International Journal of Computers and Applications

ISSN: 1206-212X (Print) 1925-7074 (Online) Journal homepage: https://www.tandfonline.com/loi/tjca20

Spams classification and their diffusibility prediction on Twitter through sentiment and topic models

Mohammad Ahsan & T. P. Sharma

To cite this article: Mohammad Ahsan & T. P. Sharma (2020): Spams classification and their diffusibility prediction on Twitter through sentiment and topic models, International Journal of Computers and Applications, DOI: 10.1080/1206212X.2020.1758430

To link to this article: https://doi.org/10.1080/1206212X.2020.1758430

Published online: 04 May 2020.

INTERNATIONAL JOURNAL OF COMPUTERS AND APPLICATIONS
https://doi.org/10.1080/1206212X.2020.1758430

Taylor & Francis
Taylor & Francis Group

Spams classification and their diffusibility prediction on Twitter through sentiment and topic models

Mohammad Ahsan and T. P. Sharma

Computer Science and Engineering Department, National Institute of Technology, Hamirpur, India

ABSTRACT

Alike web spam has been a major issue to almost every aspect of the current World Wide Web, social spam has led to a serious threat to the utilities of online social media, particularly in information diffusion. To combat this challenge, the significance and impact of social media and its content should be analyzed critically. To address this issue, Twitter is used as a case study, where we have modeled the contents of information through topic models and coupled it with the user-oriented features to predict spam diffusion with a good accuracy. Latent Dirichlet Allocation (LDA), a widely used topic modeling technique is applied to capture the latent topics from the tweets' documents. The major contribution of this work is twofold. Firstly, analyzing the variation of sentiment and topics in spam/non-spam information, and secondly, constructing a feature set that classifies tweets more accurately into spam and non-spam category as compared to the existing approaches. The extensive simulation reveals the variation in sentiments and topics shared by the spam and non-spam tweets.

ARTICLE HISTORY
Received 31 July 2019
Accepted 16 April 2020

KEYWORDS
Decision tree; latent Dirichlet allocation; random forest; sentiment analysis; spam; Twitter

1. Introduction

Social media has emerged as a new communication paradigm that promotes the dissemination of information among the people. There is a proliferation of social media applications like Facebook, Twitter, YouTube, etc. They allow Internet users to produce, share, and consume content worldwide. A prominent feature of these sites is relationship formation among the users. These relationships serve as a channel to share information with others. Twitter has become a major source of information. Apart from providing the valuable information, social media also have the power of influencing people's perception. During 2016 US Presidential elections, it was found that Twitter played a key role in campaigning [1]. Social media also provides services to healthcare professionals in engaging with public, counseling patients, and consulting colleagues regarding health issues [2].

The rise in the popularity of Online Social Networks (OSNs), not only attracts legitimate users but also spammers. Legitimate users have a good purpose for using the services of OSNs i.e. maintaining relations with friends/colleagues, sharing information of interest, increasing the reach of their business through advertisement, spreading health related information, etc. While spammers misuse OSNs for illicit purposes, i.e. driving attention to unrelated products or services, lure others to click on malicious links [3]. These links are created with the intent to harm or misguide other people or their devices. A click on malicious links triggers activities ranging from downloading a malware attachment to stealing confidential information. So, preventive measures are required to control the spread of such spam my information.

There is a proliferation of spamming at all online communication mediums i.e. web spam and review spam. In web spam, people boost ranks of their webpages through link farming or manipulating page content [4]. Nowadays, reviews of online products have become an important part of the buying process for people. Review spam is used for presenting a false image of the products. Due to a large number of users of online social networking sites, spammers have extended their targets to OSNs. They can easily access data from these sites through the application of programming interfaces (APIs). So, to tackle the threats of social spam, there is a need of understanding the diffusion dynamics of spammy information. Twitter defines spam as a bulk or aggressive activity that attempts to drive traffic or attention to unrelated accounts, products, or services. Spam detection systems examine the content and behavior of the users to tackle spam problem [5,6]. It is found that 89% of spam accounts have fewer than 10 followers and 17% of spam users exploit hashtags to make their tweets visible in search and trending topics [7]. It is analogous to web spamming where popular and trending search keywords are hijacked for spamming [8]. Spammers post duplicate content and this activity is recorded by using near duplicate detection techniques [9,10]. But these user-based features (i.e. number of followers or followees) do not always work well due to the complex behavior of human beings. Hashtags are used by both spammers and non-spammers for attaching a topic or theme to their tweets. In this paper, we have coupled the existing features (i.e. number of followers, number of followees, and hashtags) with the sentiment and latent topics. These features reveal how spammers frame the content of their tweets for convincing others to retweet (or spread) a piece of information. Sentiment extraction is done by using SentiStrength tool [11] and latent topics are extracted through well-known topic modeling technique (LDA). This analysis clearly reveals the variations of sentiment and topics in spam and non-spam tweets.

1.1. Motivation

The main motivation behind this work is to protect people from misleading information. In the literature, user-based features [3,7] (i.e. number of followers, number of followees) and content-based

CONTACT Mohammad Ahsan ahsan@nith.ac.in, T. P. Sharma teek@nith.ac.in, Computer Science and Engineering Department, National Institute of Technology, Hamirpur, Himachal Pradesh, India

features [9] (i.e. number of links, number of hashtags) have been used for the categorization of spam and non-spam content. Spammers tend to have fewer number of followers and more number of followees [3]. However, the number of followers and followees can be easily manipulated by spammers. On Twitter, there is a probability that users follow back their followers. Spammers exploit this opportunity and follow/unfollow a large number of users to have more number of followers and comparatively fewer number of followees. First, they follow other users in bulk and after being followed back, unfollow them. This bulk following/unfollowing get their job (having more followers than followees) done. It means, users-based features do not always work well. So, we have extended the existing research and used the LDA technique to extract the latent topics from spam/non-spam content. This research is helpful in combating the diffusion of spam information and protecting society from panic situations.

The major contributions of this work are as follows:

- Collecting tweets which serve as the ground truth for analyzing the variation of spam/non-spam information diffusion.
- Analyzing the variation of sentiment and topics in spam/non-spam information.
- Proposing a feature-set which classifies spam/non-spam tweets with a better accuracy than the existing features and predicting the diffusion level of spam information on Twitter.

The remaining paper is organized as follows. In Section 2, we have discussed the literature to provide the insights into the existing methods and approaches. The proposed scheme is described in Section 3. Results are discussed in Section 4. Finally, a brief conclusion is contained in Section 5.

2. Literature review

Social spam has led to a serious threat to the utilities of online social media. It drives attention to unrelated products, services, and malicious links. It is estimated that out of every 200 messages or 21 tweets of social media users, there exists one spam [12]. There is a large thread of research to detect web spam and email spam, but scant attention has been paid for detecting social spam. This section contains a summary of methods (i.e. honeypots based, URLs blacklist, clustering, and classification techniques) which are used by the researchers to detect spams on OSNs i.e. Facebook, Twitter, or Google+. There exist various features that are utilized for detecting spams in online social networks like short URLs [13–15], blacklist URLs [16–18], tweets' similarity [19,20], content-based features (i.e. number of URLs, number of hashtags, number of user-mentions, number of digits etc.) [9,20–22], profile-based features (i.e. number of followers, number of friends, and follower ratio) [19,22], and graph-based features (i.e. local clustering coefficient, density, etc.) [20–22].

2.1. URLs-based methods

Due to the limitations on the number of characters of every tweet, spammers use deceptive URLs in their tweets for spreading malicious information among social media users. If some tweet contains deceptive URLs, sensitive words, or phrases, then it is more likely to be a spam [14]. Shortened URLs and blacklisted URLs are mainly used by the researchers to detect spams on Twitter.

2.1.1. Shortened URLs

URLs are not added to tweets in their original form as they contain many characters. There exist various URL shortening services like Bit.ly and Twitter URL shortener which transform original URLs to a short form. These services provide more space to twitter users to accommodate additional information in their tweets. Shortening of URLs is helpful in better utilization of tweets' length, but at the same time it increases the complexity of spam detection by obfuscating malicious links. So, to utilize the lexical characteristics of original URLs one has to reverse engineer the shortened URLs to their original forms. According to Thomas et al. [23], filtering of URLs is a very expensive process and it may incur a cost of $22,751 for filtering 15.3 M URLs/day for a month. Apart from the cost issue, these methods also fall short in detecting malicious URLs having conditional behaviors [24]. Conditional behavior enables the spammers to redirect investigators to legitimate webpages and normal users to spam links.

2.1.2. Blacklist-URLs

Twitter mainly relies on blacklists to detect spam as there exists many organizations that provide URL-blacklists i.e. Google Safe Browsing [25], SURBL [26], Twitter's Link Service [27], Capture-HPC [28], and Project Honey Pot [29]. The main issue linked with blacklists is time lag – URLs are added to these blacklists after getting clicked by the social media users. A new malicious URL, which is not a part of any blacklist, can prevaricate spam detection. After analyzing 400 M tweets and 25 M URLs, Grier et al. [30] have concluded that Google Safe Browsing blacklists are ineffective for spam detection. To overcome the limitation of URL-based methods, machine learning-based methods are used by the researchers.

2.2. Machine learning-based method

It is one of the highly used methods of spam detection. This method overcomes the lag effect of URL-blacklisting by using content-based, user-based, and network-based features. Numerous researchers have employed different algorithms for spam classification, but there is no set of features that can perform well with every algorithm. So, deciding best-performing algorithm and general features-set is a very tedious task. Spammers keep changing their behaviors to evade spam detection and modeling this behavioral change need the researches to timely update their set of features.

2.2.1. Content-based features

There are various content-based features that are very helpful in twitter spam detection i.e. the number of hashtags, URLs, mentions, digits, and the number of characters of a tweet. It is found that spam tweets tend to have more hashtags, URLs [31], and digits [32] than normal tweets. These features are easy to extract and implement, but prone to artifice by spammers.

2.2.2. Account-based features

Spams are also differentiated from non-spam content by using account-based features like the number of followers, number of followees, the reputation of a user, and account age. It is observed that legitimate users have a large number of followers and more lifecycle than spammers [32]. These features are robust to change, but there exist some exceptions i.e. spammers can have a large number of followers [21] which reduce spam detection accuracy.

2.2.3. Network-based features

Network-based features i.e. local clustering coefficient and bidirectional links are used in the existing research works. Spammers follow a large number of users in the hope of getting followed back and as a result, their bidirectional link ratio usually remains lower than legitimate users [33]. The collection of these features is impractical as there exist millions of Twitter users.

From the literature, it is observed that URLs are commonly used in spam detection. Thomas et al. [23] calculate the ratios of spam and non-spam tweets that have used URL shortening services in their text. These two ratios were used to detect the spam tweets. However, this approach has various weaknesses like high cost of URLs filtering, redirections (only HTTP redirections can be processed), and inability to detect malicious URLs with "conditional" behaviors [24]. Zhang et al. [34] have used the blacklisting technique to detect spams on Twitter. The authors passed blacklisting-based features to SVM classifier and achieved 87.6% F1-measure. The limitation of this approach is time delay in blacklisting a new spam URL. According to Li [35], the average time of blacklisting new spam URLs is four days and Twitter witnesses 90% clicks on the spam URLs within the first two days. This time delay is not suitable for timely detection of spams on Twitter. Chen et al. [32] have used content-based and user-based features for spam detection. This research proposed a feature set of 12 lightweight features which can be easily extracted from tweets' metadata. A dataset that consisted of 600 M tweets was used to examine the performance of six machine learning models: support vector machine, random forest, Naïve Bayes, C4.5 decision tree, Bayes network, and k-nearest neighbor. Alsaffar et al. [36] have also utilized the same feature set for Twitter spam detection. Both machine learning and deep learning (Recurrent Neural Network) models were trained on a dataset of 10000 tweets, where random forest outperformed other models. The size of this dataset is very small to generalize the results. In this paper, we have proposed a new feature-set which contains sentiment, latent topics (spam and non-spam related terms), content-based, and user-based features. This feature-set better captures the variation of spam and non-spam information as it considers the sentiment scores of tweets and latent spam-/non-spam-related terms. We have applied seven machine models on a dataset of 485639 tweets. The results clearly indicate that the proposed features are better in spam detection than the existing features (in [32,36]).

3. Proposed methodology

To understand the diffusion dynamics of spam/non-spam information, 485639 tweets are collected from Twitter. Data collection process is discussed in section 3.1. In the next step, collected tweets are preprocessed to filter out words that are not content-bearing (i.e. prepositions and articles). Afterward, sentiment score and latent topics are extracted from preprocessed data to analyze their variation in spam and non-spam information. Finally, we predict the diffusion of spam information on Twitter and construct a new feature-set which outperforms the existing features in classifying spams and non-spams.

3.1. Data collection

The IDs of spam and non-spam tweets are collected from HSpam14 dataset [37]. This dataset is generated by collecting tweets on the trending topics. Twitter generally provides two APIs for data collection, REST API and streaming API. The data of the past week are fetched through REST API, whereas the streaming API is used for collecting the live tweets. The keywords on trending topics are passed to the streaming API and the tweets containing those keywords are filtered and stored. After collecting 14 million tweets, the annotation process is done through heuristic and finding the near duplicate clusters. In heuristic, it is assumed that most popular hashtags are likely to contribute for spamming. The tweets containing popular hashtags are labeled as spam. HSpam14 dataset provides the tweet IDs and their labels: spam or ham (non-spam). But to study the diffusion of spam/non-spam information there is a need of tweets' text and user-related properties like the number of followers, followees, account age, and content posted. We have overcome this limitation by following the steps of algorithm 1, which takes tweet IDs as input and returns the tweets with complete information i.e. author ID, account creation date, tweet creation date, tweet text, and the retweet count.

Algorithm 1:
Input: $L \leftarrow$ a list of tweet IDs
Step 1: chunk the given input into lists with maximum 100 tweet IDs
Step 2: pass each list $l_i \in L$ to statuses_lookup module
Step 3: store the output of the previous step in a manageable file format (i.e. .json)
Output: tweets objects

In step 1, the input L gets divided into lists of maximum 100 elements due to the limitation of statuses_lookup module [38]. This module only returns up to 100 full tweet objects per request. Step 2 lookups for the tweets corresponding to IDs mentioned in l_i and returns them, if available. On Twitter, users can delete their tweets after posting them and if a tweet is deleted by its author then that would not be available anymore. In step 3, we have stored the tweets objects into.json (JavaScript Object Notation) format. It is a lightweight data interchange format and well suited for large-scale processing. The output of this algorithm is a collection of tweets in their original form.

3.1.1. Structure of raw tweet

{'source': 'a href = https://mobile.twitter.com ' rel = 'nofollow' > Mobile Web (M2), 'in_reply_to_user_id_str': null, 'entities': {'urls': [], 'symbols': [], 'hashtags': [{'text': 'TEAMFOLLOWBACK,' 'indices': [89, 104]}, {'text': 'AUTOFOLLOW,' 'indices': [112, 123]}, {'text': '500aday,' 'indices': [124, 132]}, {'text': 'RT,' 'indices': [133, 136]}], 'screen_name': 'Ifollowsjp,"created_at": 'Thu May 02 06:50:57 + 0000 2013,' 'favorited': false, 'retweeted': false, 'is_quote_status': false, 'geo': null, 'place': null, 'truncated': false, 'in_reply_to_user_id': null, 'text': '#RT 15.50,"retweet_count': 25, 'profile_sidebar_fill_color': 'E4A78E,' 'favourites_count': 21669, 'listed_count': 799, 'friends_': 179535, 'followers_count': 263745, 'profile_text_color': 'BC6A72,' 'geo_enabled': true}

Original tweets contain lots of information that is unrelated to our study (i.e. profile_sidebar_fill_color, profile_text_color, etc.). So, in section 3.2 and section 3.3 (data preprocessing and feature extraction), we have removed these irrelevant features.

3.2. Data preprocessing

The task of data preprocessing depends on the type of content that is going to be dealt. The first step of the preprocessing is to lowercase and breakdown the tweets into small units called tokens. Due to the quirks of social media text (abbreviations, shortenings, spelling mistakes, creative spellings, and hashtags), tokenization is done with TweetTokenizer [39] which is a special tool of Natural Language ToolKit (NLTK) library. In the next step, words are mapped to their normal form (i.e. I'm $\rightarrow$ I am) and tokens of stop words are removed. Stop words are those words which alone are not content-bearing i.e. articles and prepositions. Constructing a custom stoplist is a tough task and sometimes it may adversely affect the validity of results. If the words are removed too aggressively then results get biased towards an intended outcome. So, common stop words are used from NLTK corpus. Finally, extra whitespaces and words occurring only once are removed.

Table 1. Extracted tweet-based and user-based features.

S. No.	Features	Description
	Tweet-based features	
1.	#words	Number of words in the tweet
2.	#digits	Number of numerals in the tweet
3.	digits per word	#digits/#words
4.	#emoticons	Number of emoticons in the tweet
5.	emoticons per word	#emoticons/#words
6.	#hashtags	Number of hashtags in the tweet
7.	#hashtags per word	#hashtags/#words
8.	user_mentions	Number of mentions (@username) in the tweet
9.	user_mentions per word	user_mentions/#words
10.	Sentiment	Sentiment score of the tweet
11.	#urls	Number of URLs in the tweet
12.	#urls per word	#urls/#words
13.	#spam_words	Number of spam related terms in a tweet
14.	#non_spam_words	Number of non-spam related terms in a tweet
15.	retweet_count	Number of times the tweet has been retweeted
	User-based features	
16.	followers_count	Number of users who follow the tweet's author
17.	friends_count	Number of users who are followed by the tweet's author
18.	followers/friends	followers_count/friends_count
19.	reputation	followers_count/(followers_count + friends_count)
20.	statuses_count	Number of tweets issued by the user (including retweets)
21.	tweet_frequency	Number of tweets posted per day
22.	verified	Indicates whether the user's account is verified or not
23.	account_age	Time since the account was created

3.3. Features extraction

The goal of this step is to extract the informative and non-redundant features from the preprocessed corpus. It facilitates the machine learning models in generating a statement of prediction, known as a hypothesis. A set of 23 features is extracted from the collected tweets. These features are discussed in Table 1.

Features like retweet_count, followers_count, friends_count, statuses_count, and verified are directly extracted from the metadata of collected tweets (as shown in section 3.1.1); and extraction of other features (except sentiment) is done by employing regular expressions. The extraction of sentiment is discussed in section 3.3.1. To extract #spam_words and #non_spam_words, LDA model is applied on the corpus of spam/non-spam tweets. This model produced terms related to the spam and non-spam tweets. These terms are utilized to check how many spam and non-spam related terms are present in a tweet. The count of spam and non-spam related terms in a tweet is stored in #spam_words and #non_spam_words, respectively.

3.3.1. Sentiment extraction

The emotional effect a sender wishes to have on the receiver is known as 'sentiment.' In social media, especially Twitter, senders are users who post tweet and receivers are followers. In order to understand the diffusion dynamics of spam and non-spam information, sentiment of these contents is examined. Each tweet is annotated with a sentiment score by using SentiStrength [40]. It handles the quirks of social media data (i.e. informal, short texts, abbreviations and shortenings, spelling mistakes and creative spellings, special strings like hashtags) and employs additional rules for negations and booster words (e.g. very, extremely). SentiStrength reports sentiment scores for the tweets in a range of −4 to +4.

3.3.2. Extraction of latent topics

To extract topics from the corpus of tweets, LDA is applied. It is a topic modeling technique and frequently used to extract latent topics

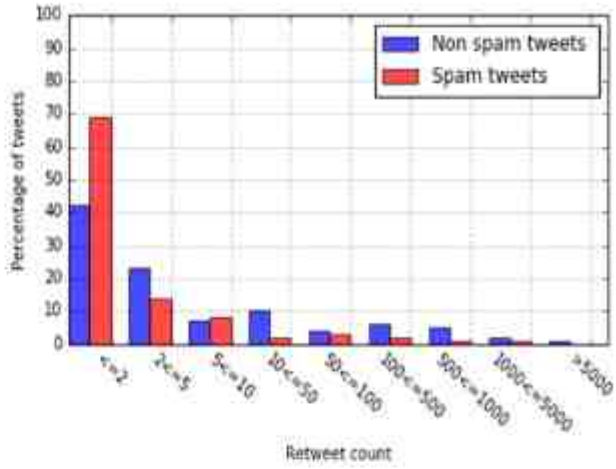

Figure 1. Diffusion of spam and non-spam tweets.

from the documents of a corpus. But, topic modeling techniques are designed for those documents that are long enough to extract robust statistics [41]. When these techniques are applied directly to the messages posted at microblogging sites, they return those topics that are hardly informative and tough to interpret. So, for getting better results of topic modeling techniques, tweets are aggregated on the basis of their diffusibility level. Twitter offers a functionality of 'retweeting' which empowers people to spread information of their choice beyond the followers of original tweet's author. It is a key mechanism by which information gets diffused on Twitter [42]. We have used retweet count as a measure of diffusibility and divided the collection of spam and non-spam tweets into 9 categories. Figure 1 shows the percentage of tweets under each category.

In category '≤ 2,' tweets received up to two retweets. It is observed that 69 percent of spam tweets fall under this category, whereas non-spam tweets are only 41 percent. It means a major portion of spam tweets are identified by users or spam detection systems by the time they receive one or two retweets. But spammers keep updating their spam posting techniques and due to that some spam tweets (around 30 percent in our study) do not get trapped initially and keep spreading among the social media users.

3.3.2.1. Applying LDA on the collected tweets. First, a corpus and dictionary (a built-in data type in python, for holding a key-value pair) are built from the collected tweets, where corpus represents the occurrences of words for each document and dictionary contains IDs for each word and each document. Then, an LDA model is trained by using the created corpus and dictionary. LDA converts the document-terms matrix (corpus) into two matrices: document-topics matrix and topics-terms matrix. Initially, the terms are randomly distributed to topics by using the gamma distribution. The probability density function for this distribution is given in equation 1:

$$p(x) = \frac{x^{T-1}e^{-x/\theta}}{\theta^T \Gamma(T)} \tag{1}$$

where T is the number of topics to be extracted, θ is the scale (1/number of topics), and Γ is the gamma function. Then, this distribution is updated until the convergence point of LDA. If the mean change between prior distribution and updated distribution is less than the given threshold, then it is called a convergence point for LDA. Topics corresponding to the spam and non-spam tweets are returned by this technique, where each topic contains 30 most probable words, as listed in Table 3.

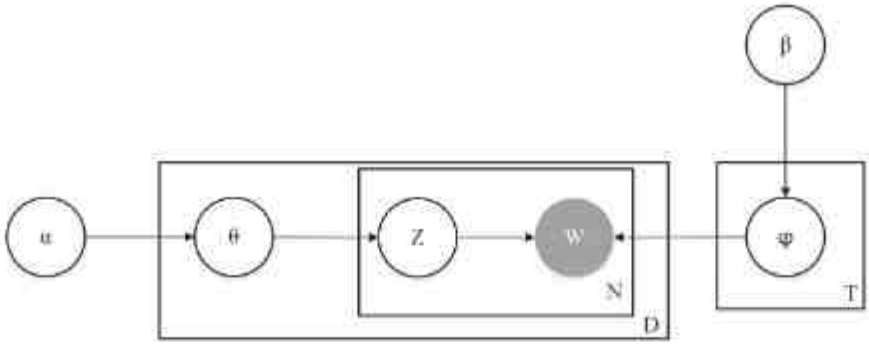

Figure 2. Graphical form of the LDA model.

3.3.2.2. Graphical representation of LDA model. The only observable variable in this model is the words of the given corpus (or tweets' documents), all other variables are hidden, inferred, or latent (i.e. topic). LDA examines these words and determines which topics are present in documents. In Figure 2, shaded and unshaded circles contain observed and latent variables, respectively. The directed edges indicate dependencies between the variables. The rectangles are 'plates' representing the repeated entities. In a graphical model, instead of drawing each repeated variable individually, a plate is used to group variables into a subgraph that repeat together and a number is drawn on the plate to represent the number of repetitions of the subgraph in the plate. The plate with D represents all the variables related to a specific document, including θ_j: the topic distribution for document j. The variables inside this plate are repeated D times, once for each document. The plate with N in the corner represents the variables associated with each of the N_j words in document j that are $Z_{j,t}$ and $W_{j,t}$. These variables are repeated for each word of document j. The rightmost plate with T in the corner represents the variable associated with each topic: φ. This variable is repeated T times, once for each topic.

α and β are the hyperparameters of the LDA model as they are not derived via training and set prior to learning. Here, α controls per-document topic distribution and β is responsible for per-topic word distribution. A high α value means that every document is likely to contain a mixture of most of the topics and not just any single topic specifically, while a low α value means that a document is more likely to be represented by just a few of the topics. Similarly, a high β value means that each topic is likely to contain a mixture of most of the words, not just any word specifically, while a low β value means that a topic may contain a mixture of just a few of the words. A good choice for α and β depends on the number of topics and vocabulary size [11]. We have used $\alpha = 50/T$ and $\beta = 0.01$ because these values work well with many different text corpus [11].

The total probability of this model is given in equation 2:

$$P(W, Z, \theta, \varphi, \alpha, \beta) = \prod_{i=1}^{T} P(\varphi_i; \beta) \prod_{j=1}^{D} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t}|\theta_j) P(W_{j,t}|\varphi_{Z_{j,t}}) \quad (2)$$

where T is the number of topics, D is the number of documents, N is the total number of words in all documents, φ_i is the distribution of words in topic i (vector of probabilities), $\theta_j$is the distribution of topics in document j (vector of probabilities), $Z_{j,t}$ is the topic identity of word t in document j (integer), and $W_{j,t}$ is the identity of word t in document j (integer). For clarity, we have listed the notations and symbols used throughout this paper in Table 2.

In this research work, the LDA model is implemented in python by using Gensim module [43]. It returns topics as a final result where each topic contains a list of words with probabilities of belonging to the extracted topics. The results are discussed in section 4.2.

Table 2. List of symbols.

Symbols	Explanation
L	A list containing tweet IDs
l_i	Index of a sub-list
$p(x)$	Probability density function for gamma distribution
Γ	Gamma function
α	A hyperparameter which controls per-document topic distribution
β	A hyperparameter which is responsible for per-topic word distribution.
T	Number of topics
D	Number of documents
N	Total number of words in all documents
W	Identity of all words in all documents, vector of integers
$W_{j,t}$	Identity of word t in document j, integer
Z	Topic identity of all words in all documents, vector of integers
$Z_{j,t}$	Topic identity of word t in document j, integer
θ_j	Distribution of topics in document j, vector of probabilities
φ_i	Word distribution in topic i, vector of probabilities
n_{samples}	Total number of tweets

3.4. Features analysis

To analyze the discriminating ability of the proposed features, these are plotted by utilizing the cumulative distribution function (CDF). The cumulative distribution function, cdf (x), for a continuous variable represents the fraction of population with a value less than or equal to x. In Figure 3, cdf (x) represents the fraction of spam and non-spam tweets having a feature value less than or equal to x. According to Figure 3(a), spam tweets tend to take in less number of words than non-spam tweets; 80% spam tweets comprise of 16 or less than 16 words and only 20% contains more than 16 words whereas there exist approximately 50% non-spam tweets that have more than 16 words. Spam tweets usually have less number of digits (Figure 3(b)) and more number of emoticons (Figure 3(c)) than non-spam tweets. The authors of spam tweets usually have a new account (Figure 3(d)), less number of followers (Figure 3(e)) and more number of friends (Figure 3(f)) than authors of non-spam tweets. The count of tweets posted in the lifespan of an account is usually higher for non-spams (Figure 3(g)) but spam tweets are posted at a high frequency (Figure 3(h)). Spam tweets likely take in more number of 'urls per word' (Figure 3(i)), 'hashtags per word' (Figure 3(j)), and 'mentions per word' (Figure 3(k)); but receive less number of retweets in comparison of non-spam tweets (Figure 3(l)).

After analyzing the discriminative power of these features, we utilized these to predict the diffusion of spam information on Twitter and classifying spam and non-spam tweets, in section 4.3 and section 4.4, respectively.

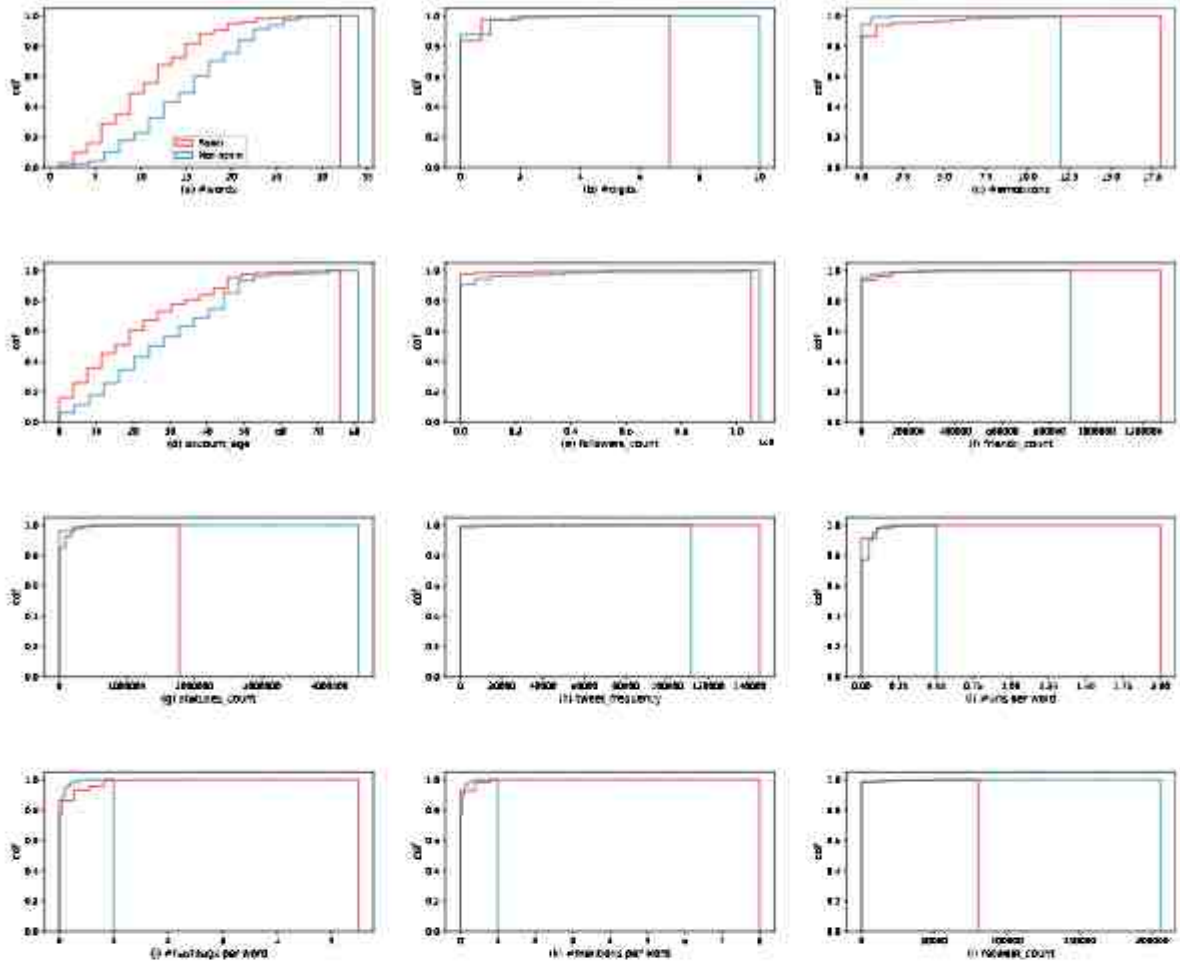

Figure 3. CDFs of the features listed in Table 1.

4. Results

To examine the diffusion dynamics in a better way, in this section, spam/non-spam information are analyzed through sentiment analysis and topic modeling. The performance of the proposed features is also discussed for modeling spam diffusion and classifying tweets as spam/non-spam.

4.1. Sentiment analysis

It helps in getting the picture of people's attitudes toward topics spreading over online social networks. There is a rise in the popularity of Twitter as people share their experiences, news, or latest updates at this platform. So analyzing the sentiment of tweets becomes a key area of research. Due to the limitation of 280 characters on tweet's size, tweets come with irregularities (misspelling, abbreviations, shortenings, and creative spellings) and increase the complexity of sentiment analysis task. SentiStrength is used to extract tweets' sentiment score, as this tool handles the quirks of tweets very well and highly tested by the researchers [44–46].

Figures 4 and 5 show sentiment scores of spam and non-spam tweets, respectively. The peak of both distributions is at 0 (Neutral) as this class contains all those tweets that either have equal number of words with positive and negative sentiment or no sentiment-bearing word. Distribution in Figure 4 is skewed toward the negative sentiment, denoting the fact that spammers tend to spread the tweets with a negative sentiment. It helps the spammers in quickly diverting or catching the attention of people. Tweets with a negative sentiment

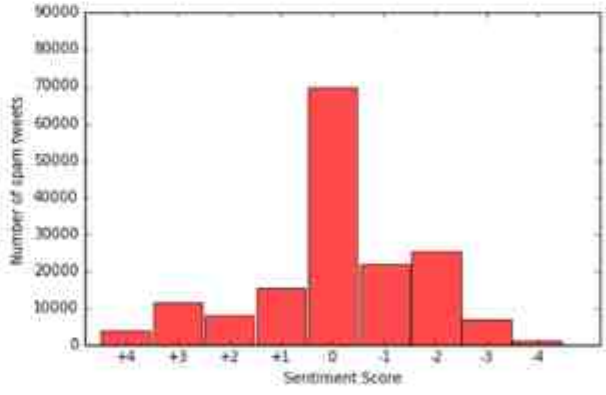

Figure 4. Sentiment of spam tweets.

spread faster than tweets with a positive sentiment [47]. In Figure 5, the distribution is skewed toward a positive sentiment which represents that legitimate users prefer to share the tweets with a positive sentiment.

4.2. Topic modeling

It is a machine learning method for extracting latent topics from the collection of documents. The most common topic model is LDA. But, it is designed for those documents that are long enough to extract the

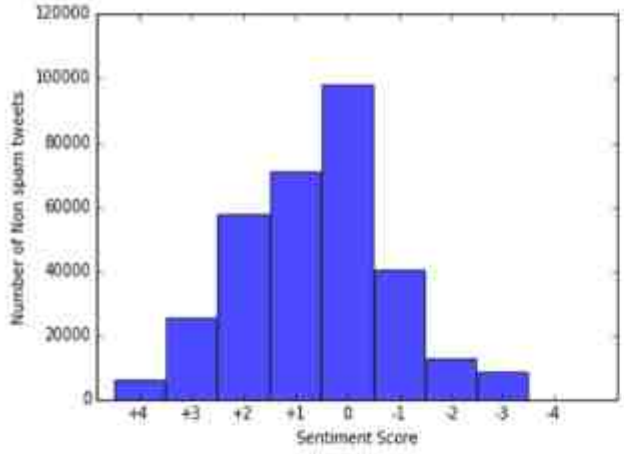

Figure 5. Sentiment of non-spam tweets.

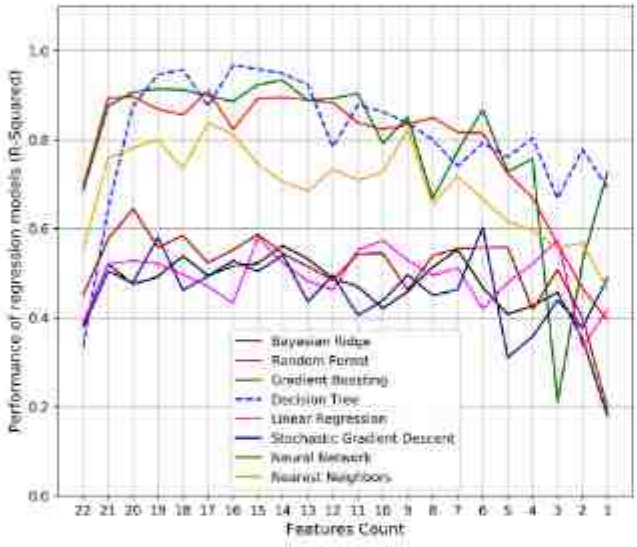

Figure 6. Performance of regression models in predicting spam diffusion on Twitter.

robust statistics [41]. On Twitter, people are bound to post within a limit of 280 characters, so we aggregated the tweets to form documents of decent length (400–500 words). The aggregation of tweets is done on the basis of their diffusibility (retweet count).

Table 3 represents that spammers target the social media users with terms like follow, free, #teamfollowback, #instantfollowback, #followngain, win, #500aday, #follow2befollowed, etc. These terms are claiming to provide certain benefits to the users i.e. 500 bucks a day (#500aday), some free service (free), more followers (#follow2befollowed), etc. As people click these hashtags to get the claimed benefits, they get redirected to some unrelated accounts or products. In order to prevent people from facing the spammy information, these terms need to be addressed properly during designing the rules for information diffusion at a social media platform i.e. Twitter or Facebook.

Terms related to non-spam tweets are free from claims that provide free services and additional followers. It means to control the spread of spam information we have to add new features to the existing spam detection systems. These new features are keywords claiming free services or new followers to the existing social media users.

4.3. Prediction of spam diffusion

For predicting the diffusion of spam information over Twitter, retweet_count is predicted for the spam tweets. Due to the continuous nature of this variable (retweet_count), the following regression models are applied: Bayesian ridge, random forest, gradient boosting, decision tree, linear regression, stochastic gradient descent, neural networks (multi-layer perceptron), and nearest neighbors. Figure 6 represents the performance of all regression models where the decision tree outperformed all other models with 0.96724 R^2 score. The performance of these regression models keeps increasing/decreasing with the change in the number of features. Initially, all models are trained with 22 features, and then, we remove one feature at a time (through algorithm 2) for identifying the best performing feature set. If there is model overfitting due to a feature, then its removal leads to performance gain of the model; otherwise performance gets reduced. After applying algorithm 2, we found that the decision tree gives its best performance on 16 features, random forest on 17, and likewise other models have a feature count on which they perform best.

4.3.1. Performance metric

The performance of these regression models is measured by using R-squared score, the coefficient of determination. This score is calculated using equation 3 and equation 4:

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=1}^{n_{\text{samples}}} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n_{\text{samples}}} (y_i - \bar{y})^2} \tag{3}$$

where $y_i, \hat{y}, \bar{y}$ are target, predicted and average value of retweet count, respectively.

$$\bar{y} = \frac{1}{n_{\text{samples}}} \sum_{i=1}^{n_{\text{samples}}} y_i \tag{4}$$

4.3.2. Selection of best performing features

The objective of feature selection algorithms is to automatically identify features that are most relevant to our problem. It improves the accuracy of the machine learning models by removing irrelevant features and helpful in protecting these models from overfitting. This paper has used a classic feature selection algorithm that is known as Sequential Backward Selection (SBS). It reduces the dimensionality of initial feature-space and helpful in finding out the best performing feature set. The steps of SBS algorithm are represented in algorithm 2.

Algorithm 2: Sequential Backward Selection (SBS)

Step 1: Set $i = n$, where n is the total number of independent features, $X_n \leftarrow$ features listed in Table 1, except retweet_count.

Step 2: Determine the feature x whose removal provides a maximum R^2 score i.e. $\arg\max(R^2(X_n - x)), \forall x \in X_n$

Step 3: Remove feature x from the feature set: $X_n = X_n - x$ and reduce the number of features by 1: $i = i - 1$

Step 4: Stop if i equals to 1, otherwise go to step 2.

It starts with a set of 22 features and in each step, one feature gets removed. The criterion that is used to determine which feature to be removed at a step is performance of the regression model after feature removal. The feature, whose removal provides best R^2 score, is removed at each step. In this manner, we get 22 feature-sets along with their R^2 scores. Table 4 contains feature sets and related performance of decision tree model, ranging from feature-set of 22 features to the feature-set of one feature '[statuses_count].' The best R^2 score

(0.96724) is provided with a feature-set of 16 features which contains 'sentiment' and '#spam_words' as its features. It means apart from other content-based and user-based features, sentiment score, and latent topics also need to be addressed in order to model spam diffusion with a good accuracy.

Apart from modeling the spam diffusion, we have also utilized the proposed features for spam/non-spam classification and compared their performance with research work by Chen et al. [32], in section 4.4.

4.4. Classification of spam/non-spam tweets

There are several URL-based techniques (i.e. shortened URLs or blacklisted URLs) and machine learning-based methods that have been utilized so far by the researchers for identifying spam information over online social networks. Chen et al. [32] have utilized these 12 features for spam/non-spam classification: account_age, no_follower, no_following, no_userfavourites, no_lists, no_tweets, no_retweets, no_hashtag, no_usermention, no_urls, no_char, and no_digits.

In order to check the classification power of these features, we extracted these features from collected tweets and applied seven machine learning classifiers: (i) support vector, (ii) random forest, (iii) naïve-Bayes, (iv) decision tree, (v) logistic regression, (vi) neural network, and (vii) nearest neighbors. Figure 7 represents the accuracy of all classification models where random forest outperformed all other models with 80.651% accuracy. The accuracy score is computed by using Equation (5):

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=1}^{n_{\text{samples}}} 1(\hat{y}_i = y_i), \tag{5}$$

where y and $\hat{y}$ are true and predicted labels of the tweets, respectively. After applying the SBS algorithm, it is found that random forest provided its best classification accuracy on this feature-set: {no_follower, no_following, no_userfavourites, no_lists, no_tweets, no_retweets, no_hashtag, no_usermention, no_urls, no_char, no_digits}.

The performance of the same classification models is also checked on the proposed features (listed in Table 1). Figure 8 shows the classification accuracy of all models where random forest outperformed all others with 91.42% accuracy. The best performance is achieved on this feature set: {#words, account_age, statuses_count, user_mentions, #hashtags per word, reputation, sentiment, retweet_count, #urls per word, #spam_words, digits per word, tweet_frequency, followers_count, #digits, #emoticons, #urls, verified, and emoticons per word}. This best-performing feature-set contains sentiment as well as #spam_words. It represents that analyses of the sentiment of users' posts (or tweets) and latent topics are helpful in detecting spams on Twitter.

Figure 9 shows a comparison of best-performing classifier on the proposed features and features used by Chen et al. [32]. It clearly represents that proposed features gave higher accuracy. There is a 10.77% (80.65% to 91.42%) gain in spam/non-spam classification accuracy. It helps in more efficiently informing the social media users about the spread of spammy information.

5. Conclusion

The ability to computationally annotate the nature (spam/non-spam) of short pieces of text, like tweets, allowed us to investigate the role that information nature plays in the diffusion of such information. In this paper, we have proposed a feature-set to predict the diffusion of spam information and classifying spam/non-spam tweets. Features like URLs, hashtags are examined and it is found that spammers use more URLs in their tweets. After extracting sentiment from tweets,

Table 3. Most probable terms of the topics extracted from spam and non-spam tweets.

	Topics and related terms for spam tweets				Topics and related terms for non-spam tweets			
Topics S.No.	Topic 1	Topic 2	Topic 3	Topic 4	Topic 1	Topic 2	Topic 3	Topic 4
1.	follow	follow	follow	follow	go	go	go	excited
2.	free	go	free	go	people	trouble	someone	@vmagazine
3.	#teamfollowback	free	win	everyone	night	looking	everyone	fans
4.	want	please	#teamfollowback	thanks	know	make	want	big
5.	#rt	download	go	win	music	finds	first	waxer
6.	#retweet	today	retweet	hey	free	usually	day	wax
7.	retweet	may	want	@hattie_james	think	things	time	halfway
8.	#instantfollowback	back	chance	bud	may	back	night	bikini
9.	#tfbjp	music	#instantfollowback	much	make	ever	tour	getting
10.	win	want	#tukfb	enter	dead	every	way	says
11.	#followngain	twitter	#500aday	free	time	someday	really	kids
12.	#follow	help	followtrg	want	day	nothing	people	turns
13.	back	retweet	us	grapes	see	It	grand	chip
14.	go	us	everyone	@kicksonfire	want	decided	yet	go
15.	#music	thanks	#tfbjp	Jordan	first	blame	game	thanks
16.	must	< 3	may	@klxify	much	needed	live	moscow
17.	may	account	#retweet	thank	woman	leave	thanks	tonight
18.	us	everyone	#rt	us	declared	used	done	#giveback
19.	#tfb	win	followback	:d	missing	night	give	job
20.	everyone	fast	read	@radiodisney	never	potter	thanks	sooooo
21.	#500aday	#instantfollowback	#tfb	fight	turns	magical	free	guys
22.	retweets	video	must	< 3	back	sight	ticketing	got
23.	#openfollow	#teamfollowback	#90sbabyfollowtrain	may	went	harry	schedule	much
24.	following	day	tweet	hobi	pa	issue	announcement	dirrrrty
25.	#teamautofollow	It	#follow2befollowed	@louis_tomlinson	alive	eye	seoul	minds
26.	else	#followback	help	today	god	fix	2pm	hurts
27.	fast	check	enter	ash	someone	government	finale	damn
28.	tom	done	back	boys	excited	need	bad	could
29.	chance	Iphone	please	amazing	walked	mom	change	It
30.	#follow2befollowed	rate	thanks	time	take	come	concert	see

Table 4. Optimal feature set selection for decision tree-based regression model.

Step	Feature set	Removed Feature	R^2 scores
1.	{#words, #digits, digits per word, #emoticons, emoticons per word, #hashtags, #hashtags per word, user_mentions, user_mentions per word, sentiment, #urls, #urls per word, #spam_words, #non_spam_words, followers_count, friends_count, followers/friends, reputation, statuses_count, tweet_frequency, verified, account_age}	–	0.33332
2.	{#words, digits per word, #emoticons, emoticons per word, #hashtags, #hashtags per word, user_mentions, user_mentions per word, sentiment, #urls, #urls per word, #spam_words, #non_spam_words, followers_count, friends_count, followers/friends, reputation, statuses_count, tweet_frequency, verified, account_age}	#digits	0.64892
3.	{#words, digits per word, #emoticons, emoticons per word, #hashtags, #hashtags per word, user_mentions, sentiment, #urls, #urls per word, #spam_words, #non_spam_words, followers_count, friends_count, followers/friends, reputation, statuses_count, tweet_frequency, verified, account_age}	user_mentions per word	0.87125
4.	{#words, digits per word, #emoticons, emoticons per word, #hashtags, #hashtags per word, user_mentions, sentiment, #urls, #urls per word, #spam_words, #non_spam_words, followers_count, friends_count, followers/friends, reputation, statuses_count, tweet_frequency, verified}	account_age	0.94584
5.	{digits per word, #emoticons, emoticons per word, #hashtags, #hashtags per word, user_mentions, sentiment, #urls, #urls per word, #spam_words, #non_spam_words, followers_count, friends_count, followers/friends, reputation, statuses_count, tweet_frequency, verified}	#words	0.95739
6.	{digits per word, #emoticons, #hashtags, #hashtags per word, user_mentions, sentiment, #urls, #urls per word, #spam_words, #non_spam_words, followers_count, friends_count, followers/friends, reputation, statuses_count, tweet_frequency, verified}	emoticons per word	0.87495
7.	{digits per word, #emoticons, #hashtags, user_mentions, sentiment, #urls, #urls per word, #spam_words, #non_spam_words, followers_count, friends_count, followers/friends, reputation, statuses_count, tweet_frequency, verified}	#hashtags per word	0.96724
8.	{digits per word, #emoticons, #hashtags, user_mentions, sentiment, #urls, #urls per word, #spam_words, followers_count, friends_count, followers/friends, reputation, statuses_count, tweet_frequency, verified}	#non_spam_words	0.95846
9.	{digits per word, #emoticons, #hashtags, user_mentions, sentiment, #urls, #urls per word, #spam_words, followers_count, followers/friends, reputation, statuses_count, tweet_frequency, verified}	friends_count	0.94828
10.	{digits per word, #emoticons, #hashtags, user_mentions, #urls, #urls per word, #spam_words, followers_count, followers/friends, reputation, statuses_count, tweet_frequency, verified}	sentiment	0.92348
11.	{digits per word, #emoticons, #hashtags, user_mentions, #urls, #urls per word, #spam_words, followers_count, followers/friends, reputation, statuses_count, tweet_frequency}	verified	0.78296
12.	{digits per word, #emoticons, #hashtags, user_mentions, #urls, #urls per word, #spam_words, followers_count, followers/friends, reputation, statuses_count}	tweet_frequency	0.87806
13.	{digits per word, #emoticons, #hashtags, user_mentions, #urls, #urls per word, #spam_words, followers/friends, reputation, statuses_count}	followers_count	0.86207
14.	{digits per word, #emoticons, #hashtags, user_mentions, #urls, #urls per word, followers/friends, reputation, statuses_count}	#spam_words	0.83742
15.	{digits per word, #emoticons, #hashtags, user_mentions, #urls per word, followers/friends, reputation, statuses_count}	#urls	0.7995
16.	{digits per word, #hashtags, user_mentions, #urls per word, followers/friends, reputation, statuses_count}	#emoticons	0.7404
17.	{digits per word, #hashtags, user_mentions, #urls per word, reputation, statuses_count}	followers/friends	0.79195
18.	{digits per word, user_mentions, #urls per word, reputation, statuses_count}	#hashtags	0.76049
19.	{digits per word, #urls per word, reputation, statuses_count}	user_mentions	0.80329
20.	{#urls per word, reputation, statuses_count}	digits per word	0.66784
21.	{reputation, statuses_count}	#urls per word	0.7775
22.	{statuses_count}	reputation	0.69072

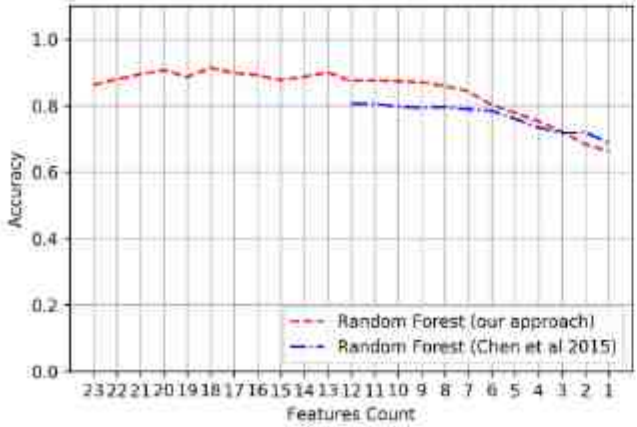

Figure 9. Performance comparison of the proposed features and Chen et al. [32] features.

we observed that spam tweets are skewed toward negative sentiment and the majority of non-spam tweets share the positive sentiment. Next, latent topics are extracted from the documents of spam/non-spam tweets and results reveal the variation in the topics shared by these two types of information (spam and non-spam). The diffusion of spam information is modeled by employing 8 regression models and the best model gives 0.967 R^2 score. Finally, we represent how the proposed feature-set is more accurate in classifying spam and non-spam tweets than the existing features.

The findings of this paper have very practical consequences that are relevant both for economic and social impact. Understanding the dynamics of information diffusion and the effect of sentiment on such phenomena becomes crucial if one wants to craft a policy to effectively hinder the sharing of spam information. In future work, the proposed features and tweets' text can be used together to check the performance of deep learning models for spam detection.

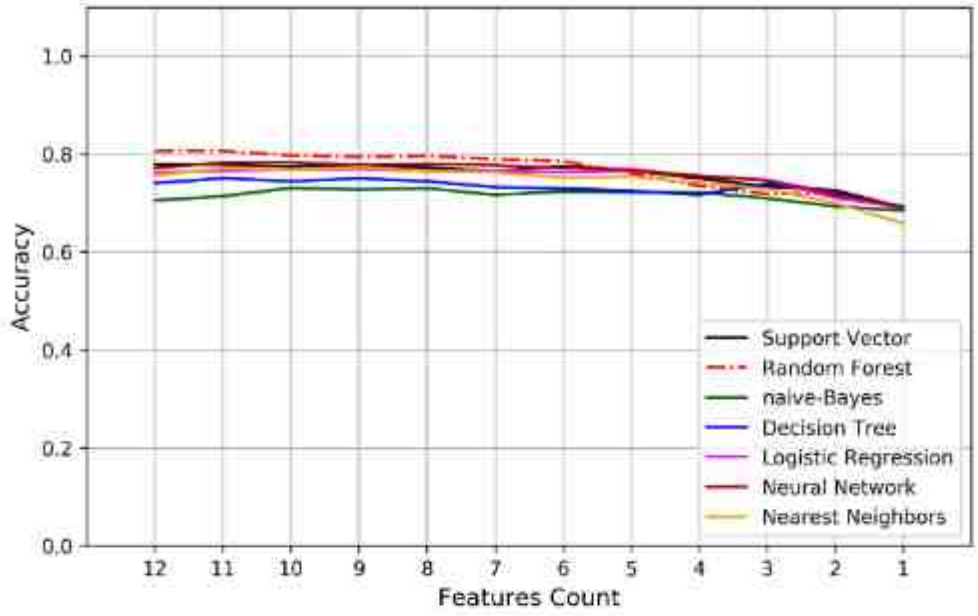

Figure 7. Performance of Spam classification using features proposed by Chen et al. [32].

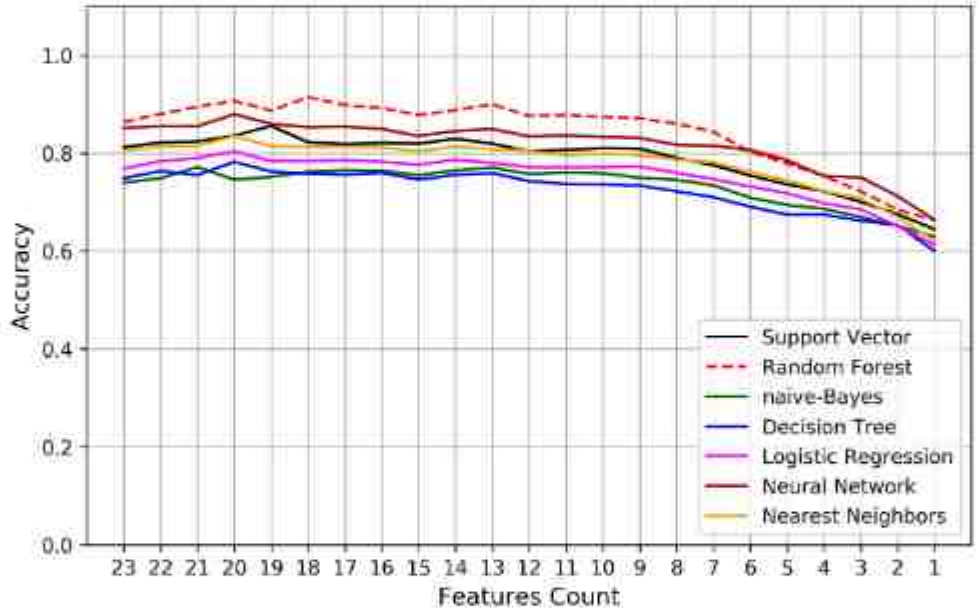

Figure 8. Performance of Spam classification by using proposed features in Table 1 .

Acknowledgements

This work was supported by the Ministry of Electronics and Information Technology (MEITY), Government of India. We are thankful to the editor and anonymous reviewers for helping in improving this manuscript with their valuable comments.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the Ministry of Electronics and Information Technology (MEITY), Government of India.

Notes on contributors

Mohammad Ahsan received his B.Tech. degree in Computer Science and Engineering from Gautam Buddh Technical University, Lucknow in 2012, and the M.Tech. degree in Computer Science and Engineering from the National Institute of Technology, Hamirpur, Himachal Pradesh, India, in 2015. He is currently pursuing Ph.D. in the field of Social Network Analysis from the National Institute of Technology, Hamirpur (H.P.), India. His research interest includes social media mining, machine learning, and natural language processing.

T. P. Sharma is an Associate Professor at National Institute of Technology, Hamirpur, India. He has done his Ph.D. from Indian Institute of Technology, Roorkee, India, in the area of Wireless Sensor Networks. He has published more than 110 high quality research papers in International journals/conferences and has many best paper awards to his credit. Also, he has many book chapters in books of publishers of international repute. His research interest includes Wireless Sensor Networks, Vehicular Area Networks, and IOTs.

ORCID

Mohammad Ahsan http://orcid.org/0000-0002-2619-6529

References

[1] Marx J. Twitter and the 2016 Presidential election. Critique: a worldwide student journal of politics. Normal (IL): Illinois State University; 2017. p. 17–37.

[2] Ventola CL. Social media and health care professionals: benefits, risks, and best practices. Pharm Ther. 2014;39(7):491–499, 520.

[3] Verma M, Sofat S. Techniques to detect spammers in Twitter – a survey. Int J Comp Appl. 2014;85(10):27–32.

[4] Spirin N, Han J. Survey on web spam detection: principles and algorithms. ACM SIGKDD Explor Newsl. 2012;13(2):50–64. DOI:10.1145/2207243.2207252.

[5] Sarma AD, Molla AR, Pandurangan G, et al. Fast distributed PageRank computation. Theor Comput Sci. 2015;561(Part B):113–121. DOI:10.1016/j.tcs.2014.04.003.

[6] Thomas K, Grier C, Song D, et al. Suspended accounts in retrospect: an analysis of Twitter spam. ACM SIGCOMM Conference on Internet Measurement Conference; Berlin; 2011. p. 243–258. DOI:10.1145/2068816.2068840.

[7] Zhu L, Sun A, Choi B. Detecting spam blogs from blog search results. Infor Proc Manag. 2011;47(2):246–262. DOI:10.1016/j.ipm.2010.03.006.

[8] Chowdhury A, Frieder O, Grossman D, et al. Collection statistics for fast duplicate document detection. ACM Trans Inform Syst. 2002;20(2):171–191. DOI:10.1145/506309.506311.

[9] Sedhai S, Sun A. Hspam14: a collection of 14 million tweets for hashtag-oriented spam research. 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, August; ACM, Santiago; 2015, p. 223–232. DOI:10.1145/2766462.2767701.

[10] Wang Z, Josephson WK, Lv Q, et al. Filtering image spam with near-duplicate detection. 4th Conference on Email and Anti-spam (CEAS), Mountain View, CA, USA, August; 2007.

[11] Steyvers M, Griffiths T. Probabilistic topic models. Handb Latent Semant Anal. 2007;427(7):424–440.

[12] Inuwa-Dutse I, Liptrott M, Korkontzelos I. Detection of spam-posting accounts on Twitter. Neurocomputing. 2018;315:496–511. DOI:10.1016/j.neucom.2018.07.044.

[13] Wang D. Analysis and detection of low quality information in social networks. IEEE 30th International Conference on Data Engineering Workshops (ICDEW), March; Chicago (IL); 2014. p. 350–354. DOI:10.1109/ICDEW.2014.6818354.

[14] Chen C, Wen S, Zhang J, et al. Investigating the deceptive information in Twitter spam. Future Gener Comput Syst. 2017;72:319–326. DOI:10.1016/j.future.2016.05.036.

[15] Oliver J, Pajares P, Ke C, et al. An in-depth analysis of abuse on Twitter. Trend Micro. 2014;225:1–22.

[16] Martinez-Romo J, Araujo L. Detecting malicious tweets in trending topics using a statistical analysis of language. Exp Syst Appl. 2013;40(8):2992–3000. DOI:10.1016/j.eswa.2012.12.015.

[17] Zhang X, Li Z, Zhu S, et al. Detecting spam and promoting campaigns in Twitter. ACM Trans Web (TWEB). 2016;10(1). Article No 4. DOI:10.1145/2846102.

[18] Rathore S, Loia V, Park JH. Spamspotter: an efficient spammer detection framework based on intelligent decision support system on Facebook. Appl Soft Comput. 2018;67:920–932. DOI:10.1016/j.asoc.2017.09.032.

[19] Lee K, Eoff BD, Caverlee J. Seven months with the devils: a long-term study of content polluters on Twitter. AAAI 5th International Conference on Weblogs and Social Media (ICWSM), Barcelona, Spain, July; 2011; p. 185–192.

[20] Soliman A, Girdzijauskas S. Adagraph: adaptive graph-based algorithms for spam detection in social networks. Springer International Conference on Networked Systems, May; Cham; 2017. p. 338–354. DOI:10.1007/978-3-319-59647-1_25.

[21] Wang AH. Detecting spam bots in online social networking sites: a machine learning approach. IFIP Annual Conference on Data and Applications Security and Privacy, June; Berlin: Springer; 2010. p. 335–342. DOI:10.1007/978-3-642-13739-6_25.

[22] Kabakus AT, Kara R. A survey of spam detection methods on Twitter. Int J Adv Comp Sci Appl. 2017;8(3):29–38.

[23] Thomas K, Grier C, Ma J, et al. Design and evaluation of a real-time URL spam filtering service. IEEE Symp Secur Priv. 2011: 447–462. DOI:10.1109/SP.2011.25.

[24] Lee S, Kim J. Warningbird: a near real-time detection system for suspicious URLs in Twitter stream. IEEE Trans Dependable Secure Comput. 2013;10(3):183–195. DOI:10.1109/TDSC.2013.3.

[25] Google Safe Browsing APIs. [cited 2019 Feb]. Available from: https://developers.google.com/safe-browsing.

[26] SURBL. URI reputation data. [cited 2019 Feb]. Available from: http://www.surbl.org.

[27] URLs shortening by using Twitter's link service (http://t.co). [cited 2019 Feb]. Available from: https://help.twitter.com/en/using-twitter/url-shortener.

[28] Capture-HPC Client Honeypot/Honeyclient API. [cited 2019 Feb]. Available from: https://projects.honeynet.org/capture-hpc.

[29] Project Honey Pot. [cited 2019 Feb]. Available from: https://www.projecthoneypot.org.

[30] Grier C, Thomas K, Paxson V, et al. @Spam: the underground on 140 characters or less. Proceedings of the 17th ACM Conference on Computer and Communications Security, October; Chicago (IL); 2010. p. 27–37. DOI:10.1145/1866307.1866311.

[31] Benevenuto F, Magno G, Rodrigues T, et al. Detecting spammers on Twitter. Collaboration, Electronic Messaging, Anti-abuse and Spam Conference (CEAS), July; Vol. 6, Redmond, Washington (DC); 2010. p. 12, DOI:10.1.1.297.5340.

[32] Chen C, Zhang J, Xie Y, et al. A performance evaluation of machine learning-based streaming spam tweets detection. IEEE Trans Comput Social Syst. 2015;2(3):65–76. DOI:10.1109/TCSS.2016.2516039.

[33] Yang C, Harkreader R, Gu G. Empirical evaluation and new design for fighting evolving Twitter spammers. IEEE Trans Inf Forensics Secur. 2013;8(8):1280–1293. DOI: 10.1109/TIFS.2013.2267732.

[34] Zhang X, Zhu S, Liang W. Detecting spam and promoting campaigns in the twitter social network. IEEE 12th International Conference on Data Mining, December; Brussels; 2012. p. 1194–1199. DOI:10.1109/ICDM.2012.28.

[35] Li CT, editor. Emerging digital forensics applications for crime detection, prevention, and security. IGI Global. 2013. DOI: 10.4018/978-1-4666-4006-1.

[36] Alsaffar D, Alfahhad A, Alqhtani B, et al. Machine and deep learning algorithms for Twitter spam detection. In International Conference on Advanced Intelligent Systems and Informatics, October; Cham: Springer; 2019. p. 483–491. DOI:10.1007/978-3-030-31129-2_44.

[37] HSpam14 Dataset. [cited 2018 Sep]. Available from: https://www.ntu.edu.sg/home/axsun/datasets.html.

[38] API Reference – Tweepy 3.5.0 Documentation. [cited 2018 Sep]. Available from: http://docs.tweepy.org/en/v3.5.0/api.html.

[39] Tweet Tokenizer package – NLTK 3.2.5 Documentation. [cited 2018 Oct]. Available from: http://www.nltk.org/api/nltk.tokenize.html.

[40] Thelwall M. The heart and soul of the web? Sentiment strength detection in the social web with SentiStrength. Cyberemotions. Understanding complex systems; Cham: Springer; 2017. p. 119–134. DOI:10.1007/978-3-319-43639-5_7.

[41] Alvarez-Melis D, Saveski M. Topic modeling in Twitter: aggregating tweets by conversations. AAAI 10th International Conference on Web and Social Media (ICWSM), Cologne, Germany, March; 2016. p. 519–522.

[42] Huang R, Sun X. Weibo network, information diffusion and implications for collective action in China. Infor Commun Soc. 2014;17(1):86–104. DOI:10.1080/1369118X.2013.853817.

[43] Gensim package – gensim 0.8.6 documentation. [cited 2018 Oct]. Available from: https://media.readthedocs.org/pdf/gensim/stable/gensim.pdf.

[44] Rodrigues RG, das Dores RM, Camilo-Junior CG, et al. SentiHealth-cancer: a sentiment analysis tool to help detecting mood of patients in online social networks. Int J Med Inform. 2016;85(1):80–95. DOI:10.1016/j.ijmedinf.2015.09.007.

[45] Saif H, He Y, Fernandez M, et al. Contextual semantics for sentiment analysis of Twitter. Inform Process Manage. 2016;52(1):5–19. DOI:10.1016/j.ipm.2015.01.005.

[46] Thakor P, Sasi S. Ontology-based sentiment analysis process for social media content. Procedia Comput Sci. 2015;1(53):199–207. DOI:10.1016/j.procs.2015.07.295.

[47] Ferrara E, Yang Z. Quantifying the effect of sentiment on information diffusion in social media. Peer J Comp Sci. 2015;1:e26. DOI:10.7717/peerj-cs.26.