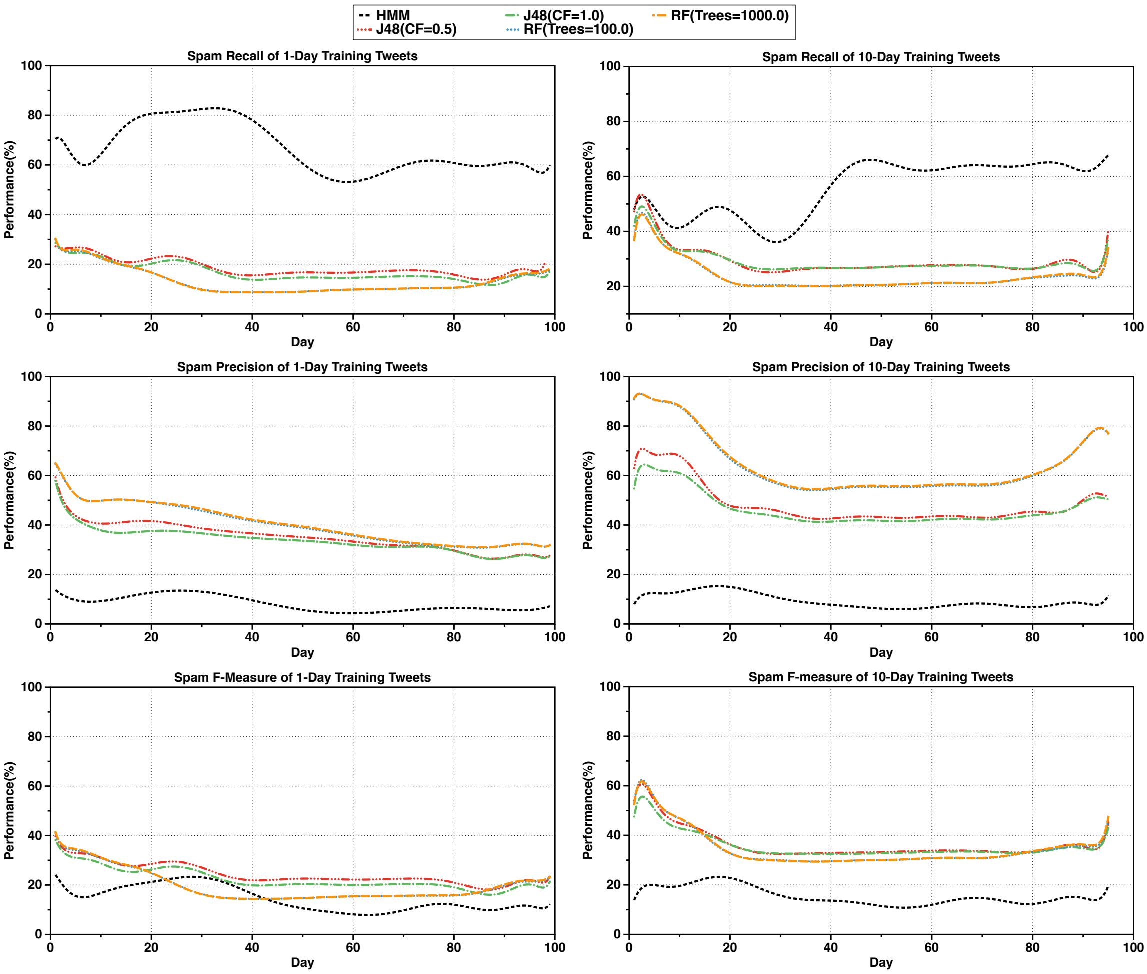


Figure 2. Performance results for different classical supervised learning methods (Random Forest and J48 decision tree) and the HMM in terms of spam recall, spam precision, and spam F-measure. The left column represents the results when performing the training stage on the first day of the streamed tweets of 100 topics, while using the tweets of the rest 99 days for testing stage. In the other-hand, the right column represents the results when performing training on the first 10-day of streamed tweets while doing the testing on the rest 90 days tweets.

5. Results and Evaluations

5.1. Experimental Setup

Performance Metrics. Since the ground truth class-label of each user is available, we exploit commonly used metrics in classification problems to asses the proposed approach. These metrics includes: precision, recall, F-measure, average precision, and average recall, computed according to the confusion matrix of the Weka tool[33]. We don't exploit the accuracy metric since our data-set is imbalanced (e.g. 11.8% spam tweets and 88,2% non-spam tweets) and thus it is not a useful metric in such a case. Also, since the problem is two-class binary classification, we compute the precision, recall, and F-measure for the "spam" class, while using the average metrics to combine both classes based on the fraction of each class (e.g. 4.9% * "spam precision" + 95.1% * "non-spam precision").

Baseline Method. The performance of the proposed time-dependent method using HMM is compared with the performance results of time-independent supervised learning methods. Hence, we adopt the Random Forest (RF) with $\#Trees \in \{100, 1000\}$, and the J48 decision tree with two different confidence factors $CF \in \{0.5, 1.0\}$, as a well-known time-independent supervised learning methods used in spam tweet detection problem[4,16,15,19]. We use the implementation of the Weka tool[33] for these two learning methods. For the features set, we exploit 16 features which are commonly used in the literature[4,16,15,19] and suitable for the real-time spam filtering. These features are the number of hashtags, number of URLs, number of words, number of characters, number of mentions, number of retweets, number of spam words, number of hashtags per words, number of URLs per words, number of numeric

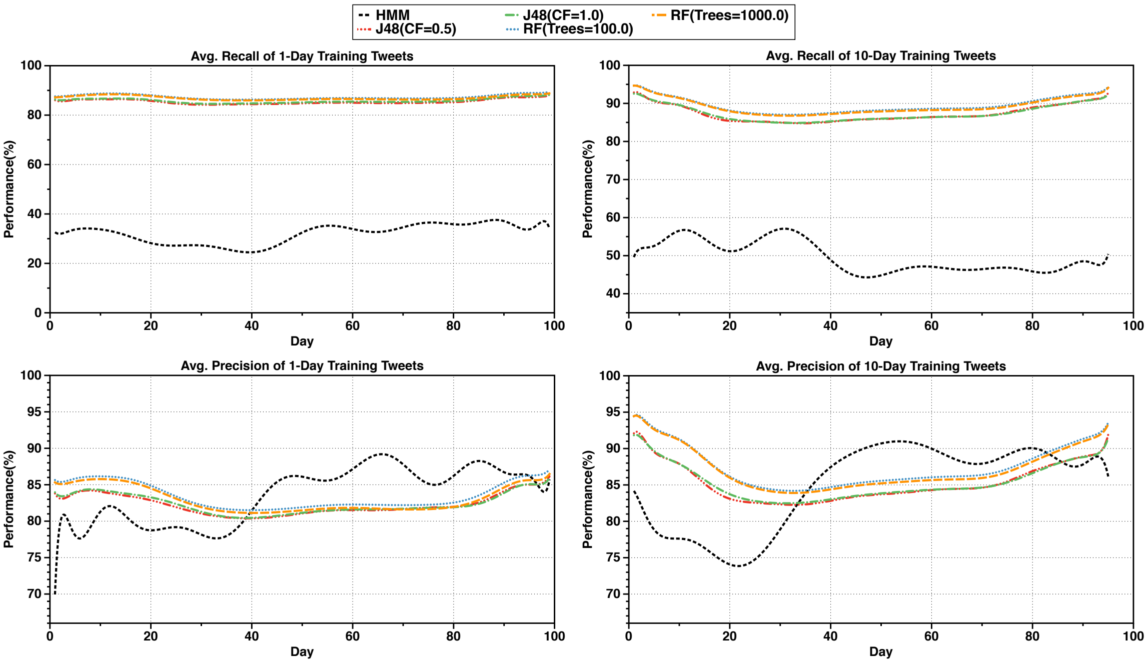


Figure 3. Performance results of different classical supervised learning methods (Random Forest and J48 decision tree) and the HMM in terms of average recall, and average precision. The left column represents the results when performing training stage on the first day of the streamed tweets of 100 topics, while using the tweets of the rest 99 days for testing stage. In the other-hand, the right column represents the results when performing training on the first 10-day of streamed tweets while doing the testing on the rest 90 days tweets.

characters, number of replies, number of favorites, number of followings, number of followers, tweet's user age, number of tweets posted by the tweet's user, and number of lists. All of these features are extractable from the tweet object directly without needing any further information from Twitter's servers.

HMM Model Setting. In this work, we adopt the tweet features introduced in the baseline methods, to represent the observations vector, $\mathbf{O}_\bullet \in \mathbb{R}^M$, where M represents the length of the observations vector and equals to the number of features ($M = 16$). Since each feature is a continues independent random variable, we model each corresponding emission probability distribution, $b_{i,v}$, $v \in \{1 \dots M\}$ of state $i \in Q$ as a Gaussian probability distribution, $\mathcal{N}(\mu_{i,v}, \sigma^2_{i,v})$, with mean and variance of $\mu_{i,v}$, $\sigma^2_{i,v}$, respectively. For each Gaussian probability distribution, we compute the optimal parameters value using the Maximum Likelihood Estimation method (MLE).

5.2. *Experimental Results*

The main purpose of our experiments on the annotated data-set is to study three aspects, summarized in (i) getting insight into the effect of time-dependent learning models in detecting effectively and efficiently spam tweets of trending topics, compared to the classical learning methods; (ii) studying the impact of the training data size (number of tweets) on the performance of spam detection; (iii) and to which extent the learned detection models can have a stable performance without dramatic degradations. Hence, in this paper, we study the impact of training of 1-day tweets and 10-day tweets streamed into 100 trending topics.

The graphs drawn in Figure 2 and 3 show the average performance results over 100 trending topics when applying two time-independent supervised learning methods (Random Forest and J48), and the proposed time-dependent HMM method. The left column shows the results when considering the 1-day tweets of each trending topic as a training set, while the second column shows the results when using the 10-day tweets as a training set. The streaming period of each topic is 100 days. Thus, in the first experiment, we have tested the models on the rest 99-day tweets, while in the second experiment, we have leveraged the 90-day tweets in experimenting the classification models. Each graph consists of five curves (HMM, 2 Random Forest, 2 J48) which represent the results in terms of the proposed performance metrics (e.g., spam recall), and the number of days used in the training stage.

The spam recall curves in Figure 2 show the strength of proposed time-dependent HMM model in detecting spam tweets effectively, compared to the Random Forest and J48 as time-independent classification models. The exploitation of the 1-day tweets as a training set has contributed in having a good model, since the average spam recall over the 99-day is about 70% for HMM, compared to 20% as an average spam recall obtained by Random Forest and J48. As an interpretation, the degradation in the spam recall of HMM after the 40-day is because the emission probability distributions of the spam class have given low probability values in that period. These low values of emission probabilities are due to the new spam campaigns that have been launched by spammers in that period which had not been modeled well during the training phase. On the other side, performing training on 10-day tweets has increased the spam recall values when using Random Forest and J48 method for the first 20-day. Afterward, the spam recall values have decreased to about 25%. This ensures the true fact about the dynamicity of social spammers in their spam content. Conversely, the HMM has performed badly in the first 30-day and that because of having low transition probabilities ($a_{spam,spam}$ and $a_{spam,non-spam}$). For the rest of 60-day, the performance of HMM has been increased since more tuning have been achieved regarding the prior state probability HMM model.

In terms of the spam precision metric, Random Forest and J48 methods have superior performance in all testing days, compared to the performance of HMM. The low transition probability $a_{non-spam,spam}$ value is the main source of having low spam precision values since it is possible to receive non-spam tweets after spam ones. The amount of training tweets has an obvious impact on the time-independent classification models (Random Forest and J48) since the precision values of these models have increased, while the HMM has almost the same performance when exploiting the 1-day tweets in training phase. However, the spam precision of Random Forest and J48 has decreased after 10-day of streaming tweets. This degradation in the results is because of receiving new spam content with new behaviors which are not captured in the training phase. Consequently, the low values in the spam precision metric have resulted a low spam F-measure values, making the time-independent methods better than the HMM, especially when using 10-day tweets in the training phase.

According to the results reported in Figure 2, it is consistence to have low average recall (mathematically equals to Accuracy metric) values obtained by the HMM, as shown in Figure 3. The high spam recall and low spam precision values for HMM model means that too many non-spam tweets have been classified as spam tweets and thus the non-spam recall values must be low. Since the non-spam tweet class is the dominant in our data-set, the average recall is biased towards the highest class ratio. Although of the high spam recall values, the low non-spam recall values make the average recall low as well. The average precision values of the HMM give an indication that the HMM model has not classified many truly spam tweets as non-spam, leading to have high quality tweets. On the other side, the time-independent models have almost same average precision values over time with slight degradation from day to day.

Overall, based on the results drawn in Figure 2 and 3, we can draw the following conclusions: (i) relying on the time-independent classification models with the state-of-the-art features are *not* useful to detect spam tweets effectively; (ii) the low spam recall values obtained by the time-independent classification models ensure the dynamicity of spam contents in Twitter and thus adopting a static classification model is *not* a solution at all; (iii) as a dynamic model, the HMM is suitable to have high quality tweets since it has quite stable performance in recalling spam tweets over time.

6. Conclusion

In this paper, we have studied the impact of sequential data-based model (time dependent model) in detecting topical spam tweets in real-time. We have formalized a first-order Hidden Markov Model (HMM) as a dynamic and time-dependent model. Compared to the classical time-independent classification models, the HMM has shown its ability to detect spam tweets effectively, making it as the first preferable solution to have high quality topical tweets. With this new idea in the battle of fighting social spam, we plan as a future work to study the impact of higher orders of HMM with trying to propose a new set of tweet-based features suitable for real-time filtering.

References

1. Chen, C., Zhang, J., Xiang, Y., Zhou, W., Oliver, J.. Spammers are becoming" smarter" on twitter. *IT professional* 2016;**18**(2):66–70.
2. Agarwal, N., Yiliyasi, Y.. Information quality challenges in social media. In: *International Conference on Information Quality (ICIQ)*. 2010, p. 234–248.
3. Washha, M., Qaroush, A., Sedes, F.. Leveraging time for spammers detection on twitter. In: *Proceedings of the 8th International Conference on Management of Digital EcoSystems*. ACM; 2016, p. 109–116.
4. Benevenuto, F., Magno, G., Rodrigues, T., Almeida, V.. Detecting spammers on twitter. In: *In Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*. 2010, p. 12.

5. Hoang, T.B.N., Mothe, J.. Building a Knowledge Base using Microblogs: the Case of Cultural MicroBlog Contextualization Collection (regular paper). In: Balog, K., Cappellato, L., Ferro, N., Macdonald, C., editors. *Conference on Multilingual and Multimodal Information Access Evaluation (CLEF), Evora, Portugal, 05/09/2016-08/09/2016*; vol. 1609. http://CEUR-WS.org: CEUR Workshop Proceedings; 2016, p. 1226–1237. URL: `http://www.irit.fr/publis/SIG/2016_CLEF_HM.pdf`.
6. Mezghani, M., On-at, S., Péninou, A., Canut, M., Zayani, C.A., Amous, I., et al. A case study on the influence of the user profile enrichment on buzz propagation in social media: Experiments on delicious. In: *New Trends in Databases and Information Systems - ADBIS 2015 Short Papers and Workshops, BigDap, DCSA, GID, MEBIS, OAIS, SW4CH, WISARD, Poitiers, France, September 8-11, 2015. Proceedings*. 2015, p. 567–577.
7. Mezghani, M., Zayani, C.A., Amous, I., Péninou, A., Sèdes, F.. Dynamic enrichment of social users' interests. In: *IEEE 8th International Conference on Research Challenges in Information Science, RCIS 2014, Marrakech, Morocco, May 28-30, 2014*. 2014, p. 1–11. URL: `http://dx.doi.org/10.1109/RCIS.2014.6861066`. doi:`10.1109/RCIS.2014.6861066`.
8. Abascal-Mena, R., Lema, R., Sèdes, F.. Detecting sociosemantic communities by applying social network analysis in tweets. *Social Netw Analys Mining* 2015;**5**(1):38:1–38:17. URL: `http://dx.doi.org/10.1007/s13278-015-0280-2`. doi:`10.1007/s13278-015-0280-2`.
9. Wu, T., Liu, S., Zhang, J., Xiang, Y.. Twitter spam detection based on deep learning. In: *Proceedings of the Australasian Computer Science Week Multiconference*. ACM; 2017, p. 3.
10. Wang, A.H.. Don't follow me: Spam detection in twitter. In: *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*. 2010, p. 1–10.
11. McCord, M., Chuah, M.. Spam detection on twitter using traditional classifiers. In: *Proceedings of the 8th International Conference on Autonomic and Trusted Computing*; ATC'11. Springer-Verlag. ISBN 978-3-642-23495-8; 2011, p. 175–186.
12. Stringhini, G., Kruegel, C., Vigna, G.. Detecting spammers on social networks. In: *Proceedings of the 26th Annual Computer Security Applications Conference*; ACSAC '10. New York, NY, USA: ACM. ISBN 978-1-4503-0133-6; 2010, p. 1–9. doi:`10.1145/1920261.1920263`.
13. Meda, C., Ragusa, E., Gianoglio, C., Zunino, R., Ottaviano, A., Scillia, E., et al. Spam detection of twitter traffic: A framework based on random forests and non-uniform feature sampling. In: *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*. IEEE; 2016, p. 811–817.
14. Bara, I.A., Fung, C.J., Dinh, T.. Enhancing twitter spam accounts discovery using cross-account pattern mining. In: *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. IEEE; 2015, p. 491–496.
15. Chen, C., Zhang, J., Xie, Y., Xiang, Y., Zhou, W., Hassan, M.M., et al. A performance evaluation of machine learning-based streaming spam tweets detection. *IEEE Transactions on Computational Social Systems* 2015;**2**(3):65–76.
16. Chen, C., Zhang, J., Xiang, Y., Zhou, W.. Asymmetric self-learning for tackling twitter spam drift. In: *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*. IEEE; 2015, p. 208–213.
17. Manning, C.D., Raghavan, P., Schütze, H.. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press; 2008. ISBN 0521865719, 9780521865715.
18. Twitter, . The twitter rules. `https://support.twitter.com/articles/18311#`; 2016. [Online; accessed 1-March-2016].
19. Martinez-Romo, J., Araujo, L.. Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Systems with Applications* 2013;**40**(8):2992–3000.
20. Cao, C., Caverlee, J.. Detecting spam urls in social media via behavioral analysis. In: *Advances in Information Retrieval*. Springer; 2015, p. 703–714.
21. Wang, D., Pu, C.. Bean: a behavior analysis approach of url spam filtering in twitter. In: *Information Reuse and Integration (IRI), 2015 IEEE International Conference on*. IEEE; 2015, p. 403–410.
22. Yang, C., Harkreader, R.C., Gu, G.. Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. In: *Proceedings of the 14th International Conference on Recent Advances in Intrusion Detection*; RAID'11. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-642-23643-3; 2011, p. 318–337.
23. Yang, C., Harkreader, R., Zhang, J., Shin, S., Gu, G.. Analyzing spammers' social networks for fun and profit: A case study of cyber criminal ecosystem on twitter. In: *Proceedings of the 21st International Conference on World Wide Web*; WWW '12. New York, NY, USA: ACM. ISBN 978-1-4503-1229-5; 2012, p. 71–80. doi:`10.1145/2187836.2187847`.
24. Almaatouq, A., Shmueli, E., Nouh, M., Alabdulkareem, A., Singh, V.K., Alsaleh, M., et al. If it looks like a spammer and behaves like a spammer, it must be a spammer: analysis and detection of microblogging spam accounts. *International Journal of Information Security* 2016;**15**(5):475–491.
25. Chu, Z., Widjaja, I., Wang, H.. Detecting social spam campaigns on twitter. In: *Applied Cryptography and Network Security*. Springer; 2012, p. 455–472.
26. Chu, Z., Gianvecchio, S., Wang, H., Jajodia, S.. Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *Dependable and Secure Computing, IEEE Transactions on* 2012;**9**(6):811–824.
27. Hu, X., Tang, J., Liu, H.. Online social spammer detection. In: *AAAI*. 2014, p. 59–65.
28. Hu, X., Tang, J., Zhang, Y., Liu, H.. Social spammer detection in microblogging. In: *IJCAI*; vol. 13. Citeseer; 2013, p. 2633–2639.
29. Lee, K., Caverlee, J., Webb, S.. Uncovering social spammers: Social honeypots + machine learning. In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*; SIGIR '10. New York, NY, USA: ACM. ISBN 978-1-4503-0153-4; 2010, p. 435–442. URL: `http://doi.acm.org/10.1145/1835449.1835522`. doi:`10.1145/1835449.1835522`.
30. Ghahramani, Z.. An introduction to hidden markov models and bayesian networks. *International journal of pattern recognition and artificial intelligence* 2001;**15**(01):9–42.
31. Stamp, M.. A revealing introduction to hidden markov models. *Department of Computer Science San Jose State University* 2004;.
32. Le Cam, L.M.. *Maximum likelihood: an introduction*. JSTOR; 1979.
33. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.. The weka data mining software: An update. *SIGKDD Explor Newsl* 2009;**11**(1):10–18. doi:`10.1145/1656274.1656278`.