Social Network Analysis and Mining (2022) 12:149
https://doi.org/10.1007/s13278-022-00977-7

ORIGINAL ARTICLE

Predictive modeling for suspicious content identification on Twitter

Surendra Singh Gangwar[1] · Santosh Singh Rathore[1] · Satyendra Singh Chouhan[2] · Sanskar Soni[2]

Received: 9 February 2022 / Revised: 24 August 2022 / Accepted: 17 September 2022

Abstract
The wide popularity of Twitter as a medium of exchanging activities, entertainment, and information is attracted spammers to discover it as a stage to spam clients and spread misinformation. It poses the challenge to the researchers to identify malicious content and user profiles over Twitter such that timely action can be taken. Many previous works have used different strategies to overcome this challenge and combat spammer activities on Twitter. In this work, we develop various models that utilize different features such as profile-based features, content-based features, and hybrid features to identify malicious content and classify it as spam or not-spam. In the first step, we collect and label a large dataset from Twitter to create a spam detection corpus. Then, we create a set of rich features by extracting various features from the collected dataset. Further, we apply different machine learning, ensemble, and deep learning techniques to build the prediction models. We performed a comprehensive evaluation of different techniques over the collected dataset and assessed the performance for accuracy, precision, recall, and f1-score measures. The results showed that the used different sets of learning techniques have achieved a higher performance for the tweet spam classification. In most cases, the values are above 90% for different performance measures. These results show that using profile, content, user, and hybrid features for suspicious tweets detection helps build better prediction models.

Keywords Suspicious content detection · User-content features · Natural language processing · Machine learning techniques · Social network

1 Introduction

With the availability of the Internet and web-based information, platforms such as Twitter are widely used to support the distribution of information. Twitter allows users to create a network of people to disseminate information and allow the mass communication of the information to a widespread audience (Boukes 2019). For example, Twitter can serve as a platform to help with the crisis management process by looking for specific hashtags. Individuals can also narrate about the crisis or outbreaks, which can be useful in providing assistance and humanitarian support. Currently, government and private organizations, as well as individuals, are using Twitter to share information (Edo-Osagie et al. 2020). In this way, Twitter plays a vital role in the rapid distribution of information (Martinez-Rojas et al. 2018). As the data show, recently, Twitter has refreshed its dynamic client numbers over quite a while to 328 million[1].

While Twitter has established itself to distribute information successfully and rapidly, it has also become a platform for spreading misinformation and panic phenomena fueled by incomplete and inaccurate information (Wang and Zhuang 2017). The scientists at Italy's Bruno Kessler Foundation's Center for Information and Communication Technology[2] have scrutinized around 121,407,000 tweets and reported that more than half of the tweets are rumors and bots spread false news. Other studies have also reported similar findings (Hennig-Thurau et al. 2015). The magnitude

✉ Santosh Singh Rathore
santoshs@iiitm.ac.in

Surendra Singh Gangwar
surendra100598@gmail.com

Satyendra Singh Chouhan
sschouhan.cse@mnit.ac.in

Sanskar Soni
2018ucp1265@mnit.ac.in

[1] ABV-IIITM, Gwalior, India

[2] MNIT, Jaipur, India

[1] https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/.

[2] Italy's Bruno Kessler Foundation's Center for Information and Communication Technologyhttps://ict.fbk.eu/.

Published online: 05 October 2022

of this false information is huge that even the World Health Organization has declared the onslaught of messages as an "infodemic"[3]. The overabundance of information, some are correct, and others are not, makes it difficult for the end-user to find virtuous sources and reliable information when they need it. This can cause a negative impact on the psychology of users, driving them to anxiety and depression. It is highly essential to curb the pitfalls of Twitter to make it a more reliable and trustworthy place (Lingam et al 2019).

Therefore, the boon of Twitter is now turning into a curse as spammers are using these platforms to spread malicious or irritating information to achieve their malevolent intends. In a tweet, the user can add text, URLs, videos, and images. Further, it allows various functionalities, such as following a user, mentioning a topic or user, hashtag, reply, and retweet. Lee and Kim (2013). A hashtag is used to categorize a tweet into a particular category, and all tweets related to that tweet can be read by clicking that tag. At the point when any remarkable occasion happens, a large number of clients tweet about it and quickly make it a trending subject. These trending themes become the objective of spammers who post tweets consisting of some trademark expressions of the moving point with URL interfaces that lead clients to disconnected sites. As tweets usually incorporate abbreviated URL joins, it becomes for the clients to recognize the substance of the URL without stacking the site. Spammers can have a few thought processes behind spamming, for example, advertise a product to produce exceptional yield on deals, compromising the user's account (Lingam et al 2019; Barushka and Hajek 2018; Dokuz 2021). Spammers contaminate the continuous pursuit climate. However, they additionally affect tweets statistics. Filtering malicious content becomes a challenging problem because of URL shorteners, modern and informal languages, and abbreviations used on social networking sites. Spammers influence the users to click a particular URL or read the content with specific phrases or words (Tingmin et al. 2018; Madisetty and Desarkar 2018).

In their study, Kaur et al. (2016) have surveyed research papers published between 2010 and 2015 for malicious tweets and content identification. The authors reported that most of the used techniques for malicious tweets detection could be categorized into four categories. (1) User features-based techniques: These techniques classify a user as spammer or non-spammer by analyzing the user's account information such as no. of followers, no. of following, no. of mentions, and tweets creation time. (2) Content features-based techniques: These techniques analyze the text properties and decide whether tweets are spam or non-spam. The tweet content, such as the number of hashtags in comparison to total word count, users mentioned in a tweet, number of URLs, and count of numerals, are used. (3) Relation features-based techniques: These techniques use the connection degree measures such as whether a person mentioned a direct friend in a tweet or a mutual friend, etc., to identify malicious content. (4) Hybrid feature-based techniques: these techniques drive new features such as reputation (ratio of followers with following), frequency of tweets, and the rate at which user follows other users by using the user features. In 2020, Abkenar et al. (2020) performed a SLR on Twitter spam detection and reported that spam detection approaches had used content analysis approaches (15%), user analysis approaches (9%), tweet analysis approaches (9%), network analysis approaches (11%), and hybrid analysis approaches (56%). Furthermore, the authors stated that collecting real-time Twitter data, labeling datasets, spam drifting, and class imbalance problems are open challenges in Twitter spam detection approaches.

In this paper, first, we collect the spam dataset from Twitter by utilizing Twitter developer API. We fetch 4000 latest tweets, consisting of information like timestamp, tweet text, username, hashtags, followers count, following count, the number of mentions, word count, retweets, etc. Further, we perform feature engineering and extract different feature sets such as content-based and user-based features. Additionally, we create hybrid features such as the user's reputation, frequency of tweets of a user, and following frequency. Further, we label the dataset as spam or non-spam using hybrid features, blacked list URLs, and some predefined words in the text. Afterward, we apply different machine learning and deep learning techniques to predict suspicious or malicious tweets. Further, we perform an analysis to assess how different techniques are performed to predict suspicious content on Twitter. Specifically, we made the following contributions in the presented work.

1. We create a spam dataset to detect suspicious content of Twitter.
2. We extract different features from the collected Twitter dataset.These features are language based, content based, and user based. Further, we create hybrid features to enrich the feature set for building effective prediction models.
3. We apply two different natural language processing (NLP) techniques, bag of words and TF-IDF to extract different language features.
4. We apply different machine learning and state-of-the-art deep learning techniques and evaluate their performance for the suspicious content detection on Twitter.

The rest of the paper is organized as follows. Section 2 discusses works related to the techniques used for the Tweets

[3] https://www.washingtonpost.com/science/2020/03/17/analysis-millions-coronavirus-tweets-shows-whole-world-is-sad/.

spam classification. The Twitter spam data collection and feature extraction procedure are presented in Sect. 3. The experimental analysis and results are provided in Sect. 4. Section 5 concludes the paper.

2 Related studies

In this section, we discuss some of the state-of-arts related to proposed work. Kaur et al. (2016) have reported a review of various research papers published between 2010 and 2015 and discussed techniques used in these research papers. The authors stated that researchers had utilized numerous methods for spam detection. Most of the works have been done by considering tweets' content and profile-based features. Dangkesee and Puntheeranurak (2017) performed an adaptive classification for spam detection. Authors have used spam world filter and URL filter using black-listed URLs. After labeling and preprocessing the data set, the Naive Bayes classifier used 50000 and 10000 tweets. The results found that the proposed model outperformed spam world filters by comparing accuracy, precision, recall, f1-score. In the end, the authors have suggested the utilization of safe browsing instead of URL blacklisting for filtering URLs.

Raj et al. (2020) applied multiple machine learning algorithms to classify tweet content. The experimental results showed that out of the used techniques, KNN (92%), decision tree classifier (90%), random forest classifier (93%), and naive Bayes classifier (69%) outperformed other techniques. The authors suggested that the tweet be deleted after detecting it as spam. Song et al. (2011) presented Bagging, SVM, J48, BayesNet with relation-based features by creating graphs between users. The authors have used measures such as distance and connectivity between users. The results showed that Bagging outperformed other techniques with a 94.6% true positive rate and 6.5% false positive rate. The authors have also highlighted that if any user created a new account and generated a tweet, it would be added to the spammer category, even if it is not spam. It is due to the classification of the user as malicious earlier.

Alom et al. (2020) have applied CNN with tweet text and with both tweet text and meta-data features for the spam classification. The presented approach utilized NLP methods such as word embeddings and n-grams methods. The approach converts the text into a matrix before sending it to CNN. The method that combined both the features produced better accuracy of around 93.38%. The presented approach outperformed other used deep learning methods.

Mateen et al. (2017) proposed a hybrid solution for spam detection that used different combinations of features such as content-based, graph-based, and user-based features. The authors have applied J48, decorate, and naive Bayes classifiers on the dataset having these features. The results showed that content and graph-based feature-based models achieved an accuracy of 90%, and the user and graph-based feature-based models achieved an accuracy of 92%. The presented work also performed correlation analysis between features and removed features with higher correlations.

Sagar and Manik (2017) have applied different machine learning algorithms for Twitter spam detection. The presented work has used SVM as the principal classifier. The authors have introduced a new feature that matches the tweet content with URL destination content. The experimental dataset consists of a random set of 1000 tweets; out of those 1000 tweets, 95–97% were classified correctly. Arushi and Rishabh (2015) proposed an integrated algorithm that combines the benefits of three distinctive learning algorithms (to be specific naive Bayes, clustering, and decision trees) was implemented. This incorporated calculation classifies a record as spammer/non-spammer with a by and large precision of 87.9%. Lin and Huang (2013) analyzed the importance of existing features for recognizing spammers on Twitter and utilized two basic yet compelling features (i.e., the URL rate and the collaboration rate) to characterize the Twitter accounts. This study, dependent on 26,758 Twitter accounts with 508,403 tweets, shows that the classification has precision up to around 0.99 and 0.86 and a higher recall.

Willian and Yanqing (2013) proposed a versatile strategy to distinguish spam on Twitter using content, social, and graph-based data, and after various examinations, an edge and acquainted-based model is made. This new model is contrasted with SVM and two other existing calculations utilizing accuracy, precision, and recall. The new classifier with an accuracy of 79.26% is superior to SVM with a precision of 69.32%. Wu et al. (2017) used various deep learning techniques utilizing training through word vectors and creation of various classifiers through ML algorithms. Doc2Vec was used as the word vector training model, and machine learning algorithms included random forest, naive Bayes, and decision tree. The author collected 10 days ground truth data from twitter consisting of 1,376,206 spam tweets and 673,836 non-spam messages and created four different datasets with varying spam to non-spam ratios. MLP proved to perform the best on all four datasets. Tang et al. (2014) tried a unique approach of extracting out features from tweets using deep learning networks in order to capture syntactic texts of embedded words and labels. However, the machine learning algorithms using these features did not perform that well as the best f1 score was reported to be 87.61% (<90%).

The previous work done in spam detection on Twitter predominantly centers around the profile and content-based features. Better utilization of other features in Twitter spam detection is still a major concern (Tingmin et al. 2018). Additionally, there is a need for adding hybrid features in training set for tweet classification. The proposed work uses two different datasets with different features combinations

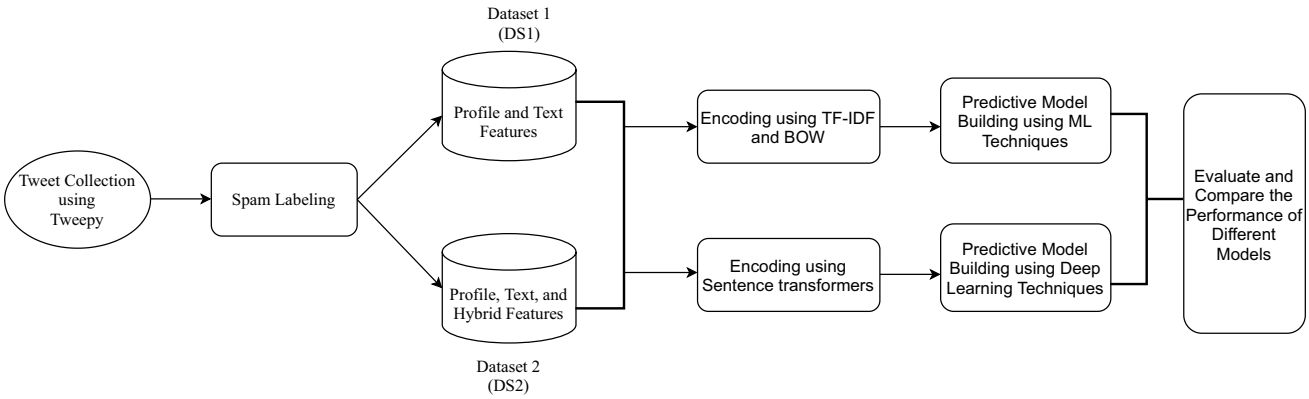

Fig. 1 Twitter data collection, feature extraction procedure and ML/DL model evaluation

to analyze different machine learning, ensemble, and deep learning techniques.

3 Twitter spam dataset collection

The overview of the dataset collection, feature extraction, and model evaluation procedure is depicted in Fig. 1.

The proposed work utilizes the tweets fetched using Twitter developer API[4]. Twitter allows its users to fetch Twitter data using the Tweepy library[5]. The Tweepy library required four user credentials like *consumer_key*, *consumer_secret*, *access_key*, *access_secret* to send the request over API. We fetched 4000 latest tweets, consisting of many features like timestamp, tweet text, username, hashtags, followers count, the following count, number of mentions, word count, retweet, etc. All of these features are categorized into content-based features and user-based features. Further, we create various hybrid features such as the user's reputation, frequency of tweets of a user, and following frequency. For labeling the dataset as spam or non-spam, we use hybrid features, blacked list URLs, and some predefined words in the text (Gupta et al. 2018. Finally, the dataset is prepared for analyzing the performances of different machine learning models. Two different datasets are created by combining user-content features, user-relation features, user-content-relation features.

We collect the features of three different categories as described below.

1. *Profile-based features* These features concern the profile properties of the users. A user's account includes important information such as the number of followers, the number of following, the number of mentions, and tweets creation time.
2. *Content-based features* These features concern the text properties of the tweets (Chen et al. 2017). A tweet content has some crucial information such as the number of hashtags, total word count, users mentioned in a tweet, the number of URLs, and count of numerals.
3. *Hybrid features* These features are derived from the user-based features. Some new features that can be derived are reputation (ratio of followers with following), frequency of tweets, the rate at which user follows other users, account age, metric entropy for all textual features, the proportion of similarity in username and screen name, etc.

Table 1 describes the important features that we have extracted from the collected dataset. Specifically, we focus on the following properties to extract different feature sets.

1. *Count of the number of followers and followees* Followers are those users who follow a specific user, while followees are the users who a specific user follows. In general, spammers have limited numbers of followers but large followees. Therefore, users with large followees and limited numbers of followers can be considered spam account.
2. *URLs* URLs are the connections that direct to some other page on the program. With URL shorteners' improvement, it has become simple to post irrelevant connections on any OSN. This is because URL shorteners hide the original content of the URL, making it hard for detection algorithms to detect malicious URLs. An excessive number of URLs in tweets of a user are an expected pointer of the user being a spammer.
3. *Spam words* A record with spam words in pretty much every tweet can be viewed as a spam account. Subse-

[4] https://developer.twitter.com/en.

[5] https://www.tweepy.org/.

Table 1 Description of the features collected for the Twitter's spam dataset

Feature name	Feature type	Feature description
AccountAge	Profile	Days since account creation to date of collection
FollowersCount	Profile	In user profile meta-data
FriendsCount	profile	In user profile meta-data
StatusesCount	Profile	In user profile meta-data
DigitsCountInNmae	Content	Number of digits in screen name
TweetLen	Content	Number of characters in tweet
UserNamelen	Content	Number of characters in user name
ScreenNameLen	Content	Number of characters in screen name
Metric entropy for all textual features: tweet, user profile description, user name and screen name, respectively	Hybrid	To measure randomness in text. $\frac{H(X)}{\lvert X\rvert}$. Where $\lvert X\rvert$ is the length of a string X, and H(X) is the Shannon entropy of text
URIsRatio	Hybrid	$\frac{\lvert Characters\ in\ URLs\rvert}{\lvert tweet\ length\rvert}$
MentionsRatio	Hybrid	$\frac{\lvert Characters\ in\ user\ mentions\rvert}{\lvert tweet\ length\rvert}$
NameSim	Hybrid	Proportion of similarity in user name and screen name
Friendship	Hybrid	$\frac{FriendsCount}{FollowersCount}$
Followership	Hybrid	$\frac{FollowersCount}{FriendsCount}$
Interestingness	Hybrid	$\frac{FavouritiesCount}{StatusesCount}$
Activeness	Hybrid	$\frac{StatusesCount}{AccountAge}$
VerifiedAccount	Profile	In tweet meta-data
FavouritiesCount	Profile	In user profile meta-data
NamesRatio	Hybrid	$\frac{\lvert ScreenName\ length\rvert}{\lvert UserNamelength\rvert}$

quently, text including spam words can be considered as a significant factor for identifying spammers.

4. *Replies* Since, data or message sent by a spammer is pointless, thusly individuals once in a while answers to its post. On the other hand, a spammer answers to an enormous number of presents altogether on getting seen by numerous individuals. This example can be utilized in recognition of spammers.
5. *Hashtags* Hashtags are the novel identifier ("#" trailed by the identifier name) which is utilized to bunch comparative tweets together under a similar name. Spammers utilize enormous #hashtags in their posts, with the goal that their post is posted under all the hashtag classifications and consequently gets high viewership and is perused by others.

The hybrid features are included in the dataset to understand the dynamism of features such as "statuses count, friends count, followers count, favorites count, naming conventions and tweeting patterns." Account age shows the frequency of user activity. Accounts with a very high value of status and friends count, but a low value of favorites count and followers count are prone to spam accounts. The username and screenname of a legitimate user are usually similar, and the username is not very lengthy and does not begin with a digit. If these naming conventions are not followed, such users are usually spam accounts. NameSim and NamesRatio features capture this aspect of the accounts. A suspicious spam account usually posts 12 or more tweets per day, whereas a legitimate account posts on average 4 tweets per day. We have considered these characteristics of the user accounts and calculated hybrid features. The details of these features are given in Inuwa-Dutse et al. (2018).

3.1 Labeling of spam dataset

Initially, all of the tweets are unlabeled. We perform a data labeling process and assign spam or non-spam label to reach tweets. Concone et al. (2019) have presented a labeling technique for the Twitter spam account. The authors have used malicious URLs and recurrent content information to decide whether a tweet is spam or not. In our work, we use the same technique to label the tweets. The labeling technique's first step is defining some criteria that help decide between spam and trustworthy content. The first criteria to consider

Table 2 Word categories

Category	Words
Ads	Ads, images, banners, Hedberg, RealMedia, img, announcer, popup, offer, adserver, sales, gifs, media, exit, out, adv, splash, pub, pop, graphics
Books	Catalog, book, patterns, weaving, product, sniacademic, news, ebook, educator, library, store, wilecyda
E-commerce	Shop, store, catalog, tickets, art, users, business
Games	Juegos, Jeux, category, game, Xbox, jeunesse, pc, online, Comunidad, consoles, flash, PSP, arcade, Wii, emulator, gratis, Nintendo, PlayStation
Medical	Health, conditions, article, content, diseases, meds, group
News	News, newspapers, media, publications, section, feed, opinion, business, community, archive, papers, profile
Sport	Sport, athletics, team, basketball, football, college, women, track, tennis, soccer, baseball, golf, mens

are the publication of URLs of some malicious sites in the tweet. It is simple to detect malicious content. Another criterion is the publication of duplicate content or messages to spread some information. This strategy is often used to disseminate misinformation. The use of vocabulary and other meta-information is also used as the criteria. Based on these characteristics, we design and use the labeling technique.To label a tweet as spam or non-spam, we used a combination of word category filter, URL filter, and some hybrid and profile-based meta-features. They are described as follows.

1. *Word category filter* In this filter, we create some rules combining different words as given in Table 2. For example, the words such as free available, dear friend, new offer, click here, unlimited offers, and register here are considered. Furthermore, some suspicious words used for marketing purposes offer, register, extra, guarantee, discount, deal, collect, buy, apply now, bonus, free, sales, unlimited, win, purchase, order now, lowest are also considered (Martinez-Romo and Araujo 2013). If a combination of these words occurs in the tweet, we put it into the spam category. Tweets that contained at least two of the keywords are marked as spam.
2. *URL filter* In this filter, we check the URL that is shortened, from this URL we found the original URL. After finding the original URL, we match it with black-listed URLs. Additionally, we check whether it is a secured or non-secured URL. If a tweet consists of any black-listed URL, we label that tweet as spam. We have considered three factors when analyzing URLs in a tweet. (1) Is it a safe URL or not according to the Google Safe Browsing (GSB), (2) the total number of URLs posted in a tweet, and (3) the ratio of the total number of URLs and the unique URLs in a tweet. A tweet is labeled as spam if at least one URL is malicious or the ratio of the unique URLs <=0.25.
3. *Based on hybrid and profile features* There are some important hybrid and profile features on the basis of which we can label a tweet. These features include the ratio of friends count and followers count, the ratio of the status count, and account age. Some profile features are also used for labeling, such as *Is_verified and Listed*. *Is_verified* represents whether the user is verified or not checked from the Twitter security bot. *Listed* represents how many times the user reported. Table 1 lists all hybrid and profile features used in the paper.

We produced a labeled dataset after completing the labeling procedure, which will be used for model building and evaluation. For the experimentation, we create two different datasets, DS1 and DS2. These datasets can be found here[6].

- *Dataset-1 (DS1)* It consists of profile-based features and content-based features. DS1 dataset has total 3650 instances, out of which 1897 are normal and 1753 are spam.
- *Dataset-2 (DS2)* It consists of profile-based features, content-based features, and hybrid features. DS2 dataset has total 9678 instances, out of which 5398 are normal and 4280 are spam.

3.2 Extraction of NLP features from tweet's text

The used spam datasets consist of text of the tweets. This textual information can classify the tweets into spam and non-spam categories. However, the used machine learning techniques cannot work with raw text directly (Kim and Gil 2019). Therefore, the text must be converted into numbers. We have used bag of words and TF-IDF vectorizer NLP techniques to extract features from the tweets' text.

[6] https://github.com/ssrathore/Suspicious_Tweets-datasset.

3.2.1 Bag of words (BOW)

Bag of words is a popular and simple feature extraction method from text data. This technique changes tokens of words into a series of features to utilize information within the words. Each word is utilized to prepare the classifier in the BoW model. There are mainly three steps used to create the BOW model (Qader et al. 2019). (1) The preprocessing step converts text into lower case and removes all unnecessary information. (2) Building vocabulary step counts the occurrences of the words, checks whether words from sentences exist in the vocabulary or not, and prepares a final dictionary of the words. (3) The text vectorization step constructs a matrix of features by analyzing the presence or absence of words in sentences.

3.2.2 Term frequency-inverse document frequency (TF-IDF)

The TF-IDF technique is used to count the number of words in a set of documents. It assigns each word a score to indicate its prominence in the text and document. *Term frequency (TF)* determines how often a term shows up in the whole document. It can be considered the likelihood of discovering a word inside the document. *Inverse document frequency (IDF)* is a metric that determines whether a word is uncommon or common among all documents in a corpus. The closer a term is to zero, the more common it is. IDF is calculated by taking the total number of documents, dividing it by the number of documents that contain a word, and then calculating the logarithm (Aizawa 2003). *Term frequency-inverse document frequency (TF-IDF)* is the multiplication of TF and IDF. A word with a high recurrence in a record and a low archive recurrence in the corpus has a high TF-IDF score. The IDF value reaches 0 for a term that appears in almost all texts, bringing the tf-idf closer to 0. When both IDF and TF have higher values, the TF-IDF value is high, indicating that the word is uncommon in the document but common within it.

The description of the BOW and TF-IDF methods can be referred from Appendix B.

4 Experimental analysis and results

We have used various machine learning techniques, ensemble techniques, and deep learning techniques for building the prediction models to classify tweets into spam and non-spam categories. We have reported and compared the performance of different techniques for dataset-1 (DS1) and dataset-2 (DS2).

4.1 Used machine learning and ensemble techniques

We have used five different machine learning techniques, K-nearest neighbors, logistic regression, naive Bayes, decision tree, and random forest. Further, we have used three different ensemble techniques, bagging, boosting, and stacking. These are the used widely used techniques for the tweets spam classification task. Therefore, we selected these techniques in the presented work.

4.2 Used deep learning techniques

We have used seven different deep learning techniques, ANN (64 and 32 layers), long short-term memory (LSTM),GRU, single convolution layer, two convolution layers, very deep convolution neural network (VDCNN), and convolution + LSTM. A brief description of these techniques is given below.

4.2.1 Artificial neural networks (ANN)

It can be imagined as a single or a group of neurons and is also referred to as a feed-forward neural network. It consists of 3 layers: the input, hidden, and output layers. It is very well capable of handling the non-linearity in data. For the implementation, we used a basic ANN with two hidden layers consisting of 64 and 32 neurons, respectively. We also used a Dropout layer with a dropout rate of 0.2 between two hidden layers.

4.2.2 Long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997; Adhikari et al. 2019)

Long short-term memory networks (LSTMs) are a unique form of recurrent neural network (RNN) capable of handling long-term dependencies. Instead of having a single layer of the neural network, four communicate uniquely. Some of the works used the LSTM model for different text classification tasks Adhikari et al. (2019); Yang et al. (2018); Zhou et al. (2016); Yang et al. (2016). We used a single LSTM layer with 32 memory cells for the experimental purpose.

4.2.3 Gated recurrent units (GRU) (Cho et al. 2014)

GRU are more or less similar to LSTM but only has two gates, namely the reset and the update gate. This has a much lower training time than the LSTM due to fewer parameters. It was initially proposed to capture features from different time scales adaptively. We used a GRU with 128 units for the

experimental purpose, followed by a fully connected layer of 32 neurons and a classification layer.

4.2.4 Convolution layer (Conv1D)-based networks

Employing convolution layers, we implemented four different models based on it. Using convolution in NLP-related tasks is a recent development. All the CNN-based networks extract the *n-gram*-based feature using varied sizes of kernels/filters.

- *Single and multilayer convolution* Two separate models where the one with only a single convolution layer had 100 filters with five as the kernel size. The other model had two consecutive convolution layers with 100 filters and kernel sizes of 3 and 4, respectively.
- *Very deep convolution neural network (VDCNN)* (Simonyan and Andrew 2014) VDCNN uses multiple layered convolution and max-pooling operation. The model makes use of four pooling operations, each of which reduces the resolution by half, resulting in four different feature map tiers: 64, 128, 256, and 512. At the end of 4 convolution pair operations, the resulting feature vector of size $512 \times k(k = 3)$ resulting features is transformed into a single vector. This is fed into a three-layer fully connected classifier (4096,2048,2048) with ReLU hidden units.
- *Convolution + LSTM* A mixed model captures short and long-range dependencies. It consists of a convolution layer followed by a pooling, LSTM, and the classification layer. The convolution layer with 100 filters uses a kernel of size 5. The max-pooling layer has a pool size of 2. The LSTM layer used has 32 memory cells followed by a fully connected layer of 32 neurons and the classification layer.

4.3 Performance evaluation measures

We have used four different performance evaluation measures to assess the performance of different used techniques for the spam tweets detection. They are: accuracy, precision, recall, and f1-score Gorunescu (2011). The description of performance measures is given in Appendix A, Table 8.

4.4 Implementation details

We have used different Python libraries to implement different machine learning and deep learning techniques. All the experiments were carried out on a system having with Dual-Core Intel Core i5 processor and 8 GB RAM, running MacOs BigSur, with 64- bit processor and access to NVidia K80 GPU kernel. To implement the machine

Table 3 Different Ml models with bag of words on DS1 and DS2

Classifier	Accuracy	Precision	Recall	F1-score
ML models with Bag of words on DS1				
Logistic Regression	0.8763	0.87351	0.86263	0.87543
Naive Bayes	0.68367	0.67542	0.67324	0.68453
KNN	0.83106	0.83225	0.84751	0.82257
Decision Tree	0.90127	0.90543	0.91248	0.89112
Random Forest	0.91602	0.90251	0.92152	0.91358
ML models with bag of words on DS2				
Logistic Regression	0.916	0.953	0.855	0.901
Naive Bayes	0.544	0.495	0.932	0.647
KNN	0.839	0.894	0.726	0.802
Decision Tree	0.99	0.99	0.98	0.991
Random Forest	0.992	0.99	0.985	0.992

Table 4 Different ML models with TF-IDF on DS1 and DS2

Classifier	Accuracy	Precision	Recall	F1-score
ML models with TF-IDF on DS1				
Logistic Regression	0.91375	0.91256	0.90145	0.90628
Naive Bayes	0.6912	0.68845	0.69014	0.68158
KNN	0.83219	0.82156	0.84751	0.83348
Decision Tree	0.9241	0.92147	0.91254	0.91469
Random Forest	0.94666	0.93458	0.94375	0.94112
ML models with TF-IDF on DS2				
Logistic Regression	0.753	0.83	0.567	0.671
Naive Bayes	0.544	0.495	0.932	0.647
KNN	0.839	0.895	0.726	0.801
Decision Tree	0.99	0.988	0.989	0.989
Random Forest	0.99	0.99	0.98	0.99

Table 5 Ensemble techniques with BOW on DS1 and DS2

Classifier	Accuracy	Precision	Recall	F1-score
Ensemble techniques with BOW on DS1				
Bagging	0.997	0.99	0.99	0.99
Boosting	0.986	0.986	0.987	0.986
Stacking	0.92	0.869	0.993	0.927
Ensemble techniques with BOW on DS2				
Bagging	0.783	0.878	0.598	0.712
Boosting	0.823	0.79	0.824	0.807
Stacking	0.932	0.876	0.98	0.929

Table 6 Ensemble techniques with TF-IDF on DS1 and DS2

Classifier	Accuracy	Precision	Recall	F1-score
Ensemble techniques with TF-IDF on DS1				
Bagging	0.94368	0.94283	0.93451	0.93457
Boosting	0.90354	0.90228	0.9134	0.91586
Stacking	0.93765	0.92506	0.92355	0.93679
Ensemble techniques with TF-IDF on DS2				
Bagging	0.95242	0.94525	0.95221	0.94625
Boosting	0.93691	0.93542	0.92231	0.92042
Stacking	0.91969	0.90125	0.90589	0.91287

Table 7 Deep learning techniques based models on DS1 and DS2

Models	Accuracy	Precision	Recall	F1-score
Deep learning techniques on DS1				
BASIC ANN 64 and 32 layers	0.979	0.969	0.988	0.978
LSTM	0.673	0.652	0.637	0.646
Single convolution layer	0.979	0.974	0.982	0.978
Two convolution layer	0.986	0.997	0.972	0.985
GRU	0.983	0.978	0.986	0.982
VDCNN	0.938	0.99	0.868	0.929
Convolution + LSTM	0.923	0.88	0.954	0.92
Deep learning techniques on DS2				
BASIC ANN 64 and 32 layers	0.928	0.948	0.868	0.916
LSTM	0.612	0.606	0.378	0.464
Single convolution layer	0.906	0.951	0.832	0.88
Two convolution layer	0.87	0.896	0.801	0.846
GRU	0.558	0.559	0.047	0.086
VDCNN	0.829	0.977	0.631	0.767
Convolution + LSTM	0.558	0.682	0.021	0.041

learning models, we used TF-IDF and bag of words as the text embedding, whereas to test the deep learning models, we used the pre-trained *Paraphrase-distilroberta-base-v1 embedding* (Reimers et al. 2019). The experimental package with the twitter spam dataset and source code can be found here[7].

4.5 Experiment results

We have used five different machine learning techniques and three different ensemble techniques to build and evaluate the prediction models. These techniques have been applied to both DS1 and DS2 datasets with the bag of words (BOW) and TF-IDF feature extraction methods. The results of the experimental analysis are reported in Tables 3, 4, 5, 6, and 7.

[7] https://github.com/ssrathore/Suspicious_Tweets-datasset.

4.5.1 Results of machine learning techniques for the tweet spam classification

Tables 3 and 4 show the results of ML techniques with BOW and TF-IDF on DS1 and DS2 datasets in terms of accuracy, precision, recall, and f1-score measures. From Table 3, it can be seen that among the used ML techniques, random forest and decision tree have produced the best prediction performance across all the measures. The highest achieved values for all the measures are above 90%. The naive Bayes technique produced the lowest performance for all the measures. The performance of the ML techniques has been improved for the DS2, which consists of the profile, user, and hybrid features. Similarly, from Table 4, it can be observed that again decision tree and random forest techniques are the top performers for all the performance measures. The values of all the measures are greater than 90% for the decision tree, random forest, and logistic regression techniques. Again, the performance of ML techniques has been improved for the DS2.

4.5.2 Results of ensemble techniques for the tweet spam classification

Tables 5 and 6 show the results of ensemble techniques with BOW and TF-IDF on DS1 and DS2 datasets in terms of accuracy, precision, recall, and f1-score measures. From Table 5, it can be observed that the bagging technique produced the best prediction performance for all the measures followed by the boosting technique. The three ensemble techniques have achieved values above 90% for all the measures. The stacking technique produced the lowest performance for all the measures. However, the performance of the ensemble techniques has been decreased for the DS2, which consists of the profile, user, and hybrid features. Similarly, from Table 6, it can be seen that again the bagging technique is the best performer, followed by the boosting technique. It is true for both DS1 and DS2 datasets. The values of all the measures are again greater than 90% for the bagging and boosting techniques. Again, the performance of ensemble techniques has been decreased for the DS2.

4.5.3 Results of deep learning techniques for the tweet spam classification

Table 7 shows the results of different deep learning techniques on DS1 and DS2 datasets in terms of accuracy, precision, recall, and f1-score measures. The table shows that except for the LSTM technique, all other used deep learning

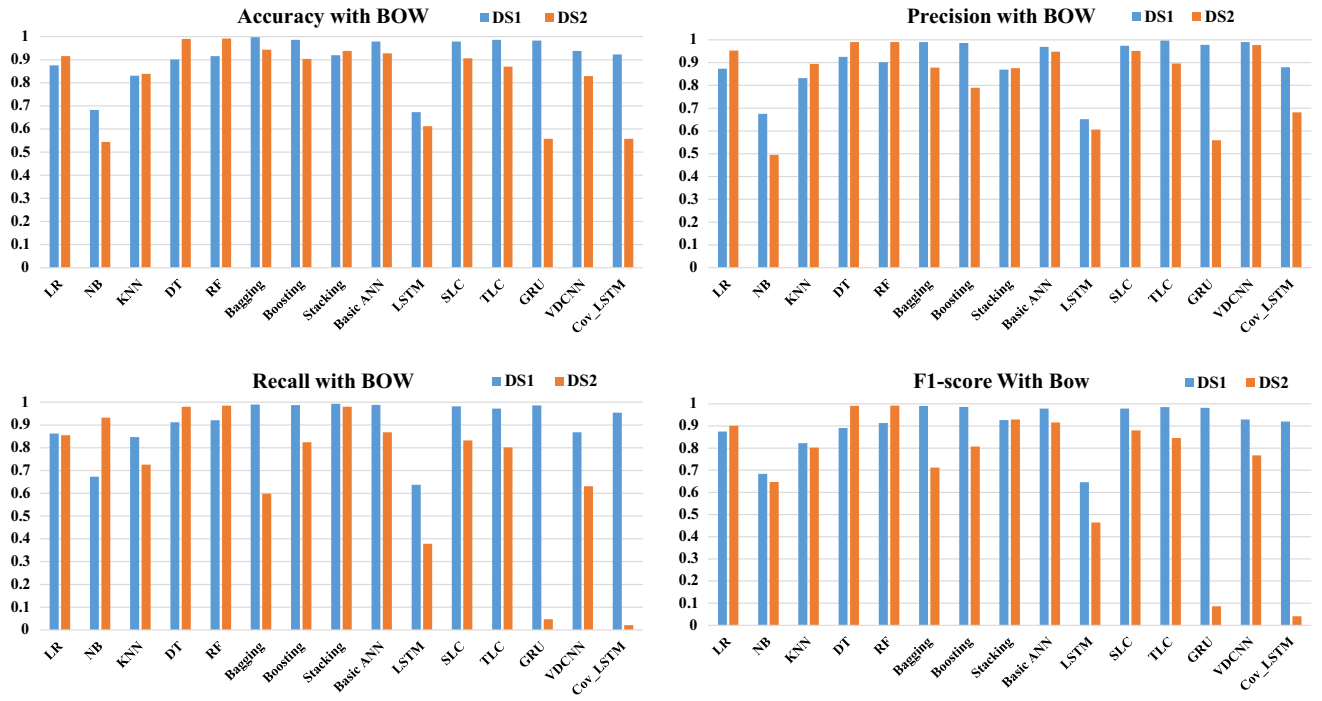

Fig. 2 Comparison of different used ML, ensemble, and deep learning techniques with bag of words (BOW) on DS1 and DS2 datasets, (*LR= Logistic Regression, NB= Naive Bayes, KNN= K-nearest neighbors, DT= Decision Tree, RF= Random Forest, SLC= Single layer convolution, TLC= Two layer convolution, GRU= Gated recurrent unit, Cov_LSTM= Convolution + LSTM, VDCNN= Very deep convolution neural network)

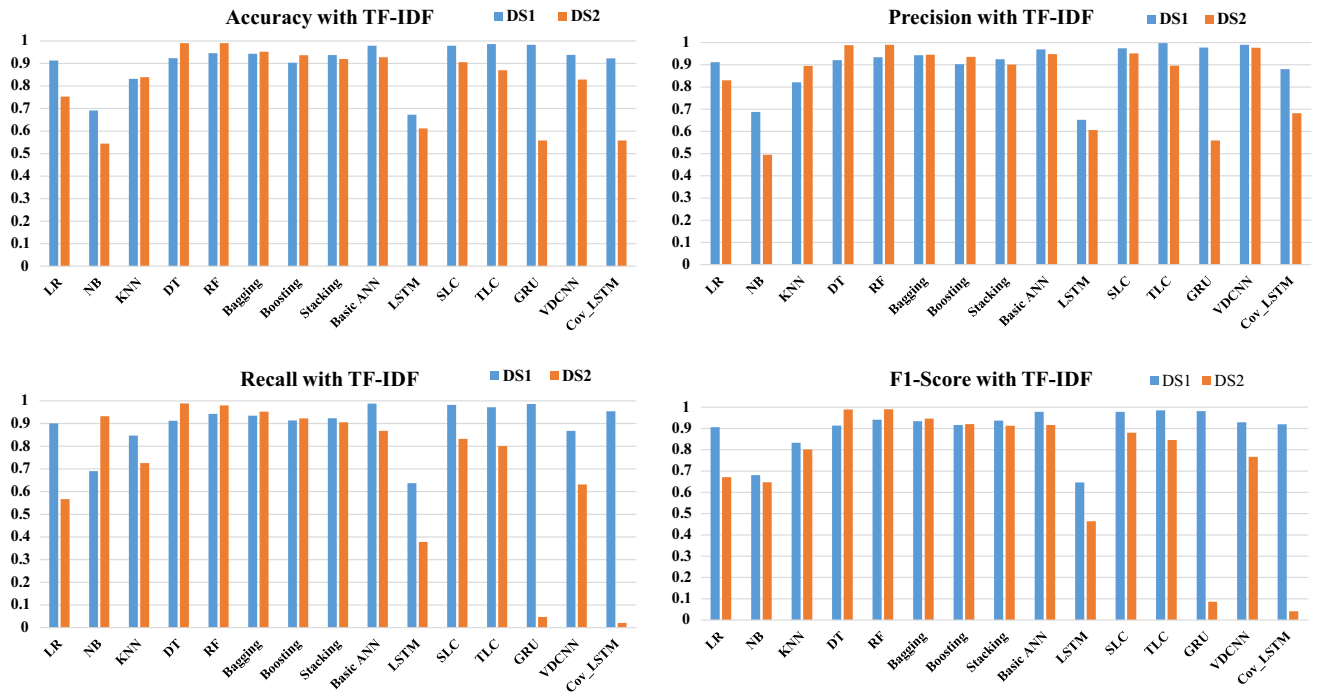

Fig. 3 Comparison of different used ML, ensemble, and deep learning techniques with TF-IDF on DS1 and DS2

techniques have produced a higher performance for spam classification on the DS1 dataset. The values are above 90% for all the measures in most cases. For the DS2 dataset, the performance of the deep learning techniques has been decreased. Here, basic ANN and single convolution layer techniques produced a performance greater than 90%. GRU and the Convolution + LSTM have performed relatively poorly on DS2.

4.5.4 Performance comparison of the used machine learning, ensemble, and deep learning techniques for the tweet spam classification

Figures 2 and 3 show the performance comparison of the used different set of techniques for the BOW and TF-IDF on DS1 and DS2 datasets. The X-axis represents the set of techniques, and Y-axis shows the achieved performance values. From Fig. 2, it is observed that overall, ensemble techniques and deep learning techniques (except the LSTM technique) have performed better than the machine learning techniques on the DS1 dataset. However, the performance of decision tree and random forest techniques is comparable or better than ensemble and deep learning techniques for the DS2 dataset. For the recall and f1-score measures, LSTM, GRU, and convolution+LSTM techniques have performed relatively poorly compared to other used techniques in the case of DS2. Overall, techniques performed better for the DS1 and relatively poorly for DS2. Similarly, from Fig. 3, it is seen that again ensemble learning techniques and deep learning techniques produced a better performance compared to the machine learning techniques on DS1. The performance of machine learning and ensemble techniques has improved for the DS2. In comparison, the performance of deep learning techniques has been decreased for the DS2.

Overall, from the presented experimental analysis, we found that the used different sets of learning techniques have achieved a higher performance for the tweet spam classification. In most cases, the values are above 90% for different performance measures. These results show that using profile, content, user, and hybrid features for suspicious tweets detection helps build better prediction models.

4.6 Discussion of results

This paper aims to develop models for the suspicious tweets' identification using different features such as profile-based, content-based, and hybrid features. Different machine learning and deep learning techniques have been applied to build the prediction models. We tried two different combinations of features and thus created two datasets of 3650 and 9778 tweets, respectively. Dataset-1 (DS-1) includes the only profile-based and content-based features. Dataset-2 (DS-2) includes profile-based, content-based, and hybrid features. The results showed that ensemble learning techniques-based models produced equal or better performance than deep learning techniques-based models. The possible reason behind it is that the DS1 and DS2 datasets are not large enough to optimally train the deep learning-based models. Moreover, no improvement in the performance of the deep learning models has been recorded when DS2 is used. Therefore, it can be inferred that adding a hybrid does not help with performance improvement. One exception report has been reported for the Convolution+LSTM model, where the recall value was very low. This issue can be further investigated by optimally tuning the hyperparameters of the technique. Furthermore, it can also be inferred that time-series models such as LSTM are not an ideal choice for the suspicious tweets' identification.

5 Conclusions and future work

This paper focused on detecting suspicious tweets in trending Twitter topics by analyzing the profile, user, content features, and combinations. First, we crawled and extracted the data of Twitter trending topics by using the tweepy library. Further, we extracted different sets of features from the collected Twitter data. Additionally, we labeled the dataset with spam and non-spam labels. Then, we applied and assessed the performance of different machine learning, ensemble, and deep learning techniques for tweet spam classification. The results showed that the dataset with the combination of profile, content, and hybrid features improved the performance of machine learning and ensemble techniques but did not improve deep learning techniques' performance. The used learning techniques performed almost equally for both NLP feature extraction methods, BOW and TF-IDF. In most of the cases, used machine learning techniques produced the performance of 90% or above for different performance measures. The presented work showed that the hybrid features are most important for tweet spam classification. In this paper, we used some common behaviors of the users and content to label the tweets as spam or not and built several models for the identification of spam tweets. The idea was to recognize some patterns to design a method capable of automatically annotating the large-scale dataset. However, there is further scope for improving the filters to use in the spam labeling of tweets. The experimental analysis presented in this work showed that factors such as feature selection and the use of filters for spam labeling greatly influence the performance of the learning techniques. A stable and better-annotated dataset could result in improved performance of the models. Future research work would present an approach to classify a user as a malicious or a valid user. Further, we would like to investigate the dependence among the features and their significance in malicious bot detection.

Table 8 Description of the performance measures

Measure	Description
Accuracy	It is defined as the ratio of correctly predicted examples to the total examples. Accuracy $= \frac{(TP+TN)}{(TP+TN+FP+FN)}$
Precision	It is calculated as the proportion of accurately predicted positive examples to all positive examples predicted. Precision $= \frac{TP}{(TP+FP)}$
Recall	It is defined as the proportion of correctly predicted positive examples to all positive examples in the actual class. Recall $= \frac{TP}{(TP+FN)}$
F1-score	It is the weighted average of precision and recall. F1-score considers both the false positives and false negatives. F1 $-$ score $= \frac{2*Precision*Recall}{(Precision+Recall)}$

**TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative*

Appendix A

Appendix B

Term frequency (TF) It ascertains the occasions a word w_i occurs in a survey r_j; with respect to the total number of words. It is defined by Eq. 1.

$$tf(w_i, r_j) = \frac{\text{Number of times } w_i \text{ occurs in } r_j}{\text{Total number of words in } r_j} \quad (1)$$

Inverse document frequency (IDF) It highlights terms that appear in a small number of documents throughout the corpus, or in plain English, words with a high IDF score. It is defined by Eq. 2.

$$idf(d, D) = log\frac{|D|}{\{d \in D : t \in D\}} \quad (2)$$

where $f_{t,D}$ is the recurrence of the term t in the record D.

$|D|$ is the absolute number of reports in the corpus.

$\{d \in D : t \in D\}$ is the include of archives in the corpus, which contains the term t.

Term frequency-inverse document frequency(TF-IDF) TF-IDF is the multiplication of TF and IDF. It is defined by Eq. 3.

$$tf - idf(t, d, D) = tf(t, D) \times idf(d, D) \quad (3)$$

Acknowledgments This work is partially supported by a Research Grant under National Super computing Mission (India), Grant number: *DST/NSM/R &D_HPC_Applications/2021/24*.

Declarations

Conflict of interest The authors declare no potential conflict of interests with respect to the research, authorship, and/or publication of this article.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Not applicable.

References

Abkenar SB, Kashani MH, Akbari M, Mahdipour E (2020) Twitter spam detection: a systematic review. arXiv preprint arXiv:2011.14754

Adhikari A, Ram A, Tang R, Lin J (2019) Rethinking complex neural network architectures for document classification. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (Long and Short Papers), pp 4046–4051

Aizawa Akiko (2003) An information-theoretic perspective of TF-IDF measures. Inf Process Manag 39(1):45–65

Alom Z, Carminati B, Ferrari Elena (2020) A deep learning model for Twitter spam detection. Online Soc Netw Media 18:100079

Barushka A, Hajek P (2018) Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. Appl Intell 48(10):3538–3556

Boukes M (2019) Social network sites and acquiring current affairs knowledge: the impact of Twitter and Facebook usage on learning about the news. J Inf Technol Politics 16(1):36–51

Chen W, Yeo CK, Lau CT, Lee BS (2017) A study on real-time low-quality content detection on Twitter from the users' perspective. PLoS ONE 12(8):e0182487

Cho K, van Merrienboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation, encoder-decoder approaches. CoRR arXiv:1409.1259

Concone F, Re GL, Morana M, Ruocco C (2019) Twitter spam account detection by effective labeling. InITASEC

Dangkesee T, Puntheeranurak S (2017) Adaptive classification for spam detection on twitter with specific data. In: 2017 21st international computer science and engineering conference (ICSEC), pp 1–4. IEEE

Dokuz AS (2021) Social velocity based spatio-temporal anomalous daily activity discovery of social media users. Appl Intell 52:2745–2762

Edo-Osagie O, De La Iglesia B, Lake I, Edeghere O (2020) A scoping review of the use of Twitter for public health research. Comput Biol Med 122:103770

Gharge S, Chavan M (2017) An integrated approach for malicious tweets detection using NLP. In: 2017 international conference

on inventive communication and computational technologies (ICICCT), pp 435–438. IEEE
Gorunescu F (2011) Classification performance evaluation. In: Data mining. Intelligent systems reference library, vol 12. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-19721-5_6
Gupta A, Kaushal R (2015) Improving spam detection in online social networks. In: 2015 International conference on cognitive computing and information processing (CCIP), pp 1–6. IEEE
Gupta H, Jamal MS, Madisetty S, Desarkar MS (2018) A framework for real-time spam detection in twitter. In: 2018 10th international conference on communication systems & networks (COMSNETS), pp 380–383. IEEE
Hennig-Thurau T, Wiertz C, Feldhaus Fabian (2015) Does twitter matter? the impact of microblogging word of mouth on consumers' adoption of new movies. J Acad Mark Sci 43(3):375–394
Hochreiter S, Schmidhuber Jürgen (1997) Long short-term memory. Neural Comput 9(8):1735–1780
Hua W, Zhang Y (2013) Threshold and associative based classification for social spam profile detection on twitter. In: 2013 ninth international conference on semantics, knowledge and grids, pp 113–120. IEEE
Inuwa-Dutse I, Liptrott M, Korkontzelos I (2018) Detection of spam-posting accounts on Twitter. Neurocomputing 315:496–511
Kim S-W, Gil Joon-Min (2019) Research paper classification systems based on TF-IDF and LDA schemes. Hum-centric Comput Inf Sci 9(1):1–21
Lee S, Kim J (2013) Fluxing botnet command and control channels with URL shortening services. Comput Commun 36(3):320–332
Lingam G, Rout RR, Somayajulu DVLN (2019) Adaptive deep Q-learning model for detecting social bots and influential users in online social networks. Appl Intell 49(11):3947–3964
Lin P-C, Huang P-M (2013) A study of effective features for detecting long-surviving Twitter spam accounts. In: 2013 15th international conference on advanced communications technology (ICACT), pp 841–846. IEEE
Martinez-Rojas M, del Carmen Pardo-Ferreira M, Rubio-Romero JC (2018) Twitter as a tool for the management and analysis of emergency situations: a systematic literature review. Int J Inf Manag 43:196–208
Martinez-Romo J, Araujo Lourdes (2013) Detecting malicious tweets in trending topics using a statistical analysis of language. Expert Syst Appl 40(8):2992–3000
Mateen M, Iqbal MA, Aleem M, Islam MA (2017) A hybrid approach for spam detection for Twitter. In: 2017 14th international Bhurban conference on applied sciences and technology (IBCAST), pp 466–471. IEEE
Pengcheng Y, Sun X, Li W, Ma S, Wu W, Wang H (2018) Sgm: sequence generation model for multi-label classification. arXiv preprint arXiv:1806.04822
Prabhjot K, Anubha S, Jasleen K (2016) Spam detection on twitter: a survey. In: 2016 3rd international conference on computing for sustainable global development (INDIACom), pp 2570–2573. IEEE
Qader WA, Ameen MM, Ahmed BI (2019) An overview of bag of words; importance, implementation, applications, and challenges. In: 2019 international engineering conference (IEC), pp 200–204. IEEE
Raj RJR, Srinivasulu S, Ashutosh A (2020) A multi-classifier framework for detecting spam and fake spam messages in Twitter. In: 2020 IEEE 9th international conference on communication systems and network technologies (CSNT), pp 266–270. IEEE
Reimers N, Gurevych I, Thakur N (2019) Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 conference on empirical methods in natural language processing. Association for Computational Linguistics
Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
Song J, Lee S, Kim J (2011) Spam filtering in Twitter using sender-receiver relationship. In: International workshop on recent advances in intrusion detection, pp 301–317. Springer, Cham
Sreekanth M, Sankar DM (2018) A neural network-based ensemble approach for spam detection in Twitter. IEEE Trans Comput Soc Syst 5(4):973–984
Tang D, Wei F, Qin B, Liu T, Zhou M (2014) Coooolll: a deep learning system for Twitter sentiment classification. In: Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014), pp 208–212, Association for Computational Linguistics, Dublin
Tingmin W, Wen S, Xiang Y, Zhou Wanlei (2018) Twitter spam detection: survey of new approaches and comparative study. Comput Secur 76:265–284
Wang B, Zhuang Jun (2017) Crisis information distribution on twitter: a content analysis of tweets during hurricane sandy. Nat Hazards 89(1):161–181
Wu T, Liu S, Zhang J, Xiang Y (2017) Twitter spam detection based on deep learning. In: Proceedings of the Australasian computer science week multiconference, ACSW '17, Association for Computing Machinery, New York
Zhou P, Shi W, Tian J, Qi Z, Li B, Hao H, Xu B (2016) Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th annual meeting of the association for computational linguistics (vol 2: Short papers), pp 207–212
Yang Zi, Yang D, Dyer C, He X, Smola A, Hovy E (2016) Hierarchical attention networks for document classification. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 1480–1489,