Multimed Tools Appl (2019) 78:3925–3960
DOI 10.1007/s11042-017-5018-x

SMSAD: a framework for spam message and spam account detection

Kayode Sakariyah Adewole[1,2] · **Nor Badrul Anuar**[1] · **Amirrudin Kamsin**[1] · **Arun Kumar Sangaiah**[3]

Received: 2 March 2017 / Revised: 31 May 2017 / Accepted: 6 July 2017 /
Published online: 21 July 2017

Abstract Short message communication media, such as mobile and microblogging social networks, have become attractive platforms for spammers to disseminate unsolicited contents. However, the traditional content-based methods for spam detection degraded in performance due to many factors. For instance, unlike the contents posted on social networks like Facebook and Renren, SMS and microblogging messages have limited size with the presence of many domain specific words, such as idioms and abbreviations. In addition, microblogging messages are very unstructured and noisy. These distinguished characteristics posed challenges to existing email spam detection models for effective spam identification in short message communication media. The state-of-the-art solutions for social spam accounts detection have faced different evasion tactics in the hands of intelligent spammers. In this paper, a unified framework is proposed for both spam message and spam account detection tasks. We utilized four datasets in this study, two of which are from SMS spam message domain and the remaining two from Twitter microblog. To identify a minimal number of features for spam account detection on Twitter, this paper studied bio-inspired evolutionary search method. Using evolutionary search algorithm, a compact model for spam account detection is

✉ Kayode Sakariyah Adewole
adewole.ks@siswa.um.edu.my

✉ Nor Badrul Anuar
badrul@um.edu.my

Amirrudin Kamsin
amir@um.edu.my

Arun Kumar Sangaiah
arunkumarsangaiah@gmail.com

[1] Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

[2] Faculty of Communication and Information Sciences, University of Ilorin, Ilorin, Nigeria

[3] VIT University, Vellore 632014, India

proposed, which is incorporated in the machine learning phase of the unified framework. The results of the various experiments conducted indicate that the proposed framework is promising for detecting both spam message and spam account with a minimal number of features.

Keywords Online social network · Microblog · Spam message · Spam account · Evolutionary computation · Machine learning

1 Introduction

In the past few years, short message communication media, such as mobile and microblogging social networks have become essential part of many people daily routine. Mobile devices offer a plethora of textual communication and provide convenient platforms for users to carry out different activities, such as accessing resources on the Internet, e-banking transactions, entertainments, instant messaging and Short Message Service (SMS). The number of mobile users has dramatically increased in the recent years with an estimate of over 7 billion subscriptions globally [19]. The common form of textual communication between mobile devices is the use of SMS, which utilizes standardized communication protocols to enable mobile phones exchange short text messages with 160 character long [6]. On the other hand, microblogging online social networks, such as Twitter and Sina Weibo, have been utilized for a range of social activities including the posting of interesting contents about past experiences, locating long-lost friends, posting photos and videos, building communities joined by families, acquaintances, and friends. Microblogging social networks have been in existence for almost a decade. For instance, the launch of Twitter in 2006 witnessed a rise in the number of microblogging platforms [2, 44]. The common characteristic of microblogging networks is that they allow users to share short messages usually called microposts or tweets with a maximum of 140 characters. These distinguished characteristics of SMS and microblogging messages forced users to introduce many domain-specific words. As a consequence, the traditional semantic analysis approach for spam detection degraded in performance [6, 24]. The increasing popularity of microblog and mobile communication media has attracted the attention of spammers who utilize the platforms to spread bogus contents [1, 6, 13]. Despite the various benefits offer by mobile and microblogging platforms, they have become the popular media for distributing spam messages [26, 46].

Spamming is a method of spreading bulk unsolicited contents usually for the purpose of advertisements, promoting pornographic websites, fake weight loss, bogus donations, fake news, online job scams, and a host of other malicious intents, which are perpetrated by spammers. The rise in spamming activities on various communication media has long been investigated. For instance, between the year 2009 and 2012, Akismet identified over 25 billion comment spams in Wordpress blogs and the proportion of email spam traffic generated in 2013 was about 69.6% [12]. The problem of spam distribution on communication media has spanned beyond email and blog communication platforms. The increasing rate of mobile SMS spam messages was analyzed in Cloudmark report [6]. This report revealed that the distribution rate of mobile SMS spam varies according to regions. For instance, in the part of Asia, about 30% of mobile messages were represented by spam. An estimate of 400% increase in unique SMS spam campaigns was witnessed in the U.S during the first half of the year 2012 [6]. According to the Nexgate report in 2013, social spam has grown for almost 355% and every seven new social media accounts created contain at least five spammers' accounts [34].

As a result, mobile and social networks are becoming the target for spam distribution. Aside the use of microblogging social network for spreading spam contents, spammer also creates fake profiles to mislead legitimate users. They engage in underground market services where spammers can purchase fake followers to boost their profile reputation. This illegal behavior hinders the reliance on the information generated on microblogging social network and negatively affects the systems that utilize followers and friends' connections to predict user's influence [25].

Unlike email spam corpus with rich contextual information and large public datasets, mobile and microblogging messages are usually shorter, which permit inclusion of entities, such as abbreviations, bad punctuations, shorten URLs, and emoticon symbols. These characteristics degrade the performance of email spam detection filters when utilized to identify spam contents in short message communication media [6, 12]. In addition, the use of traditional content-based analysis using bag of word model have produced low detection accuracy [17]. Thus, majority of the studies on spam message detection has focused on email and webpage spam filtering [32, 36]. Recently, research in short message spam identification has witnessed a growing interest in the research community and several approaches have been studied. For instance, Almeida et al. [6] introduced raw non-encoded SMS spam collection corpus known to be the largest public SMS spam dataset in the literature. The authors proposed several classification models to benchmark the dataset and found that Support vector machine (SVM) outperformed other classifiers. However, the accuracy of this baseline models still need to be improved and further analysis of the proposed corpus is a welcome development. Martinez-Romo and Araujo [31] proposed statistical language based model and content analysis techniques to detect spam message in trending topics on Twitter microblog. Chan et al. [12] studied adversarial attack on short message spam detection filter and proposed a reweight method with a new rescaling function to combat evasion on spam filters. Although the proposed model increases the security level of spam message detection system, however, its classification accuracy on untainted samples drops significantly. El-Alfy and AlHasan [19] proposed a Dendritic Cell Algorithm (DCA) inspired by the danger theory and immune based systems to detect email and SMS spam messages. In this paper, we consider both spam message and spam account detection problems within a single framework in order to provide a more efficient and compact method to combat evasion on spam filters.

Exiting studies on spam accounts detection have utilized different detection approaches [21, 26]. For example, Ghosh et al. [21] applied social network analysis to distribute trust values using both known spammers and legitimate accounts as initial seeds. The algorithm, Collusionrank, assigned trust and untrust values to the neighbor of the selected seeds. The value assigned to each account depicts the strength of trust and for identifying other spammers on the network. Since the number of seeds is very limited taking into consideration the overall size of Twitter microblogging network, the initial score of the original seeds can dilute easily. This may propagate imprecise scores to many accounts on the network, which are less efficient to rank unknown users as spammers or legitimates [29]. Another line of research focused on identifying features for spammer detection, which can be utilized to train machine learning algorithms. For instance, Lee and Kim [26] proposed five name-based features from Twitter account group. The problem with this approach is the evasion of name-based features. For example, spammers can break this detection method using different character combinations to generate account names that mimic the characteristics of the legitimate account. In addition, the use of underground markets to purchase fake followers and tweets further limit the capability of existing solutions that rely on the number of followers and tweets. Hence, it is

important to investigate the different features that can be used to identify spam message and spam account in order to provide a compact and more secure spam detection system.

This paper proposes a unified framework that can detect both spam message and spam account in short message communication media. By exploring five categories of features using bio-inspired evolutionary search algorithm, a compact model for spam account detection in microblogging social network is proposed. The paper further identifies minimal features that can be utilized to detect spam message on both mobile and microblogging platforms. Through rigorous experiments using ten (10) machine learning algorithms, the best classifier for the unified approach is identified. In particular, the contributions of this paper are summarized as follows:

1. Propose a unified framework for spam message and spam account detection (SMSAD), which explores a minimal number of features to provide effective spam filter for short message communication media.
2. Apply bio-inspired evolutionary search algorithm to identify reduced features for spam account detection in Twitter microblogging social network.
3. Introduce a set of unique features to complement the features proposed in the related studies.
4. Train and test ten (10) classification algorithms to identify the best classifier for the proposed unified framework.
5. Propose Random Forest classifier as the best algorithm for spam message detection and LogitBoost classifier as the best algorithm for spam account detection, which are incorporated in the machine learning phase of the unified framework based on the results of the various experiments conducted.

The remainder of this paper is organized as follows. Section 2 discusses related work on spam message and spam account detection. Section 3 presents a detailed discussion of the proposed method. Section 4 highlights the results obtained from the different experiments conducted. Section 5 discusses how the results of the proposed unified spam message and spam account detection framework are compared with the related studies. Finally, Section 6 concludes the paper and highlights future directions.

2 Related work

Research in spam message and spam account detection in communication media has received growing interests in the recent years. Spam message detection studies the textual information posted by spammer using techniques such as natural language processing with machine learning [9, 12, 31]. Majority of the studies in spam message detection focus on content-based analysis and treat textual contents as collection of documents where individual message is preprocessed and represented using vector space model (VSM). VSM is a widely used method for text representation. Each vector identify by VSM is described using bag of word model where a document is represented as the bag of words it contains neglecting grammar and words order. Individual document can further be represented using Boolean occurrence of each word in the document or by counting the frequency of occurrence of each word [17, 48]. A more sophisticated scheme using Term Frequency Inverse Document Frequency (TF-IDF) has been studied to establish the importance of a word in the document [37]. Authors have

proposed Bayesian model for SMS spam classification using content analysis techniques [11, 48]. Yoon et al. [45] combined content analysis and challenge-response to provide hybrid model for mobile spam detection. The content based spam filter first classify message as spam, legitimate or unknown. The unknown message is further authenticated using a challenge-response protocol to determine if the message is sent by human or automated program. El-Alfy and AlHasan [19] introduced a DCA algorithm to improve the performance of anti-spam filters using email and SMS data. Chan et al. [12] investigated the capability of existing spam filters in defending against an adversarial attack. The authors introduced a reweight method with a new rescaling function to prevent an adversarial attack on linear SVM classifier. Although the proposed model increases the security level of spam filter, however, its classification accuracy on untainted samples drops significantly.

Research in spam account detection in social networks utilized three major approaches, which include blacklist, graph-based, and machine learning. The first of its kind blacklist-based analysis on Twitter was investigated by Grier et al. [23]. The authors demonstrated that 8% of 25 million links shared on Twitter point users to phishing, malware, and different scams websites, which are listed on the most popular blacklists. They also found that a large proportions of accounts used for spamming on Twitter were hijacked from legitimate users. A further analysis of the clickstream data of users' activities confirmed that Twitter is a successful platform for distributing spam messages. Grier et al. [23] investigated the effectiveness of using blacklist approach to reduce spamming activities. However, the authors discovered that blacklists method is too slow in detecting new social threats, exposing more than 90% of legitimate users to spam risk. It takes a longer period before a newly identify malicious link is flagged by the popular blacklists. In addition, blacklist based approach is sometimes platform-dependent. For instance, a malicious link caught by Google Safe Browsing may be unidentified by URIBL blacklist, making spam account detection filter depends on many external resources.

In graph-based method, social network is modeled as a network consisting of nodes (users) and edges (connections). The connections between nodes are analyzed in order to detect accounts with unusual characteristics [2]. This method has proved suitable for separating spam account from legitimate ones. For example, Ahmed and Abulaish [4] applied Markov clustering (MCL) algorithm to group a set of profiles as spam and non-spam. The MCL algorithm takes a weighted graph as input and uses random walk approach to assign probabilities to each node on the network. Based on the assigned probabilities, the algorithm is able to cluster set of profiles using Frobenius norm. Ghosh et al. [21] analyzed link farming activities on Twitter and proposed a Collusionrank algorithm, which penalizes users that connect with spammers on the network. This approach discourages the activities of link farming by lowering users score for establishing suspicious connections to malicious accounts. The algorithm assigned trust and untrust values to the neighbor of the accounts chosen as initial seeds. The value assigned to each account depicts the strength of trust and for identifying other spammers on the network. Since the number of seeds is very limited taking into consideration the overall size of OSN, the initial score of the original seeds may propagate imprecise scores, which are less efficient to rank unknown users as spammers or legitimates [29]. While graph-based method provides suitable approach to identify spammers on social network, one of the notable weaknesses lies on the computational complexity when dealing with large social network graph.

Machine learning (ML) approach plays significant roles in spam account detection on microblogging social networks. ML incorporates two main methods: supervised and unsupervised learning. Supervised learning analyzes training samples and generates a classification

model for predicting new user. Unsupervised learning, also known as clustering method, differs in the sense that no labeled data is present during the training stage, and the algorithm learns from the data itself by identifying similarities among the instances. One of the advantages of supervised and unsupervised machine learning approaches is that they provide opportunity to study different features for spammer detection, which are encoded to train machine learning algorithms. However, the major challenge with these methods centers on the evasion tactics posed by spammers to avoid detection.

Using supervised learning, Chu et al. [15] combined both content and behavioral features to distinguish spam from legitimate campaigns using Random Forest algorithm. Aggarwal et al. [3] combined different categories of features based on content and user profile information to build a tool called PhishAri, which is capable of identifying tweets with malicious URLs on Twitter. Martinez-Romo and Araujo [31] combined language and content based features to train SVM classifier. Liu et al. [28] and Zheng et al. [49] studied spam account detection in Sina Weibo microblogging social network. Sina Weibo is the most popular microblogging network in China with more than 500 million users. The authors studied different content and user profile information to train machine learning algorithms. Benevenuto et al. [10] applied SVM algorithm to detect spammers on Twitter. The authors identified spammers' characteristics related to tweet contents and user-behavior to separate spam and legitimate accounts.

In unsupervised ML method, Egele et al. [18] developed COMPA, a system that exploits statistical model and anomaly detection technique to identify compromised accounts on social networks. The system extracts different features from user's messages, such as time of the day, message source, message text, message topic, message link, and direct user interaction. A behavioral model is built for each category of feature and a global threshold value is computed for all the models. Therefore, any new message from the same user that violates these behavioral characteristics is considered malicious. Lee and Kim [26] proposed hierarchical clustering approach to initially group spammers with malicious profile names. They trained Markov chain model with valid account names identified from Twitter. Thus, an account is flagged as malicious if the account name deviates from identified patterns of legitimate account names. In addition to unsupervised learning approach, the researchers trained SVM algorithm using different name-specific features based on the clusters identified by the hierarchical model. The main issue with this approach is that spammers can launch evasion tactics to generate account names that mimic legitimate account. To evade existing spam account detection models that rely on the number of followers and tweets, spammers purchase fake followers and tweets from different underground markets [44]. For instance, a platform such as Intertwitter (http://intertwitter.com/) offered 10,000 fake followers accounts at the rate of $79, giving spammers the opportunity to embed themselves within the network of legitimate users. Fake accounts are now offered in large volumes, varying from thousands to millions [47]. These bogus accounts and their fake links are infringing on the normal social network trust and disrupting the media for effective social communication.

Motivated by the challenges inherent in spam message detection filters due to the limited size in message length and the evasion of features identified for spam account detection, this paper proposes a unified framework that is cable of detecting spam message and spam account in short message communication media using a minimal number of features. To achieve these objectives, the proposed framework exploits a slightly different approach from the traditional contents spam analysis that is based on bag of words model. The subsequent section provides a detail discussion on the proposed method in this paper.

3 Methodology

The proposed unified framework targets spam message and spam account detection using Twitter and mobile data as test samples. Twitter is a microblogging online social networking service, which enables users to post and read short messages usually known as tweets. Tweet can be embedded with entities such as hashtag, mention, and shortened URLs [14]. Users on Twitter microblog utilize hashtag to group tweets according to topics such as the case of #RioOlympics2016, which is a popular topic discussed on Twitter during the 2016 Rio Olympic Games. A topic can be categorized as trending, if it receives many attentions from the users on the network. For example, #JustinBieber is one of the popular topics in 2011 on Twitter. Mention feature uses the "@" symbol to indicate the users who can receive tweet directly on their timelines. Studies have shown that spammers employ mention tool for target attack since the Twitter microblog featured a unidirectional user binding [24]. Although Twitter has introduced features to deactivate unsolicited mention, however, a majority of users on Twitter still utilize default account settings. The visibility of a tweet on the network is increased through a process of re-tweeting. Re-tweeting a user's tweet has been identified as another strategy used by spammers to keep their accounts running [27]. In addition, spammers' accounts exhibit automated posting patterns since there is a need for spammers to get across to a large number of users on the network [14]. Even though Twitter microblogging social network has become an important platform for real-time communication [5], however, it has gone through several cases of abuses in the hands of social spammers. This is evidence in the rules introduced by Twitter to suspended account with abusive behaviors [41].

On the other hand, mobile users can communicate using short text messages, which are delivered by the message center. The proposed framework can be implemented at the message center to provide a central SMS spam message filter. The framework can also be implemented on microblogging social networks to provide a robust classification system. The assumption upon which the proposed framework is based is that spammers will find it difficult to evade both spam message and spam account detection models at the same time. Thus, combining these two classification models will provide efficient spam filter. To reduce the spread of spamming activities on Twitter and mobile communication media, we propose a unified framework shown in Fig. 1.

The input to the framework can originate from either microblogging networks or mobile phone where in the case of microblogging network the user's screen name or ID is provided to the proposed system. The system collects both contents and network information around the user's social connection. Both the content and network data are passed to the processing and feature extraction phase. The system extracts features from five categories of features: user profile, content, network, timing, and automation, which are used to detect spam account on Twitter microblog. In the case of spam message detection on Twitter, the content data is passed to the preprocessing module where the text is represented using a minimal number of features. This representation stage is discussed further in the feature analysis section. In the case of input from mobile platform, the text message is passed to the preprocessing module where it is also represented using a minimal number of features similar to the case of Twitter spam message. The features extracted from the various components are passed to the machine learning predictive models that have been pre-trained for both spam message and spam account detection to identify the class category. During the training phase of the spam

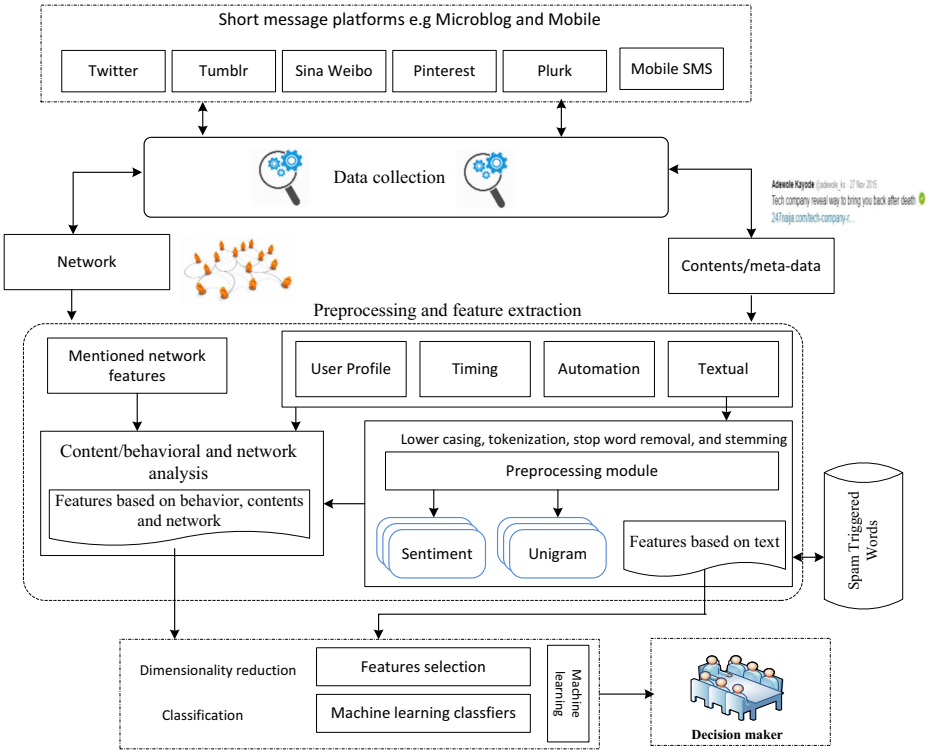

Fig. 1 Proposed unified framework for spam and spammer detection

account detection, the framework incorporates evolutionary search algorithm to provide a minimal number of features for spam account detection model. The results obtained from the different experiments reveal that the proposed unified framework is promising for detecting both spam message and spam account in short message communication media.

3.1 Data collection

In order to evaluate the proposed unified framework for spam message and spam account detection, the data collection stage is divided into two parts. The first part deals with the data used for spam message detection and the second part focuses on how we collect data for spam account detection.

3.1.1 Datasets for spam message detection

Table 1 shows the statistics of the three datasets used to evaluate the proposed framework for spam message detection. The first two datasets, SMS Collection V.1 and SMS Corpus V.0.1 Big, hereafter refer to as Dset1 and Dset2, are publicly available and have been used in ([16, 19] respectively. The third dataset, Twitter Spam Corpus, hereafter refers to as Dset3, is a collection of messages selected from the tweets posted by over 7000 users identified based on the Twitter suspension algorithm. Each of these datasets is discussed further in the subsequent sections.

Table 1 Dataset statistics for spam message detection

Ref ID	Dataset	Type	Spam	Legitimate	Total Sample
Dset1	SMS Collection V.1	SMS	747	4827	5574
Dset2	SMS Corpus V.0.1 Big	SMS	322	1002	1324
Dset3	Twitter Spam Corpus	Microblog	8000	10,000	18,000

SMS spam message datasets The first dataset, Dset1, is a corpus of spam and ham messages publicly available as raw messages at http://www.comp.nus.edu.sg/~rpnlpir/downloads/corpora/smsCorpus/. The corpus contains 747 spam messages and 4827 ham or legitimate messages making a total of 5574 ham and spam messages. This dataset is freely available for research purpose. All the 5574 messages are composed using English language. There are 425 SMS spam messages extracted manually from the Grumbletext website, which is a UK forum at http://www.grumbletext.co.uk/, where mobile phone users can make public claims about SMS spam messages received. A list of 450 SMS legitimate messages collected from Caroline Tag's PhD Theses at http://etheses.bham.ac.uk/253/1/Tagg09PhD.pdf. A collection of 3375 SMS ham messages from the total of 10,000 legitimate messages obtained from the National University of Singapore (NUS). In addition, the corpus contains 1002 SMS ham messages and 322 spam messages extracted from the SMS spam corpus collected by José María Gómez Hidalgo. This corpus is also freely available and can be downloaded at http://www.esp.uem.es/jmgomez/smsspamcorpus/.

The second dataset for SMS spam message detection, Dset2, is a collection of 1002 ham messages and 322 SMS spam corpus in English language, which are collected by José María Gómez Hidalgo and Enrique Puertas Sanz. The corpus is freely available at http://www.esp.uem.es/jmgomez/smsspamcorpus/. This dataset contains a list of 202 legitimate messages from Jon Stevenson and a randomly selected ham messages from NUS SMS corpus, which is a corpus of about 10,000 legitimate messages collected at NUS in Singapore. The raw messages were collected from volunteers who have agreed that the corpus be made available publicly. In addition to the number of legitimate messages, the dataset also contains a collection of 322 SMS spam messages extracted manually from the Grumbletext website.

Twitter spam message dataset To the best of our knowledge, no public dataset is available for Twitter spam message and spam account detection. This is due to the fear of violating user's privacy and the Terms of Use of Twitter API that prevents researchers from sharing tweets data. Therefore, we develop a crawler in python, which takes advantage of the Twitter REST API [42]. We executed eight crawlers using seven Amazon Web Services (AWS) instances and a Desktop computer in our security lab. Amazon AWS is a secure cloud service that provides a platform for computing power, data storage, content delivery, and a host of other functionalities for scalability [8]. The crawling process covers a period of 24 days starting from 20 March to 12 April 2016. The crawlers collected all the tweets posted by the users in our dataset. The statistics of the tweets collected as well as the number of spam accounts identified from the dataset is shown in Table 2. Section 3.1.2 discusses how the spam accounts were identified. Therefore, to build a collection of spam messages, we randomly selected 8000 spam tweets posted by spammers in the dataset. In addition, 10,000 tweets were randomly selected from the accounts identified as legitimate making a total of 18,000 tweets used to test the proposed framework for spam message detection on Twitter microblog as shown in Table 1.

Table 2 Summary of the data collected from Twitter

Dataset item	Number of samples
Ref ID	Dset4
Total Tweets	3,755,367
Total accounts	20,998
Total Hashtag	1,652,405
Total Mention	3,351,656
Total URLs	1,297,288
Mention edges	1,833,086
Total features	69
Spammers identified	3648
Non-Spammers	4000
Total labeled samples	7648

3.1.2 Dataset for Twitter spam account detection

One of the most important stages in developing a reliable classification model is the identification of the labeled samples to be used for both training and validation. Several techniques have been employed in the related studies to achieve this objective. These include honeypot, blacklist, and the Twitter suspension algorithm.

The honeypot approach was originally proposed by Lee et al. [27]. This approach uses some social honeypots to harvest deceptive spam accounts from Twitter and MySpace. The social honeypot logs users' activities, such as content posting patterns, friendship requests, and profile information in the database. All accounts that send unsolicited friend requests are analyzed to find evidence of spamming before they are added to the spammer's list. The goal of this approach is to reduce the challenges of manually identifying spam accounts on social networks. Yang et al. [44] adopted this approach to identify spammers in their dataset. One of the issues with honeypot approach is that the honeypot needs to collect a large number of data for behavioral analysis before the suspected accounts can be categorized as spammers or legitimate. In addition, this approach is slow to acquire a significant proportion of spam accounts. To obtain more spammers for developing a classification model, Yang et al. [44] combined the honeypot and blacklist approaches to detect 2000 spam accounts from their dataset.

The second approach involves the use of the popular blacklists APIs, such as PhishTank, Google Safe Browsing, and URIBL [22, 35, 43]. As discussed in Section 2, the goal of blacklist based approach is to identify accounts that include malicious links in their tweets as flagged by the blacklist APIs. These accounts are marked as spammers and added to the list of labeled samples. This approach was employed to identify spammers in the work of Aggarwal et al. [3] and Yang et al. [44].

The third approach involves reliance on Twitter suspension algorithm [41]. Twitter suspends accounts once it detects abnormal behaviors in the accounts posting patterns, such as spreading malware, pornographic contents, harassment, invitation spam, and other abusive behaviors [1, 41]. Thomas et al. [40] and Hu et al. [24] applied this technique to identify spammers. Since the suspended accounts come from the target microblogging social network, we utilize this labeling approach to identify spammers in this study. We run a batch script in python after a period of six month to identify those accounts that have been suspended by Twitter. In total, the script returns 3648 suspended accounts. We selected a total of 4000

accounts from unsuspended users, totaling 7648 labeled samples as shown in Table 2. Based on the identified spammers and legitimate accounts, we introduced a set of unique features in addition to the previously identified features in the related studies. We construct the mentioned graph network based on the users' tweets. The goal is to extract some graph-based features around the users' mentioned behaviors to classify an account as spammer or legitimate.

3.2 Features analyses

A critical stage in developing effective classification model is the identification of features that can separate one class from another. The use of machine learning approach to identify spam depends on many factors. The most important of these factors is the identification of features that can help distinguish spam from legitimate class. This section discusses the features used for both spam message and spam account detection.

3.2.1 Spam message detection features

This paper adopts a slightly different approach to extract features from the three spam message datasets discussed in the previous section. Motivated by the feature extraction process in [19], we extract eighteen (18) features from the three datasets as shown in Table 3. The reason for adopting this feature extraction method is to provide a compact representation of each of the collections utilized for spam message detection. Unlike VSM and bag of words models where features for spam message detection are represented using the words present in each corpus either by adopting a unigram, bigram or ngram approach, this paper proposes a different method that can provide a more compact representation of the text messages. Each instance of the message in the corpus is passed through a preprocessing module.

The preprocessing module first converts the message to lower case and then proceeds to tokenization phase in order to separate the message by the words it contains using the unigram

Table 3 List of features extracted for spam message detection

Feature name
Frequency of Comm100 spam words
Frequency of ultimate spam words
Frequency of 438 spam words
Frequency of 100 worst spam words
Frequency of combined spam words
Message length in character
Number of words
Frequency of money words
Frequency of money symbols
Number of words in capital
Frequency of function words
Number of special character
Number of emoticon symbol
Number of links
Frequency of phone number
Average number of words
Number of sentence
Sentence ratio

approach. The tokenized collection is processed by removing stop words that will not provide any significance contribution to the final representation. The stop words removal stage is followed with stemming process, which allows us to generate the base or root form of each word in the corpus. For instance, the word "*buying*" is reduced to "*buy*" and the words "*credited*" and "*crediting*" are both reduced to "*credit*". The stemming stage was implemented using the Porter stemming algorithm embedded in NLTK package in python. After completing the preprocessing steps, we extract 18 features discussed as follows:

Frequency of spam triggered words: We collected 257 list of spam triggered words and phrases from Comm100 website at (https://emailmarketing.comm100.com/email-marketing-ebook/spam-words.aspx), such as *urgent*, *call now*, and *free access*. Comm100 is an establishment that provides global enterprise-level customer service and communication solutions. It has been shown that spammers tend to use more spam words and phrases when composing spam message [19]. Similarly, we collected a list of 393 spam words and phrases described at HubSpot blog (http://blog.hubspot.com/blog/tabid/6307/bid/30684/The-Ultimate-List-of-Email-SPAM-Trigger-Words.aspx#sm.00000h35svjkfxez7rh42q7pa3mpp), a list of 438 spam words and phrases at Automational blog (http://blog.automational.com/2016/03/08/spam-trigger-words-to-avoid/), and a list of 100 spam triggered words and phrases at Benchmark blog (http://www.benchmarkemail.com/blogs/detail/the-100-worst-spam-words-and-phrases). We compute the frequency of spam words that appear in each message as a feature. For instance, the frequency of comm100 spam words, frequency of ultimate spam words, frequency of 438 spam words, and frequency of 100 worst spam words presented in Table 3 are calculated from the spam words and phrases collected from Comm100, HubSpot blog, Automational blog, and Benchmark blog respectively. In addition to these spam words and phrases, we selected a list of spam words from each spam message that appear in corpus Dset1, Dset2, and Dset3 respectively. Thus, the frequency of combined spam words is calculated from this list.

Message length in character: This is the length of each message based on the number of characters present in the message.

Number of words: This feature represents the total number of words in the message. For instance, the message *"Act now to win cash price"* contains six (6) words.

Frequency of money words: In some situations, spammer tries to overpower legitimate users by sending unsolicited messages that request for money. For this reason, we collected a list of money words such as thousand, million, and trillion. We compute the frequency of money words that appear in the each message as feature.

Frequency of money symbols: The value of this feature is calculated using regular expression. The regular expression identifies the occurrence of money symbol in each message and then computes the total number of time the money symbol is used in the message.

Number of words in capital: We applied regular expression to compute the number of words that appear in capital letter from each message.

Frequency of function words: Similar to the approach used in [19], the frequency of function words that appear in each message is computed and used as a feature. These are words with little or ambiguous lexical meaning, which are used to express structural relationship with other words in a sentence. A comprehensive

list of function words can be found at (www2.fs.u-bunkyo.ac.jp/~gilner/wordlists.html#functionwords).

Number of special character: The number of special character in each message is computed using regular expression and this value is utilized as a feature.

Number of emoticon symbol: Emoticon symbols like sad, sigh, and happy are mostly used by legitimate users to express mood in a message. Similarly, this features is extracted using regular expression to find the number of emoticon symbol that appear in each message.

Number of links: Studies have shown that spammers can redirect their victims to phishing website where their sensitive information can be collected and subsequently used for malicious purpose [13]. For this reason, the number of links that appear in each message is computed using a regular expression.

Frequency of phone number: This feature represents the number of time a phone number appear in each message. Almeida et al. [6] has shown that a large proportion of SMS spam messages contain phone numbers, which are intentionally added by spammer to lure their victims. This feature is extracted using regular expression.

Average number of words: The average number of words in each message is calculated as the ratio of the number of words to the message length in character.

Number of sentence: This feature represents the total number of sentences present in each message. The sentence tokenizer in python NLTK package is used for this purpose.

Sentence ratio: This is the ratio of the number of sentences to the message length in character.

Figures 2, 3, and 4 show the empirical cumulative distribution function (ECDF) of the length of spam and legitimate messages for each of the corpuses as well as the distribution of words that appear in each class category. These figures show that the length of spam messages is longer than legitimate messages and the number of words in spam messages is more than the legitimate messages. These findings reveal that a majority of spammers leveraged the maximum character length of spam messages to further deceive their victims. As a result, they tend to use more words during message composition.

3.2.2 Spam account detection features

To detect spam accounts on Twitter microblogging network, this paper focuses on five main categories of features. We propose a number of unique features to complement some of the existing features for spam account detection in related studies. This section discusses the five

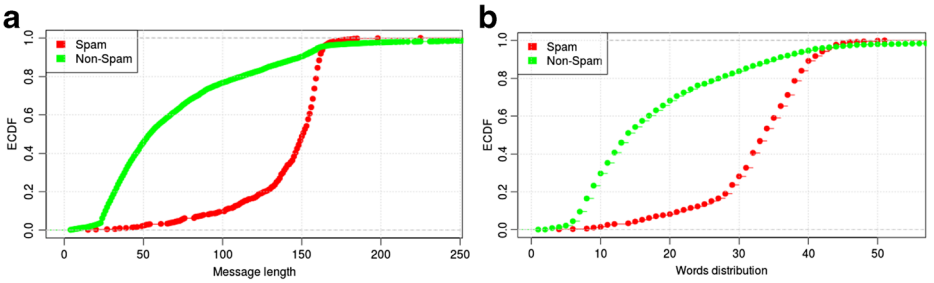

Fig. 2 ECDF of: **a** Message length; **b** Words distribution for Dset1

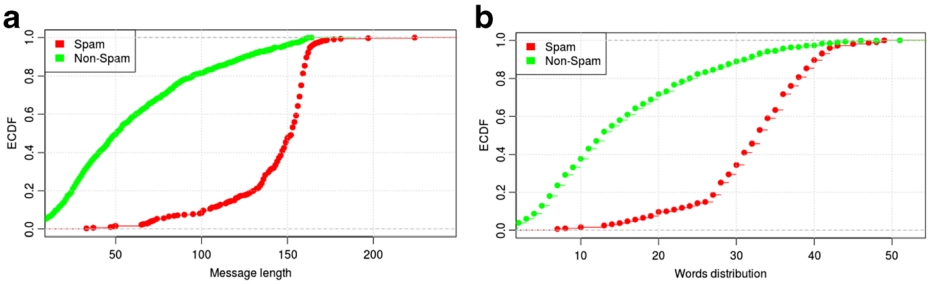

Fig. 3 ECDF of: **a** Message length; **b** Words distribution for Dset2

categories of features utilized to train and test ten (10) classification algorithms. The features are user profile, content, mentioned network, timing, and automation.

- User profile features

  The user profile features have been considered for spam account detection in the work of Yang et al. [44]. The features captured the basic profile information of an account, such as the number of followers, the number of friends, and so on. The values of these features are extracted from the meta-data returned from Twitter microblog. The user profile features capture the behavioral changes of one account to the other based on their profile contents. For instance, Lee and Kim [26] established that the length of the screen name of spammers is usually longer than legitimate users. Table 4 shows the user profile features used in this study with the additional features introduced to complement the existing ones. Additional features introduced in this study are highlighted in bold.

- Content-based features

  Content-based features study the behavioral patterns of social network accounts around the tweets posted by the users. Studies have shown that spammers lure their victims to click malicious links embedded within the tweets. Thus, the accounts of the victims are compromised upon visiting the malicious website [23, 44]. Many social spammers dedicate their efforts posting duplicate tweets. In addition, they employed automated tools to post tweets with very similar semantic [44]. We design a set of statistical features as shown in Table 5 to evaluate the classification results of the selected classifiers. The newly introduced features are highlighted in the table.

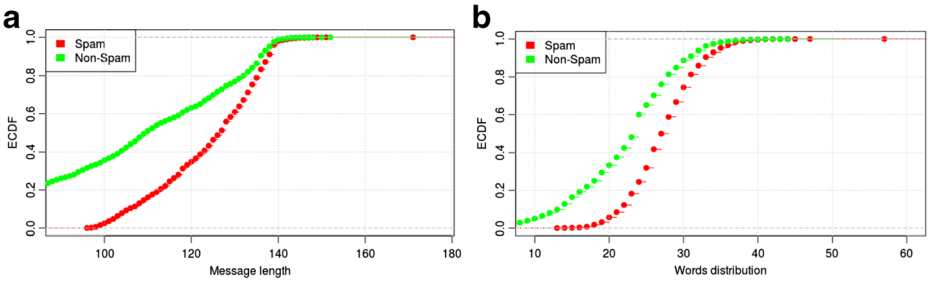

Fig. 4 ECDF of: **a** Message length; **b** Words distribution for Dset3

Table 4 Description of user profile features

Feature name	Description	Reference
Screen name length	The length of the screen name based on characters.	Lee and Kim [26]
User location	The presence or absence of profile location.	**Proposed**
Profile URL	Whether the user includes URL or not in his profile.	**Proposed**
Age in days	Age of the account in days.	Zheng et al. [49]
Followers count	Number of followers of the user.	Yang et al. [44]
Friends count	Number of friends/followees of the user.	Miller et al. [33]
Statuses count	Total statuses of the account.	**Proposed**
Favourites count	Number of tweets the user has favorited.	Miller et al. [33]
User description	Indicating presence or absence of profile description.	Aggarwal et al. [3]
Default profile	When true, indicates that the user has not modified the theme of their profile.	**Proposed**
User Time zone	Indicates presence or absence of time zone.	**Proposed**
Account verified	Indicates whether the account has been verified or not.	Chu et al. [14]
Default profile image	When true, indicates that the user has not changed the default profile egg avatar.	Alsaleh et al. [7]
Listed count	The number of the public lists the user is a member.	Miller et al. [33]
Geo-enabled	Indicates whether or not the user has enabled the possibility of geotagging their tweets.	**Proposed**
Account reputation	Normalized ratio of followers to friends.	Shyni et al. [38]
Follower following ratio	Ration of the number of follower to friends.	Yang et al. [44]
Following follower ratio	Ratio of the number of friends to followers.	Zheng et al. [49]

- Network-based features

These features capture the connections or interactions among users on the Twitter microblog. We modeled users' mentions as a graph $G = (V, E)$, where V represents the vertexes and E the edges corresponding to the mention links between users. If a user u mentions user v in his tweet, we construct an edge $u => v$, which indicates a direct link between u and v. Thus, the graph G is a directed graph that modeled users' mention patterns. We extract a set of graph-based network features as shown in Table 6 from graph G and some network features based on the neighborhood as defined in the work of [44]. The newly introduced features are highlighted in the table.

(a) Local clustering coefficient of mention

We extract local clustering coefficient feature, which is a useful metric to determine how close a vertex's neighbors are to being a clique. A clique is a small group of people with shared interests. As opposed to the work of [44], we focus on extracting the graph-based features around the mentioned network, which enables us to study the mention relationships among users in the dataset. For each vertex in the mentioned graph G, its local clustering score can be computed with Eq. (1), where K_u is the sum of the in-degree and out-degree of the vertex, and e^u is the total number of edges built by all u's neighbors. We noticed that the local clustering coefficient of spammer based on mentioned network is smaller compared to legitimate users. The reason

Table 5 Description of content-based features

Feature name	Description	Reference
Total tweets	Total tweets sent by the user.	Yang et al. [44]
Total hashtag	Total number of hashtag used.	Shyni et al. [38]
Total link	Total number of link posted.	Miller et al. [33]
Total mention	Total number of users mentioned.	Shyni et al. [38]
Total retweet	Total number of retweet.	Miller et al. [33]
Hashtag ratio	Ration of total hashtags to total tweets	Yang et al. [44]
Link ratio	Ratio of total links to total tweets.	Yang et al. [44]
Mention ratio	Ratio of total mention to total tweets.	Yang et al. [44]
Retweet ratio	Ratio of total re-tweet to total tweets.	Yang et al. [44]
Total tweet favorite count	The number of time the user's tweets has been favorited.	**Proposed**
Deviation of hashtag	Population deviation of hashtags.	**Proposed**
Deviation of link	Population deviation of links.	**Proposed**
Deviation of mention	Population deviation of mentions.	**Proposed**
Deviation of re-tweet	Population deviation of retweets.	**Proposed**
Deviation of tweet length	Population deviation of tweet lengths.	**Proposed**
Deviation of hashtag position aggregate	Population deviation of hashtag position aggregate.	**Proposed**
Deviation of link position aggregate	Population deviation of link position aggregate.	**Proposed**
Deviation of mention position aggregate	Population deviation of mention position aggregate.	**Proposed**
Average daily tweet	Ratio of the total tweet to the number of days between first and last tweets posted.	**Proposed**
Average tweet length	Mean of tweet length.	**Proposed**
Average sentiment polarity	Mean of sentiment polarity for each tweet posted.	**Proposed**
Average sentiment subjectivity	Mean of sentiment subjectivity for each tweet posted.	**Proposed**
Average TF-IDF score	Mean of TF-IDF weight of the tweets.	**Proposed**
Popularity ratio	Ratio of the sum of total tweets favorite and total re-tweet to the number of tweets posted.	**Proposed**
Tweet similarity	Similarity of the tweets text using cosine similarity.	Yang et al. [44]
Unique URL ratio	Ratio of unique URLs posted to total tweets.	Yang et al. [44]
Duplicate tweet count	Number of duplicate tweets posted.	Yang et al. [44]
Unique hashtag	Total number of unique hashtags used.	Shyni et al. [38]
Unique mention	Total number of unique mentions.	Shyni et al. [38]
Maximum frequency of hashtag	Maximum value of hashtag frequency.	Shyni et al. [38]
Average frequency of hashtag	Mean of hashtag used.	Shyni et al. [38]
Average frequency of mention	Mean of mentions used.	**Proposed**
Average frequency of URLs	Mean of URLs posted.	Shyni et al. [38]

may be that spammer mentions target users randomly and these accounts may not know each other in reality.

$$LCC(u) = \frac{2|e^u|}{K_u(K_u - 1)} \tag{1}$$

(b) Betweenness centrality of mention

Betweenness is a centrality measure that uses shortest paths to compute the strength of a vertex in the graph. The metric is obtained using Eq. (2), where σ_{st} is

Table 6 Description of network features

Feature name	Description	Reference
Average neighborhood followers	Ratio of sum of the followers of a user's friends to the number of friends of the user.	Yang et al. [44]
Average neighbor tweets	Ratio of the sum of tweets of a user's friend to the number of friends of a user.	Yang et al. [44]
Local clustering coefficient of mention	User's local clustering coefficient based on mention network.	**Proposed**
Betweenness centrality of mention	Betweenness centrality of user based on mention network.	**Proposed**
Bidirectional link of mention	Bidirectional link of user based on mention network.	**Proposed**
Bidirectional link ratio of mention	User's bidirectional link ratio from mention network.	**Proposed**
In-degree of mention	User's In-degree from mention graph.	**Proposed**
Out-degree of mention	User's Out-degree from mention graph.	**Proposed**
Degree reputation of mention	Degree reputation based on mention network.	**Proposed**
Degree centrality of mention	Degree centrality of user from mention graph.	**Proposed**
Closeness centrality of mention	User's closeness centrality based on mention network.	**Proposed**
Eigenvector centrality of mention	Eigenvector centrality of user mention network.	**Proposed**
Pagerank of mention	User's Pagerank from mention graph.	**Proposed**

the total number of shortest paths from node s to t and $\sigma_{st}(u)$ is the number of those paths that pass through the vertex u. n is the total number of nodes in graph G. Similar to the behavior of spammer as identified in the local clustering coefficient of mentioned network, we noticed that betweenness centrality of spammer is smaller than the legitimate users.

$$C_B(u) = \sum_{s \neq u \neq t \in V(G)} \frac{\sigma_{st}(u)}{\sigma_{st}} \tag{2}$$

(c) Bidirectional link of mention

Bidirectional link of mention network defines the total number of links reciprocated by those users mentioned in the tweets. Because spammers randomly mention users in their tweets to launch target attacks, they tend to receive low bidirectional links from the account mentioned as compared to legitimate accounts.

(d) Bidirectional link ratio of mention

Bidirectional link ratio defines the ratio of the number of bidirectional link of a vertex to the total number of out-degree of the vertex. The value is usually low for spammers and high for legitimate users.

(e) In-degree of mention

This feature defines the total number of edges that enters a node. It is computed using Eq. (3). The value is low for spammers and high for legitimate users.

$$d^{in}{}_u = \sum_{[v,u]} G(v,u) \tag{3}$$

(f) Out-degree of mention

This feature represents the total number of edges that leaves a node. It is computed using Eq. (4). The value is high for spammers and low for legitimate accounts. The reason is that spammers tend to mention more users for target attacks than legitimate users.

$$d^{out}{}_u = \sum_{[u,v]} G(u,v) \tag{4}$$

(g) Degree reputation of mention

This is the normalized ratio of the In-degree to the Out-degree of a vertex. The value of degree reputation of mention for spammers is low compared to the degree reputation legitimate users. The feature is computed as shown in Eq. (5).

$$dr(u) = \frac{\left|d^{in}{}_u \cup d^{out}{}_u\right|}{\left|d^{in}{}_u\right|} \tag{5}$$

(h) Degree centrality of mention

This feature defines the sum of the total In-degree and Out-degree of a vertex. The degree centrality of spammers based on the mention network is low compared to legitimate accounts as observed in our analysis. Eq. (6) shows how to compute degree centrality for a vertex.

$$\deg(u) = d^{in}{}_u \cup d^{out}{}_u \tag{6}$$

(i) Closeness centrality of mention

The closeness centrality metric measures the importance of a vertex based on how close a given vertex is to the other vertices in the graph. The most center vertices are important as they can reach the whole network more quickly than non-central vertices. This can be utilized to measure the quality of the connection of a node within the network. Closeness centrality metric can be obtained using Eq. (7), where $d(v,u)$ is the distance between vertices v and u. We notice that the closeness centrality of spammers based on the mention network is low compared to legitimate accounts.

$$C(u) = \frac{1}{n-1} \sum_{v \in V(G)} d(v,u) \tag{7}$$

(j) Eigenvector centrality of mention

This is useful for measuring how the centrality of a node depends on its neighbors' centralities. The metric does not only measure how the vertex is position within the network, but also the quality of the links built with the vertex neighbors. Eq. (8) shows how the eigenvector centrality is computed from the mentioned graph, where $EC(v_j)$ is the eigenvector of the vertex v_j connected to u, $A = [a_{ij}]$ is the adjacency matrix, and λ

is a constant. The EC of one vertex relies on the *EC* of another vertex it is connected to. The *EC(v)* is calculated by finding the eigenvector associated with the highest eigenvalue according to Perron-Frobenious theorem [20]. The i^{th} entry of the vector corresponds to the eigenvector centrality score of i^{th} vertex. The value of eigenvector of spammers is low compared to legitimate users.

$$EC(u_i) = \frac{1}{\lambda} \sum_{j=1}^{N} a_{ij} EC(v_j) \tag{8}$$

(k) PageRank of mention

The Google PageRank is a modified version of eigenvector centrality metric. PageRank of a vertex u relates the PageRank of the vertex it is connected to in the graph. Eq. (9) shows how PageRank score is obtained from the graph, where $d = 0.85$ is the damping factor. N is the total number of vertices considered in the mentioned graph, $PR(v_j)$ is the PageRank of the vertex v_j, $M(u)$ is the set of vertices that link to vertex u. $L(v_j)$ is the number of outbound links of vertex v_j. We found that spammers have low PageRank score compared to legitimate accounts.

$$PR(u) = \frac{1-d}{N} + d\sum_{v_j \in M(u)} \frac{PR(v_j)}{L(v_j)} \tag{9}$$

- Timing-based features

Timing-based features deal with the tweeting rate and following rate of an account. The features examine the posting and following patterns of users on the Twitter microblog. These features have been studied in the work of [38, 44]. We adopted the timing-based features from the related studies. Table 7 shows the description of the two features in this category. Spammer follows a large number of users and generates more tweets than the legitimate users.

- Automation-based features

Similar to the timing-based features, we adopted automation features utilized in [44]. Yang et al. [44] established that spammers resolved to use automation technique for posting tweets due to the high cost of manually maintaining many spam accounts. The technique relies on the use of API to post a large number of spam tweets on the network, thus, spammers' accounts exhibit a high rate of automation. In this regards, a higher API ratio implies automation behavior, which provides an indicator to flag the account as suspicious. Table 8 shows the description of automation-based features adopted in our study.

4 Experimental setup

In this paper, the performance of ten (10) machine learning algorithms is evaluated to identify the best classifier that is suitable for the proposed unified framework. The classifiers selected for the machine learning phase are Random Forest, J48, ADTree, SVM (Sequential minimal

Table 7 Description of timing-based features

Feature name	Description	Reference
Following rate	Ratio of the number of friends to the age of an account.	Yang et al. [44]
Tweeting rate	Ratio of the total number of tweets to the age of the account.	Yang et al. [44]

optimization), Multilayer perceptron (MLP), AdaBoost, Decorate, LogitBoost, Bayes Network, and Random committee. The aim is to find the best performing classifiers that can provide better performance across the datasets used in this study. The parameters configuration of each classifier is shown in Table 9. This study utilized WEKA popular machine learning tool to evaluate the performance of each classifier. The algorithms are grouped into four (4) main categories according to the WEKA machine learning package. The categories include Bayes, function, meta/ensemble, and tree based. We used the same parameters throughout the experiments conducted on each dataset. The implementation of the classification algorithms was carried out on a computer system running Ubuntu 14.04 operating system. The system has a random access memory (RAM) of 20GB and 3.40GHz Intel Core i7 CPU.

4.1 Evaluation measure

This section provides the details of the evaluation metrics employed in this study. Performance metrics provide a practical method to check the efficiency of a model. The classification performance of a mode can be measured in machine learning using a confusion matrix, which is a table that gives the classification performance on how well a classifier is able to separate one class from the other. The general structure of confusion matrix for binary class classification problem is shown in Table 10. In this table, True Positive (TP) and True Negative (TN) referred to the number of correctly classified spam and legitimate instances respectively. False Positive (FP) represents the number of non-spam labeled samples classified as spam, while False Negative (FN) represents the number of spam instances classified as non-spam.

The parameters TP, TN, FP, and FN in this table can be used to derive some standard metrics, such as True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), and False Negative Rate (FNR) as shown in Eqs. 10, 11, 12, and 13 respectively. TPR is also called detection rate (DR), sensitivity or recall, and can be used to indicate the accuracy of a classification model on the labeled samples. A combined metric known as F-measure or F1-score has been widely used to measure the performance of a classification system. This metric is calculated as the harmonic mean of precision and recall as shown in Eq. (15).

Table 8 Description of automation-based features

Feature name	Description	Reference
API ratio	Ratio of the number of tweets sent using API to total number of tweets.	Yang et al. [44]
API URL ratio	Ratio of the number of tweets sent using API that contains URL to the total number of tweets sent using API.	Yang et al. [44]
API tweet similarity	Number of similar tweets sent using API.	Yang et al. [44]

Table 9 Parameter configurations of the selected classifiers

Classifier	Category	Parameter
Random Forest	Tree	bagSizePercent = 100;batchSize = 100;breakTiesRandomly = False;calcOutOfBag = False;debug = False;doNotCheckCapabilities = False;maxDepth = 0;numDecimalPlaces = 2;numExecutionSlots = 1;numFeatures = 0;numIterations = 300;outputOutOfBagComplexityStatistics = False; printClassifiers = False;seed = 1;storeOutOfBagPrediction = False.
J48	Tree	batchSize = 100;binarySplits = False;collapseTree = True;confidenceFactor = 0.25;debug = False;doNotCheckCapabilities = False;doNotMakeSplitPointActualValue = False;minNumObj = 2;numDecimalPlaces = 2;numFolds = 3;reducedErrorPruning = False;saveInstanceData = False;seed = 1;subtreeRaising = True;unpruned = False;useLaplace = False;useMDLcorrection = True.
ADTree	Tree	debug = False;numOfBoostingIterations = 20;randomSeed = 0;saveInstanceData = False;searchPath = Expand all paths.
SVM (SMO)	Function	batchSize = 100;buildCalibrationModels = False;c = 1.0;calibrator = Logistic;checksTurnedOff = False;debug = False;doNotCheckCapabilities = False;epsilon = 1.0E-12;filterType = Normalize;kernel = PolyKernel;numDecimalPlaces = 2;numFolds = −1;randomSeed = 1;toleranceParameter = 0.001.
MLP	Function	GUI = False;autoBuild = True;batchSize = 100;debug = False;decay = False; doNotCheckCapabilities = False;hiddenLayers = a;learningRate = 0.3;momentum = 0.2;nominalToBinaryFilter = True;normalizeAttributes = True;normalizeNumericClass = True;numDecimalPlaces = 2;reset = True;seed = 0;trainingTime = 500;validationSetSize = 0;validationThreshold = 20.
AdaBoost	Meta/ensemble	batchSize = 100;classifier = J48;debug = False; doNotCheckCapabilities = False;numDecimalPlaces = 2;numIterations = 10;seed = 1;useResampling = False;weightThreshold = 100.
Decorate	Meta/ensemble	artificialSize = 2.0;batchSize = 100;classifier = J48;debug = False;desiredSize = 15; doNotCheckCapabilities = False;numDecimalPlaces = 2;numIterations = 60;seed = 1.
LogitBoost	Meta/ensemble	ZMax = 3.0;classifier = RandomTree;debug = False; doNotCheckCapabilities = False;likelihoodThreshold = −1.7977E308;numDecimalPlaces = 2;numIterations = 10;numThreads = 1;poolSize = 1;seed = 1;shrinkage = 1.0;useResampling = False;weightThreshold = 100.
BayesNet	Bayes	batchSize = 100;debug = False; doNotCheckCapabilities = False;estimator = SimpleEstimator;numDecimalPlaces = 2;searchAlgorithm = K2;useADTree = False.
Random committee	Meta/ensemble	batchSize = 100;classifier = RandomTree;debug = False;numDecimalPlaces = 2;numExecutionSlots = 1;numIterations = 10;seed = 1.

Table 10 Confusion matrix for a binary class problem (spam and non-spam)

		Predicted Class	
		Class = Spam	Class = Non-spam
Actual Class	Class = Spam	TP	FN
	Class = Non-spam	FP	TN

$$TPR/DR/Sensitivity/\mathrm{Re}call(R) = \frac{TP}{TP + FN} \tag{10}$$

$$TNR/Specificity = \frac{TN}{TN + FP} \tag{11}$$

$$FPR = \frac{FP}{TN + FP} = 1 - \frac{TN}{TN + FP} \tag{12}$$

$$FNR = \frac{FN}{TP + FN} \tag{13}$$

$$\mathrm{Pr}ecision(P) = \frac{TP}{TP + FP} \tag{14}$$

$$F\text{-}measure = \frac{2PR}{(P + R)} \tag{15}$$

4.2 Results and discussions

We conduct different experiments to evaluate the performance of the proposed unified framework. As discussed earlier, the performance of ten (10) machine learning algorithms were examined on the four datasets. The first experiment evaluates the performance of the classifiers for SMS spam message detection. The second experiment deals with microblog spam message detection. Eighteen (18) features extracted for spam message identification were used for both SMS and microblog spam message detection tasks. The third experiment evaluates the performance of the selected classifiers for spam account detection, and finally, the fourth experiment used evolutionary bio-inspired algorithm to find a number of minimal features for spam account detection in Twitter microblogging social network.

4.2.1 SMS spam message detection

The performance of the selected classification algorithms are examined on the two SMS spam datasets, Dset1 and Dset2, using 18 features discussed in Section 3.2.1. This experiment is based on 10-fold cross-validation training method. In 10-fold cross-validation, the labeled samples are divided into 10 subsets of equal size. In each round of the training, one out of 10 subsets is held as the testing set to validate the classifier, while the remaining nine subsets are used to train the classification algorithm. Random Forest classifier achieves the best results for

Table 11 Classification results with Dset1

Classifiers	10-fold cross-validation					
	DR	FPR	Precision	Recall	F1	AUC-ROC
Random Forest	**0.992**	**0.048**	**0.992**	**0.992**	**0.991**	**0.997**
J48	0.988	0.044	0.988	0.988	0.988	0.960
ADTree	0.987	0.054	0.987	0.987	0.987	0.996
SVM (SMO)	0.987	0.072	0.987	0.987	0.987	0.957
MLP	0.991	0.038	0.991	0.991	0.991	0.996
AdaBoost	0.980	0.064	0.980	0.980	0.980	0.993
Decorate	0.989	0.053	0.988	0.989	0.988	0.992
LogitBoost	0.982	0.085	0.981	0.982	0.981	0.995
BayesNet	0.961	0.066	0.964	0.961	0.962	0.983
Random committee	0.989	0.054	0.989	0.989	0.989	0.994

the two experiments on SMS spam message detection. As shown in Table 11 and Fig. 5, Random Forest produces the best accuracy, F-measure, and AUC-ROC of 0.992, 0.991, and 0.997 respectively (see the highlighted row). The least performed classifier on Dset1 is Bayesian network.

Similarly, in Table 12 and Fig. 6, Random Forest produces accuracy, F-measure, and AUC-ROC of 0.991, 0.991, and 0.999 respectively (see the highlighted row). As observed in the previous results on Dset1, Bayesian network also achieves the least accuracy on Dset2.

4.2.2 Microblog spam message detection

This section presents experiment to evaluate the performance of the selected classifiers for microblog spam message detection. Table 13 shows the results obtained from this experiment. As observed in SMS spam detection, Random Forest also outperformed other classifiers in this experiment. Although the accuracy of this result is lower than the results obtained with SMS spam datasets, this is as a result of the limited size of the tweet contents and domain specific characteristics. Twitter permits tweets length of 140 characters and a majority of the tweets posted on Twitter include many abbreviations. However, our approach is promising when applied on the Twitter spam dataset, producing accuracy of 0.932 and AUC-ROC of 0.983

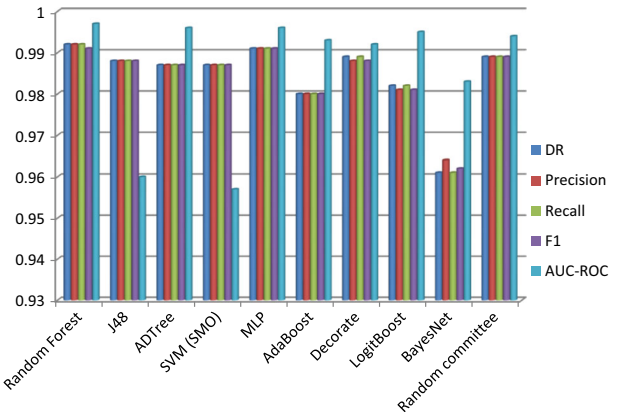

Fig. 5 Performance of the selected classifiers using Dset1

Table 12 Classification results with Dset2

Classifiers	10-fold cross-validation					
	DR	FPR	Precision	Recall	F1	AUC-ROC
Random Forest	**0.991**	**0.018**	**0.991**	**0.991**	**0.991**	**0.999**
J48	0.980	0.047	0.980	0.980	0.979	0.965
ADTree	0.982	0.033	0.982	0.982	0.982	0.997
SVM (SMO)	0.986	0.043	0.986	0.986	0.986	0.972
MLP	0.986	0.030	0.986	0.986	0.986	0.997
AdaBoost	0.980	0.051	0.980	0.980	0.979	0.997
Decorate	0.989	0.025	0.989	0.989	0.989	0.999
LogitBoost	0.984	0.035	0.984	0.984	0.984	0.998
BayesNet	0.960	0.044	0.960	0.960	0.960	0.993
Random committee	0.987	0.025	0.987	0.987	0.987	0.999

using Random Forest algorithm as highlighted in the table. In this experiment, the least performed classifier is Bayesian network in terms of detection rate (DR) and F-measure.

4.2.3 Microblog spam account detection

To detect spam account in Twitter microblogging social network, we first conduct an experiment using all the 69 features extracted as discussed in the previous section. The result of this experiment shows that Random Forest classifier produces the best result achieving accuracy of 0.932 and AUC-ROC of 0.977 as shown in Table 14 (see the highlighted row). This result is followed by Decorate and LogitBoost ensemble classifiers with each achieving F-measure of 0.929 and AUC-ROC above 0.970. This result indicates that the proposed method can achieve AUC-ROC above 0.970, showing the suitability of the framework for spam account detection on Twitter.

4.2.4 Evolutionary search method for feature reduction

Feature reduction is an important aspect of machine learning as it helps to reduce the number of features for classification and at the same time reduces classifier's processing time. This

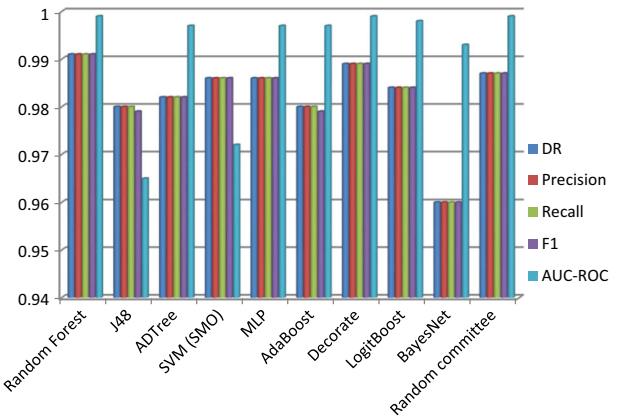

Fig. 6 Performance of the selected classifiers using Dset2

Table 13 Classification results with Dset3

Classifiers	10-fold cross-validation					
	DR	FPR	Precision	Recall	F1	AUC-ROC
Random Forest	**0.932**	**0.070**	**0.932**	**0.932**	**0.932**	**0.983**
J48	0.915	0.086	0.915	0.915	0.915	0.934
ADTree	0.892	0.107	0.893	0.892	0.892	0.963
SVM (SMO)	0.879	0.125	0.879	0.879	0.879	0.877
MLP	0.911	0.090	0.911	0.911	0.911	0.966
AdaBoost	0.920	0.081	0.920	0.920	0.920	0.974
Decorate	0.922	0.079	0.922	0.922	0.922	0.970
LogitBoost	0.924	0.077	0.924	0.924	0.924	0.974
BayesNet	0.842	0.155	0.845	0.842	0.843	0.923
Random committee	0.925	0.078	0.925	0.925	0.925	0.975

method chooses a subset of features among the set of features to determine their discriminative power in distinguishing spam accounts from legitimate users. Since the goal is to develop a compact model for simultaneous detection of spam message and spam account in Twitter microblog, we investigate the applicability of evolutionary bio-inspired algorithm to produce a reduced number of features for spam account detection.

Evolutionary algorithm (EA) is a generic meta-heuristic optimization search approach that concurrently explores numerous points in a search space, and navigates the search space stochastically in order to prevent the search exploration from being trapped at the local maxima [30]. EA utilizes biologically inspired evolution mechanisms, such as recombination, mutation, fitness, and selection. The generic structure of EA algorithm is described as follows:

Step 1: **Initialization**

For time $t = 0$, initialize a population $P(t)$ such that $P(t) = (x_1^t, x_2^t, \ldots x_n^t)$. These are the initial points, which the EA will use to explore the search space. In the case of feature selection, the population corresponds to the different features subsets selected from the original features.

Table 14 Classification results for microblog spam account detection using 69 features

Classifiers	10-fold cross-validation					
	DR	FPR	Precision	Recall	F1	AUC-ROC
Random Forest	**0.932**	**0.071**	**0.933**	**0.932**	**0.932**	**0.977**
J48	0.926	0.078	0.929	0.926	0.926	0.965
ADTree	0.925	0.080	0.929	0.925	0.925	0.974
SVM (SMO)	0.878	0.122	0.879	0.878	0.878	0.878
MLP	0.913	0.091	0.916	0.913	0.913	0.970
AdaBoost	0.918	0.083	0.918	0.918	0.918	0.969
Decorate	0.929	0.074	0.931	0.929	0.929	0.974
LogitBoost	0.929	0.073	0.930	0.929	0.929	0.977
BayesNet	0.881	0.119	0.881	0.881	0.881	0.944
Random committee	0.925	0.077	0.926	0.925	0.925	0.974

Step 2: **Evaluation**

At this stage, each solution in the initial population is evaluated by measuring its fitness.

Step 3: **Selection**

This step creates a new population by stochastically selecting individuals from *P(t)*.

Step 4: **Evolution**

At this stage, the algorithm transforms some members of the new population created in Step 3 using genetic operators, such as crossover and mutation, to form new solutions.

Step 5: **Testing for termination**

Step 2 to 4 are repeated until the termination condition is satisfied. The EA algorithm may terminate if a given number of iterations is reached, a particular fitness value has been achieved, or when the algorithm converges to a near-optimal solution. Figure 7 shows the parameters configuration of the EA algorithm used to perform the feature reduction.

To achieve the objective of reducing the number of features for spam account detection, we first apply the popular Chi-squared test feature selection evaluator to select 60 features using ranker search method. Chi-squared statistics (χ^2) tests the independence of two events, *A* and *B* where the independence is defined as $P(AB) = P(A)P(B)$ or $P(A|B) = P(A)$ and $P(B|A) = P(B)$. In the case of feature selection, the algorithm assumes that the two events are the occurrence of feature and class. The features are ranked using Eq. (16):

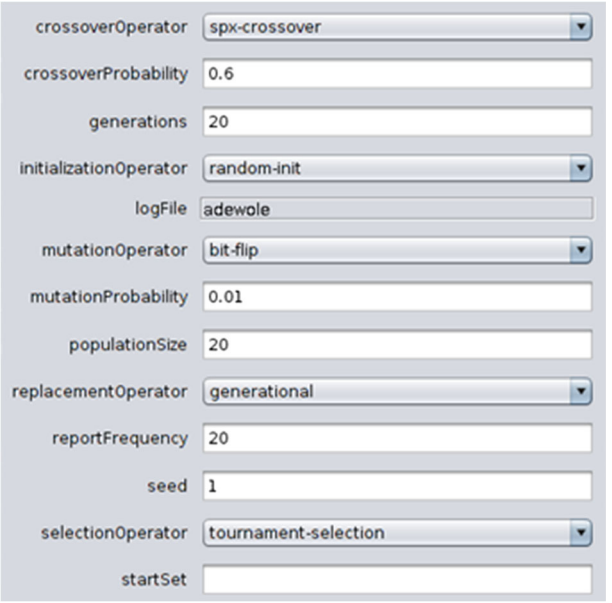

Fig. 7 Parameters configuration for evolutionary algorithm

Table 15 Eighteen (18) features selected by evolutionary search algorithm

Feature name
Age in days
User Time zone
Listed count
User location
Following rate
Average TF-IDF score
Tweet similarity
Follower following ratio
Default profile
Average sentiment subjectivity
In-degree of mention
Total tweet favorite count
Popularity ratio
Profile URL
Local clustering coefficient of mention
Deviation of link
Bidirectional link of mention
Favourites count

$$\chi^2(D,f,c) = \sum_{e_f \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{\left(N_{e_f e_c} - E_{e_f e_c}\right)^2}{E_{e_f e_c}} \tag{16}$$

where N is the observed frequency in D and E is the expected frequency. For a detailed discussion on Chi-squared test statistics for features selection, we refer the reader to [39]. The EA algorithm is executed using the 60 features selected by Chi-Squared. Table 15 shows the eighteen (18) features obtained by the evolutionary algorithm.

After reducing the 69 features to 18 features using a combination of Chi-square test feature selection and evolutionary search algorithm, the next step is to evaluate the reduced data on the selected classifiers. The result of this experiment is shown in Table 16 and Fig. 8. Although the performance of Random Forest classifier drops slightly, however, out of the ten (10) classifier selected in our study, the results of seven (7) classifiers improved using only the 18 features identified by the evolutionary search algorithm. These results are highlighted in bold. Using the bio-inspired evolutionary search method, LogitBoost classifier achieve a result close to

Table 16 Classification results with reduced features

Classifiers	10-fold cross-validation					
	DR	FPR	Precision	Recall	F1	AUC-ROC
Random Forest	0.928	0.074	0.929	0.928	0.928	0.977
J48	0.924	0.080	0.927	0.924	0.924	0.965
ADTree	**0.926**	**0.079**	**0.928**	**0.926**	**0.925**	**0.974**
SVM (SMO)	**0.883**	**0.117**	**0.883**	**0.883**	**0.883**	**0.883**
MLP	**0.914**	**0.090**	**0.917**	**0.914**	**0.914**	**0.971**
AdaBoost	**0.926**	**0.076**	**0.927**	**0.926**	**0.926**	**0.962**
Decorate	0.924	0.078	0.925	0.924	0.924	0.973
LogitBoost	**0.930**	**0.072**	**0.931**	**0.930**	**0.930**	**0.977**
BayesNet	**0.889**	**0.111**	**0.889**	**0.889**	**0.889**	**0.948**
Random committee	**0.928**	**0.076**	**0.931**	**0.928**	**0.928**	**0.973**

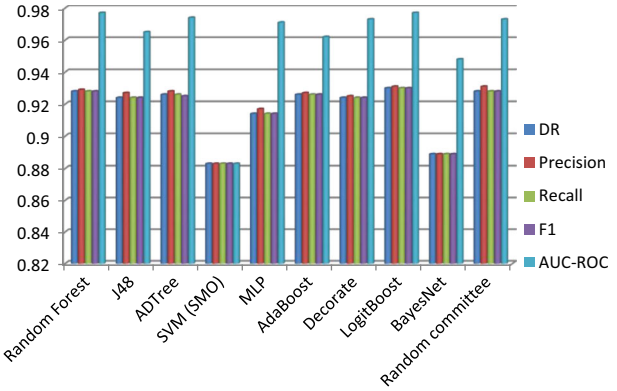

Fig. 8 Performance of the selected classifiers on the reduced features by EA algorithm

Random Forest when all the 69 features were used. LogitBoost produced F-measure of 0.930 and AUC-ROC of 0.977 after reducing the dimensionality of the data. With this result, a compact model for spam account detection is achieved with LogitBoost or Random Forest incorporated in the machine learning phase of the proposed unified framework.

5 Model evaluation

This section provides a discussion on how the results obtained for the proposed unified framework are evaluated and compared with the related studies.

5.1 Performance of SMS spam message detection models

The confusion matrices obtained for SMS spam message detection is shown in Table 17. This table indicates that based on the dataset Dset1 (SMS Collection V.1), the proposed Random Forest model is able to detect 707 messages as spam out of 747 total spam messages, achieving TPR of 94.65% using the 18 features. For non-spam message identification, the model detects 4817 messages as non-spam out of the total 4827 legitimate messages leading to TNR 99.79%. Based on the dataset Dset2 (SMS Corpus V.0.1 Big), the Random Forest model detects 314 messages as spam out of 322 total messages, which produces TPR of 97.52%. On the same dataset, 998 non-spam messages were correctly identified out of the total 1002 messages, producing a TNR of 99.60%. This result further demonstrates the capability of the proposed SMS spam detection models in classifying spam from legitimate messages.

Table 17 Confusion matrices for Random Forest classifiers based on SMS spam detection

Actual	Predicted (Dset1: SMS Collection V.1)		Predicted (Dset2: SMS Corpus V.0.1 Big)	
	Spam	Non-spam	Spam	Non-spam
Spam	707	40	314	8
Non-spam	10	4817	4	998

Table 18 Confusion matrices for Random Forest classifiers based on Twitter spam message and spam account detection

Actual	Predicted (Dset3: Spam message)		Predicted (Dset4: Spam account)	
	Spam	Non-spam	Spam	Non-spam
Spam	7399	601	3288	360
Non-spam	629	9371	162	3838

5.2 Performance of twitter spam message and spam account detection models

To evaluate the performance of the proposed Random Forest models in distinguishing spam from non-spam accounts, we constructed confusion matrices shown in Table 18. With reference to dataset Dset3, the proposed Random Forest correctly identified 7399 spam messages out of 8000 total spam messages, which produces a TPR of 92.49%. On the other hand, 9371 messages were detected as legitimate out of 10,000 messages, producing a TNR of 93.71%. However, considering the performance of the proposed Random Forest with 69 features on the dataset Dset4 (Spam account), the model correctly identified 3288 accounts as spam leading to a TPR of 90.13%. At the same time, a TNR of 95.95% is achieved based on legitimate account detection. This analysis indicates that the proposed models are promising to separate spam message and spam account on Twitter microblog.

5.3 Comparison of spam message detection models

In the case of SMS spam message detection; we compare the result of the Random Forest classifier with the related studies. El-Alfy and AlHasan [19] and Almeida et al. [6] both implemented their approaches on Dset1 (SMS Collection V.1). On this dataset, the proposed model improves in precision and F-measure when compare with El-Alfy and AlHasan [19], although their method slightly outperformed our model in terms of detection rate, recall, and AUC-ROC as shown in Table 19 (see the highlighted row). The result of our proposed model still produces promising performance. When compare with Almeida et al. [6] and Bag of words models, our approach shows significant improvement across all the evaluation metrics. It is important to note that the Bag of words model was implemented using *NaiveBayesMultinomialText* classifier in WEKA, which deals specifically with text classification task. The default parameters used for *NaiveBayesMultinomialText* classifier are shown in Fig. 9. In the case of Dset2 (SMS Corpus V.0.1 Big), our model achieves the same level of performance in F-measure and AUC-ROC

Table 19 Comparison of models on Dset1

Model	Evaluation metrics				
	DR	Precision	Recall	F1	AUC-ROC
Proposed	**0.992**	**0.992**	**0.992**	**0.991**	**0.997**
El-Alfy and AlHasan [19]	0.994	0.980	0.997	0.988	0.999
Almeida et al. [6] - SVM + tok1	0.9764	N/A	N/A	N/A	N/A
Bag of words	0.985	0.985	0.985	0.985	0.984

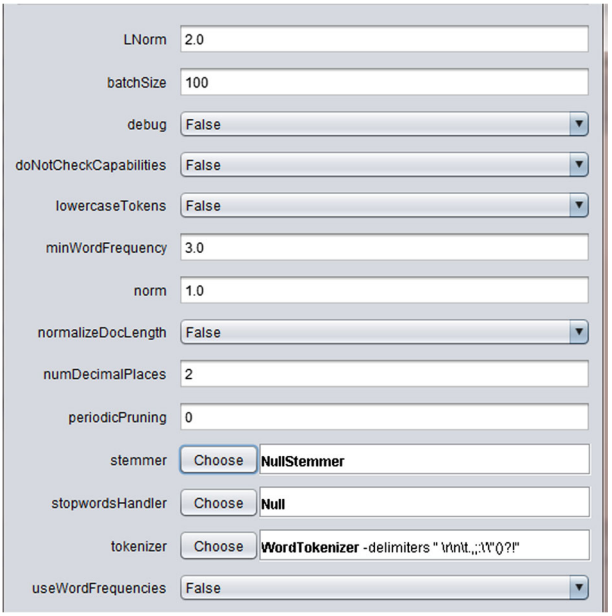

Fig. 9 Default parameters used for *NaiveBayesMultinomialText* classifier

with El-Alfy and AlHasan [19] and improves in precision as shown in Table 20. The model also outperformed the Bag of words model.

Since the Twitter SMS spam message is a private dataset, we benchmark with only the Bag of words model as shown in Table 21. This evaluation shows that the proposed model significantly outperformed the popular content analysis approach using bag of words (see the highlighted row).

5.4 Comparison of spam account detection models

Several approaches have been studied in the literature for spam account detection on Twitter using different private datasets. The reason for the different datasets is due to the Terms of Use of the Twitter API, which forbid researchers from sharing tweets data.

Therefore, to benchmark with the existing approaches selected in the literature, we extract the features used in these studies from our dataset and run the data on the classifiers utilized in the related works considered for model comparison. For instance, Yang et al. [44] evaluated their features using four classifiers: Random Forest, Decision tree, BayesNet, and Decorate. The results obtained for each experiment are highlighted in Tables 22 and 23. The proposed methods are highlighted in the tables.

Table 20 Comparison of models on Dset2

Model	Evaluation metrics				
	DR	Precision	Recall	F1	AUC-ROC
Proposed	**0.991**	**0.991**	**0.991**	**0.991**	**0.999**
El-Alfy and AlHasan [19]	0.993	0.987	0.996	0.991	0.999
Bag of words	0.989	0.989	0.989	0.989	0.997

Table 21 Comparison of models on Dset3

Model	Evaluation metrics				
	DR	Precision	Recall	F1	AUC-ROC
Proposed	**0.932**	**0.932**	**0.932**	**0.932**	**0.983**
Bag of words	0.842	0.845	0.842	0.843	0.923

These results show that our proposed models with both 69 features and 18 features outperformed the approaches in the related studies for spam account detection in microblogging social networks based on the two studies selected for comparison. Thus, the proposed unified framework is promising for developing spam filtering on mobile and microblogging social networks.

6 Conclusion

Spam detection problem is a continuous fight between spammers and spam filters. The increasing rate of evasion tactics on current detection filters has signaled the need to investigate the state-of-the-art features that can separate spam from legitimate messages as well as revealing accounts utilize for malicious activities on microblogging social networks. Existing studies for spam detection in short message communication media focus on identifying spam message and spam account using different frameworks. Due to the inherent characteristics of SMS and microblogging messages, detecting spam messages and spam accounts has been quite challenging. This paper proposes a unified framework that can be used to identify spam messages and spam accounts successfully. The performance of the proposed unified framework is investigated using four datasets, two of which are from SMS spam detection domain and are publicly available for research purpose. The remaining two datasets were collected from Twitter microblog to investigate the capability of the proposed framework for spam message and spam account detection on Twitter. Different from the traditional content-based method for spam message detection, 18 features were identified for detecting both SMS and microblog spam messages. In addition to the use of content/behavioral based features for spammer detection in Twitter, we study the mentioned network of spammers by extracting different graph-based features around the mentioned graph constructed. It was noticed that the mention behavior of spammers based on these features differs from the legitimate users.

Table 22 Comparison using Yang et al. [44] features

Classifiers	Evaluation metrics				
	DR	Precision	Recall	F1	AUC-ROC
Proposed Random Forest with 69 features	**0.932**	**0.933**	**0.932**	**0.932**	**0.977**
Proposed LogitBoost with 18 features	**0.930**	**0.931**	**0.930**	**0.930**	**0.977**
Random Forest	0.924	0.925	0.924	0.924	0.974
Decision tree	0.924	0.927	0.924	0.923	0.961
BayesNet	0.896	0.896	0.896	0.896	0.946
Decorate	0.928	0.931	0.928	0.927	0.971

Table 23 Comparison using Shyni et al. [38] features

Classifiers	Evaluation metrics				
	DR	Precision	Recall	F1	AUC-ROC
Proposed Random Forest with 69 features	**0.932**	**0.933**	**0.932**	**0.932**	**0.977**
Proposed LogitBoost with 18 features	**0.930**	**0.931**	**0.930**	**0.930**	**0.977**
Voting feature interval (VFI)	0.866	0.891	0.866	0.862	0.873
ADTree	0.924	0.928	0.924	0.923	0.970
Random Committee	0.919	0.920	0.919	0.918	0.966

The contribution of this paper centers on the need to address the problem of spam message and spam account detection within a single framework. To achieve this objective, the study introduced a unique set of features to complement the existing features in the related studies, which in turn improves the performance of the proposed framework. In addition, this study investigated the application of evolutionary computation for identifying discriminating features for spam account detection. Based on the classification performance of the different classifiers selected, the proposed framework has demonstrated the capability to detect both spam message and spam account in short message communication media.

To identify the best classifier for the proposed unified framework, the performance of ten (10) classification algorithms were investigated. The results of the various experiments conducted revealed that Random Forest classifier produced the best models for spam message and spam account detection using all the features extracted. The model on SMS spam message detection achieved accuracy, F-measure, and AUC-ROC above 99%, which shows the applicability of the framework for SMS spam message detection. Meanwhile, the model for microblog spam message detection produced accuracy and AUC-ROC of 93.2% and 98.3% respectively. In addition, the spam account detection model achieved accuracy and AUC-ROC of 93.2% and 97.7% respectively. A further investigation was carried out on the possibility of obtaining a minimal number of features for spam account detection using bio-inspired evolution search algorithm. By applying the evolutionary search method for feature reduction, the performance of seven classifiers out of the ten classifiers selected improves significantly. LogitBoost ensemble classifier produced the best result using 18 features identified by the evolutionary search algorithm. The classifier produced accuracy and F-measure of 93.0% and AUC-ROC of 97.7%.

In future, the aim to combine the results of the spam message and spam account detection models using ensemble based method in order to identify the risk level associated with accounts used by spammers to post malicious contents. Building risk assessment model could help identify the category of accounts that may pose a serious threat to legitimate users on the network.

Acknowledgements The work of the authors is supported by University Malaya Research Grant Programme (Equitable Society) under grant RP032B-16SBS.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest.

Ethical approval This research does not involve human or animal, however, it requires data collection from Twitter social network with privacy policy. This research fully complied with the privacy policy of Twitter by

following the Twitter approved procedures for data collection using oAuth authentication. In addition, we do not release the data collected from Twitter to any researcher. The identity of the individual accounts in the data is anonymized.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. Ab Razak MF, Anuar NB, Salleh R, Firdaus A (2016) The rise of "malware": bibliometric analysis of malware study. J Netw Comput Appl 75:58–76
2. Adewole KS, Anuar NB, Kamsin A, Varathan KD, Razak SA (2016) Malicious accounts: dark of the social networks. J Netw Comput Appl. doi:10.1016/j.jnca.2016.11.030
3. Aggarwal A, Rajadesingan A, Kumaraguru P (2012) Phishari: automatic realtime phishing detection on Twitter. In eCrime Researchers Summit (eCrime), 2012 (pp. 1-12). IEEE
4. Ahmed F, Abulaish M (2012) An MCL-based approach for spam profile detection in online social networks. In Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 I.E. 11th International Conference on (pp. 602-608). IEEE
5. Al-garadi MA, Varathan KD, Ravana SD (2016) Cybercrime detection in online communications: the experimental case of cyberbullying detection in the Twitter network. Computers in Human Behavior 63: 433–443
6. Almeida T, Hidalgo JMG, Silva TP (2013) Towards sms spam filtering: results under a new dataset. International Journal of Information Security Science 2(1):1–18
7. Alsaleh M, Alarifi A, Al-Salman AM, Alfayez M, Almuhaysin A (2014) TSD: Detecting sybil accounts in Twitter. In Machine Learning and Applications (ICMLA), 2014 13th International Conference on (pp. 463-469). IEEE
8. Amazon (2016) Amazon Web Services (AWS). Retrieved 3rd January, 2016, from https://aws.amazon.com/what-is-aws/
9. Balakrishnan V, Humaidi N, Lloyd-Yemoh E (2016) Improving document relevancy using integrated language modeling techniques. Malaysian Journal of Computer Science, 29(1):45-55
10. Benevenuto F, Magno G, Rodrigues T, Almeida V (2010) Detecting spammers on Twitter. In: 7th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference, CEAS 2010
11. Bozan YS, Çoban Ö, Özyer GT, Özyer B (2015) SMS spam filtering based on text classification and expert system. In: 2015 23nd Signal Processing and Communications Applications Conference (SIU)
12. Chan PPK, Yang C, Yeung DS, Ng WWY (2014) Spam filtering for short messages in adversarial environment. Neurocomputing 155:167–176. doi:10.1016/j.neucom.2014.12.034
13. Chen C-M, Guan D, Su Q-K (2014) Feature set identification for detecting suspicious URLs using Bayesian classification in social networks. Inf Sci 289:133–147
14. Chu Z, Gianvecchio S, Wang H, Jajodia S (2012a) Detecting automation of Twitter accounts: are you a human, bot, or cyborg? IEEE Trans Dependable Secure Comput 9(6):811–824. doi:10.1109/TDSC.2012.75
15. Chu Z, Widjaja I, Wang H (2012b) Detecting social spam campaigns on Twitter. In International Conference on Applied Cryptography and Network Security (pp. 455-472). Springer, Berlin, Heidelberg
16. Cormack GV, Gómez Hidalgo JM, Sánz EP (2007) Spam filtering for short messages. In: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management
17. Cui X (2016) Identifying suspended accounts in Twitter. http://scholar.uwindsor.ca/etd/5725/
18. Egele M, Stringhini G, Kruegel C, Vigna G (2015) Towards detecting compromised accounts on social networks. IEEE Trans Dependable Secure Comput. doi:10.1109/TDSC.2015.2479616
19. El-Alfy ESM, AlHasan AA (2016) Spam filtering framework for multimodal mobile communication based on dendritic cell algorithm. Futur Gener Comput Syst 64:98-107
20. Ferrara E, Fiumara G (2012) Mining and analysis of online social networks. (PhD doctoral thesis). University of Messina, Italy
21. Ghosh S, Viswanath B, Kooti F, Sharma NK, Korlam G, Benevenuto F., … Gummadi KP (2012) Understanding and combating link farming in the Twitter social network. Proceedings of the 21st International Conference World Wide Web, 61
22. Google (2015) Google safe browsing API. Retrieved 25th November, 2015, from http://code.google.com/apis/safebrowsing/
23. Grier C, Thomas K, Paxson V, Zhang M (2010) @spam: the underground on 140 characters or less. Proceedings of the 17th ACM conference on Computer and communications security, 27–37

24. Hu X, Tang J, Zhang Y, Liu H (2013) Social spammer detection in microblogging. In: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence
25. Khan UU, Ali M, Abbas A, Khan S, Zomaya A (2016) Segregating spammers and unsolicited bloggers from genuine experts on Twitter. IEEE Trans Dependable Secure Comput. doi:10.1109/TDSC.2016.2616879
26. Lee S, Kim J (2014) Early filtering of ephemeral malicious accounts on Twitter. Comput Commun 54:48–57
27. Lee K, Caverlee J, Webb S (2010) Uncovering social spammers: social honeypots plus machine learning. In SIGIR 2010: Proceedings of the 33rd Annual International ACM SIGIR Conference on Research Development in Information Retrieval
28. Liu Y, Wu B, Wang B, Li G (2014) SDHM: a hybrid model for spammer detection in weibo. 2014 Proceedings of the Ieee/Acm International Conference on Advances in Social Networks Analysis and Mining (Asonam 2014), 942–947
29. Liu D, Mei B, Chen J, Lu Z, Du X (2015) Community based spammer detection in social networks. In: Dong XL, Yu X, Li J, Sun Y (eds.) Web-age information management (Vol. 9098, pp. 554–558)
30. Manurung HM (2004) An evolutionary algorithm approach to poetry generation. (Doctor of Philosophy PhD), University of Edinburgh. Retrieved from https://www.era.lib.ed.ac.uk/handle/1842/314
31. Martinez-Romo J, Araujo L (2013) Detecting malicious tweets in trending topics using a statistical analysis of language. Expert Syst Appl 40:2992–3000. doi:10.1016/j.eswa.2012.12.015
32. Metsis V, Androutsopoulos I, Paliouras G (2006) Spam filtering with naive bayes-which naive bayes?. In CEAS (Vol. 17, pp. 28-69).
33. Miller Z, Dickinson B, Deitrick W, Hu W, Wang AH (2014) Twitter spammer detection using data stream clustering. Inf Sci 260:64–73. doi:10.1016/j.ins.2013.11.016
34. Nguyen H (2013) Research report 2013 state of social media spam
35. PhishTank. (2015). Phishtank API. Retrieved 25th November 2015, from http://www.phishtank.com/
36. Sadan Z, Schwartz DG (2011) Social network analysis of web links to eliminate false positives in collaborative anti-spam systems. J Netw Comput Appl 34(5):1717–1723
37. Schütze H (2008) Introduction to information retrieval. In: Proceedings of the international communication of association for computing machinery conference
38. Shyni CE, Sundar AD, Ebby GSE (2016) Spam profile detection in online social network using statistical approach. Asian Journal of Information Technology 15(7):1253–1262
39. Stanford University (2008) Chi-squared feature selection. from http://nlp.stanford.edu/IR-book/html/htmledition/feature-selectionchi2-feature-selection-1.html
40. Thomas K, Grier C, Song D, Paxson V (2011) Suspended accounts in retrospect: an analysis of Twitter spam. In: Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference
41. Twitter (2016) The Twitter rules. Retrieved 28th January, 2016, from https://support.twitter.com/articles/18311
42. Twitter rate limit (2015) Twitter rate limit for search/tweets REST API calls. from https://dev.twitter.com/rest/public/rate-limits
43. URIBL (2015) URIBL API. Retrieved 25th November, 2015, from http://uribl.com/
44. Yang C, Harkreader R, Gu G (2013) Empirical evaluation and new design for fighting evolving Twitter spammers. IEEE Transactions on Information Forensics and Security 8(8):1280–1293. doi:10.1109/tifs.2013.2267732
45. Yoon JW, Kim H, Huh JH (2010) Hybrid spam filtering for mobile communication. Computers & Security 29(4):446–459. doi:10.1016/j.cose.2009.11.003
46. Zainal K, Jali MZ (2015) A perception model of spam risk assessment inspired by danger theory of artificial immune systems. Procedia Computer Science 59:152–161
47. Zhang Y, Lu J (2016) Discover millions of fake followers in Weibo. Soc Netw Anal Min 6(1):1–15
48. Zhang H.-Y, Wang W (2009) Application of bayesian method to spam sms filtering. In: 2009 International Conference on Information Engineering and Computer Science
49. Zheng X, Zeng Z, Chen Z, Yu Y, Rong C (2015) Detecting spammers on social networks. Neurocomputing 159:27–34. doi:10.1016/j.neucom.2015.02.047

Kayode Sakariyah Adewole received B.Sc. and M.Sc degrees in Computer Science from University of Ilorin, Nigeria. He is an academic staff at the Department of Computer Science, Faculty of Communication and Information Sciences, University of Ilorin, Nigeria. Adewole is currently on his PhD program in the Department of Computer System & Technology, Faculty of Computer Science & Information Technology, University of Malaya, Malaysia. His Ph.D. research is in Network Security with specific focus on social networks. His research interests include Network Security, Biometrics, Machine learning, and Big Data analytics.

Nor Badrul Anuar obtained his Master of Computer Science from University of Malaya in 2003 and a Ph.D. at the Center for Information Security & Network Research, University of Plymouth, UK. He is a senior lecturer at the Faculty of Computer Science and Information Technology at University of Malaya, Kuala Lumpur. He has published a number of journal papers related to security areas locally and internationally. He has a good profile of publications in renowned Journals. His research interests include Intrusion Detection System (Intrusion Detection Systems, Intrusion Response Systems, Security Event and Management, Digital Forensic and Network Security), High Speed Network (Switching, Routing, IPV6, and Multicast) and Management Information System (E-thesis, Library Systems and Online Systems). He is also a member of IEEE Communications Society, IEEE Young Professionals and IEEE Computer Society.

Amirrudin Kamsin is a senior lecturer at the Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. He received his BIT (Management) and MSc in Computer Animation from University of Malaya and Bournemouth University, UK respectively. He obtained his PhD from University College London (UCL). His research areas include human computer interaction (HCI), authentication systems, e-learning, mobile applications, serious game, augmented reality and mobile health services.

Arun Kumar Sangaiah has received his Master of Engineering (ME) degree in Computer Science and Engineering from the Government College of Engineering, Tirunelveli, Anna University, India. He had received his Doctor of Philosophy (PhD) degree in Computer Science and Engineering from the VIT University, Vellore, India. He is presently working as an Associate Professor in School of Computer Science and Engineering, VIT University, India. His area of interest includes software engineering, computational intelligence, wireless networks, bio-informatics, and embedded systems. He has authored more than 100 publications in different journals and conference of national and international repute. His current research work includes global software development, wireless ad hoc and sensor networks, machine learning, cognitive networks and advances in mobile computing and communications. Also, he was registered a one Indian patent in the area of Computational Intelligence. Besides, Prof. Sangaiah is responsible for Editorial Board Member/Associate Editor of various international journals.