Detection of spam-posting accounts on Twitter

Isa Inuwa-Dutse[*], Mark Liptrott, Ioannis Korkontzelos

Department of Computer Science, Edge Hill University, UK

Abstract

Online Social Media platforms, such as Facebook and Twitter, enable all users, independently of their characteristics, to freely generate and consume huge amounts of data. While this data is being exploited by individuals and organisations to gain competitive advantage, a substantial amount of data is being generated by spam or fake users. One in every 200 social media messages and one in every 21 tweets is estimated to be spam. The rapid growth in the volume of global spam is expected to compromise research works that use social media data, thereby questioning data credibility. Motivated by the need to identify and filter out spam contents in social media data, this study presents a novel approach for distinguishing spam vs. non-spam social media posts and offers more insight into the behaviour of spam users on Twitter. The approach proposes an optimised set of features independent of historical tweets, which are only available for a short time on Twitter. We take into account features related to the *users* of Twitter, their *accounts* and their *pairwise engagement* with each other. We experimentally demonstrate the efficacy and robustness of our approach and compare it to a typical feature set for spam detection in the literature, achieving a significant improvement on performance. In contrast to prior research findings, we observe that an average automated spam account posted at least 12 tweets per day at well defined periods. Our method is suitable for real-time deployment in a social media data collection pipeline as an initial preprocessing strategy to improve the validity of research data.

Keywords: Social network; Twitter; spam; social media; Twitter microblog; spam detection

[*] Corresponding author

Email addresses: `dutsei@edgehill.ac.uk` (Isa Inuwa-Dutse), `Mark.Liptrott@edgehill.ac.uk` (Mark Liptrott), `Yannis.Korkontzelos@edgehill.ac.uk` (Ioannis Korkontzelos)

1. Introduction

Online social media is one of the defining phenomena in this technology-driven era. Platforms, such as Facebook and Twitter, are instrumental in enabling global connectivity. 2.46 billion users are estimated to be now connected and by the year 2020 one-third of the global population will be connected[1]. Users of these platforms freely generate and consume information leading to unprecedented amounts of data. Several domains have already recognised the crucial role of social media analysis in improving productivity and gaining competitive advantage. Information derived from social media has been utilised in health-care to support effective service delivery [2, 3], in sport to engage with fans [4], in the entertainment industry to complement intuition and experience in business decisions [5] and in politics to track election processes, promote wider engagement with supporters [6] and predict poll outcomes. However, alongside the benefits, the rapid increase in social media spam contents questions the credibility of research based on analysing this data. A report by Nexgate [7] estimates that on average one spam post occurs in every 200 social media posts and a more recent study reports that approximately 15% of active Twitter users are automated bots [8]. The growing volume of spam posts and the use of autonomous accounts (social bots) to generate posts raises many concerns about the credibility and representativeness of the data for research.

In this study, we focus on Twitter and we propose a novel, effective approach to detect and filter unwanted tweets, complementing earlier approaches in this direction [8, 9, 10, 11]. Previous studies rely on historical features of tweets that are often unavailable on Twitter after a short period of time, hence not suitable for real-time use. Our approach utilises an optimised set of readily available features, independent of historical textual features on Twitter. The employed features are categorised as related to the *Twitter account*, the *user* or referring to the *pairwise engagement* between users. A number of machine learning models have been trained. Recursive feature elimination has been employed in order to ascertain the robustness and the discriminative power of each feature. In comparison to an earlier study [9], the proposed features exhibit stronger discriminative power with more consistent performance across the different learning models. Spam posting users exhibit some evasive tactics, such as posting on average of 4 tweets per day, and tricks to balance the follower-followee relationship [9]. Our analysis shows that an average automated spam posting account posts at least 12 tweets per day within well-defined activity periods. The activity pattern resembles the staircase function exhibiting surges of intermittent activities. Our study contributes (a) a new set of lightweight features suitable for real-time detection of spammers on Twitter and (b) an additional dataset source[1] offering an insight into the behaviour of spam users on Twitter to support further studies.

The paper is structured as follows: Section 2 offers a high-level overview of spamming on social media and Section 3 presents a survey of the relevant literature. The Dataset and the feature selection process are presented in Sections 4 and 5, respectively. Section 6 presents the experimental results and Section 7 discusses relevant findings. Finally, Section 8 concludes this work and suggests some directions for further future research.

2. Online social media spamming

Online spamming activities come in different forms such as malware dissemination, posting of commercial URLs, fake news or abusive contents, automated generation of large volume of contents [8] and following or mentioning random users [9]. Another form of online spamming is the growing use of machine learning models to generate fake reviews on products and services [12] and the use of social bots to influence the opinion of users [13]. The volume of global spam is growing tremendously, with an estimated rate of 355% in 2013 [7]. Specifically on Twitter, for every 21 tweets, one is spam and about 15% of active users are autonomous agents, i.e. social bots [8]. The growth rate of spam volume can be attributed to the lack of physical contact between the communicating parties. This makes it difficult to ascertain the actual identity of the user and the legitimacy of the contents being posted. Evidently, utilising data directly from social media platforms without effective filtering may mislead the analysis and lead to wrong conclusions due to unrepresentative data. Numerous sophisticated approaches have been developed in this direction and are reviewed in Section 3. However, at the same time, spammers evolve rapidly to evade detection systems. As a result, some approaches may be rendered obsolete and ineffective in responding to the new tricks introduced by the spammers.

3. Literature Review

Spam entails any form of activity that causes harm or disrupts other online users. The increasing amount of spam tweets can be attributed to humans' inclination to spread misleading information, even if such information originated from unreliable sources, such as a social bot account. Recently,

[1] We are not able to provide the fully-hydrated tweets, i.e. accompanied with full details, due to sharing restrictions but we provide the relevant IDs and computed features.

Vosoughi et al. [14] discover that both genuine and false news spread at equal rate. False news on Twitter spread rapidly. Social bots are deployed to accelerate the process and human users further amplify the content. To detect spam tweets, numerous detection systems have been proposed, using various techniques that are reviewed in this section.

The pioneering work of [15] on spam detection utilised directed graph models to analyse *follower – friend* relationships on Twitter and define feature sets for effective spam detection. In broad context, approaches for spam detection can usually be classified under the following categories: social graph analysis [16, 17, 18], text analysis and activity patterns [19], analysis of user profile meta-data, URL usage and the effect of URL obfuscation [20, 21, 22], analysis of interaction behaviour [23, 8, 9], and URL blacklisting and its effect [24].

Recently, in November 2017, Twitter increased the maximum number of characters in a tweet for most users, after just over a month of testing [25]. Up to that time, users were limited to 140 characters per tweet thereby making URLs and URL shortening services widespread. Thomas [20] and Lee et al. [21] analysed streams of URLs used by spam users and studied how spammers exploit URLs obfuscation to redirect users to malicious sites. Grier et al. [24] analysed a large number of distinct URLs pointing to blacklisted sites due to their involvement in scam, phishing and malware activities. Although the approach is effective, it is often slow and fails to detect URLs that point to malicious sites but have not been blacklisted previously. Gao [19] also studied URL usage on Facebook to detect spamming activity and observed that this form of spamming is mostly associated with compromised accounts rather than accounts created solely for spam activity. Benevenuto [22] studied the statistical properties of user accounts and how URL shortening services affect spam detection mechanisms. However, the universal use of URLs and URL shortening by the vast majority of Twitter users makes it difficult to directly identify potentially nefarious links on a large scale. In general, the use of URLs relies on historical information, limiting the possibilities for real-time detection.

Danezis and Mittal [18] utilised a social network model to infer legitimate user accounts that are being controlled by an adversary. Lee et al. [9] created *social honeypot accounts* mimicking naive Twitter users to entice spam posting users. Users who fall prey by engaging with these accounts are assumed to be in violation of usage policy. Users identified using this method were analysed to distinguish different user types focusing on link payloads and features that can capture the dynamics of *follower-following* networks of users. Varol et al. [8] employed many features related to users, content and the network to develop a system for social bot account detection.

Sedhai and Sun [26] analyse the distribution of spammy words, i.e. terms with higher probability of occurrence in spam than in non-spam tweets, in tweets to detect spam. Chen et al. [27] provides an in depth analysis of deceptive words used by spammers on Twitter. The work of Chao and Chen [28] is motivated by *Twitter Spam Drift*, i.e. the property of statistical features of spam tweets to change over time. *Twitter Spam Drift* is caused because spammers continuously adopt and abolish various evasive tricks. Features related to this phenomenon were utilised in training machine learning classifiers. Li and Liu [29] analysed how the effect of unbalance datasets can be mitigated in detection tasks.

Standard machine learning methods are sometimes considered as inadequate in capturing the variability of spamming behaviour. Wu et al. [30] utilised a deep learning technique based on Word2Vec [31] to capture the variation of spam-related challenges. While it is essential to allow detection models to continuously learn features strong enough to distinguish spam from non-spam, methods that solely rely on textual information are be inadequate to draw the distinction between a habitual spam posting account and a non-spam posting account. Hand-crafted features related to the account and the user need to be considered. In this study, a set of hand-crafted features are leveraged in tandem with features learn by deep neural networks. Features studied by humans and encoded to classifiers can achieve better performance and low false positive rates [32].

The use of a large number of features introduces extra overheads to the detection system, some of which may be unavailable for real-time use. Subrahmanian et al. [13] offer insights into techniques utilised in identifying *influence bots*, i.e. autonomous entities determined to influence discussions on Twitter. Influence bots comprise a category of social bot accounts that seek to assert influence on topical or new discussions thereby generating unrepresentative or fake data.

The surveyed studies on spam detection largely rely on either historical tweets of a user to extract features which contribute to an extra overhead for the detection system [33] or limited features learnt by unsupervised techniques. Our proposed approach relies on readily available features in real-time for better performance and wider applicability.

4. Dataset

This section discusses the collection and validation of datasets utilised in our experiments: *Honeypot*, the automatically annotated *spam-posts detection* dataset *($SPD_{automated}$)* and the manually annotated *spam-posts detection* dataset *(SPD_{manual})*. Table 1 presents statistics about these datasets. The

Dataset	**Size of Original Data**	**Size of Preprocessed Datasets**	**Class**	**Collection**	**Verified?**
Honeypot	19,297	19,276	Legitimate	Automated	No
Honeypot	23,869	22,223	Polluter	Automated	No
$SPD_{automated}$	10,318	8,515	Legitimate	Automated	Yes
$SPD_{automated}$	25,568	9,831	Spam	Automated	No
SPD_{manual}	2,000	1,300	Legitimate	Manual	Yes
SPD_{manual}	2,000	700	Spam	Manual	No

Table 1: Summary of datasets: The size of original data refers to data collected before some preliminary preprocessing steps such as discarding non-English tweets and duplicates.

Honeypot dataset [9] is publicly available and useful for studying spam activity on Twitter. It was utilised both as a dataset per se and for collecting the *SPD* datasets using keywords. Keywords play a crucial role in retrieving specific documents from large corpora [34] and this study speculates that keywords extracted from the *Honeypot* dataset can be used in retrieving large quantities of similar data.

In Table 1, *Legitimate* refers to data from genuine users whose accounts have been verified by Twitter. A verified account is certified by Twitter to be genuine and such information is available from the meta-data section of the tweets. In contrast to the randomised approaches utilised in [9] to ascertain user legitimacy on Twitter, we used accounts verified by Twitter in building the legitimate part of the *SPD* datasets to avoid the potential risk of a high false positive rate.

SPD_{manual} is a manually annotated dataset created to supplement evaluation. It contains tweets randomly selected from the full set of tweets that have been downloaded between February, 2017 and June, 2017 via the traditional Twitter API[2] using relevant keywords as query terms. It consists of 1,700 tweets of *legitimate users* and 300 tweets of *spam users*.

For our analysis, we took a sample of 2000 accounts for manual annotation resulting in the disproportionate ratio of 700:1300 (spam:non-spam). Unbalanced datasets often affect the performance of detection systems [29], including ours. To mitigate that, we applied the SMOTE resampling technique [36] to balance the data by upscaling the minority class. Additionally, we further query the accounts of spam users to retrieve more spam tweets. This technique was used before training Word2Vec. The cost and labour intensiveness of annotations as well as the general unbalanced ratio of spam/non-spam on Twitter contributes to this disproportionate ratio in

[2] The dedicated channel provided by Twitter to enable access to public datasets [35].

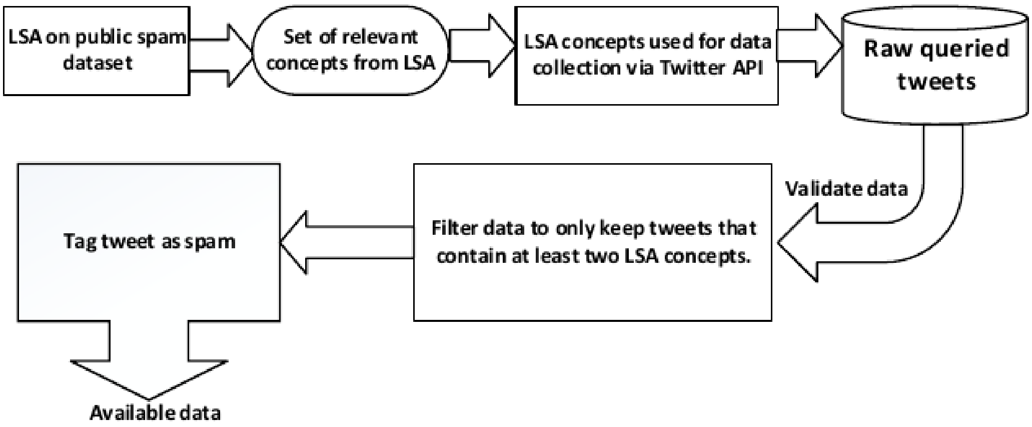

LSA on public spam dataset
Set of relevant concepts from LSA
LSA concepts used for data collection via Twitter API
Raw queried tweets
Validate data
Filter data to only keep tweets that contain at least two LSA concepts.
Tag tweet as spam
Available data

Figure 1: Collection and validation of the spam part of the $SPD_{automated}$ dataset from Twitter

$\mathrm{SPD}_{\mathrm{manual}}$.

$SPD_{automated}$ contains tweets that have been collected between February, 2017 and June, 2017, and have been automatically marked as legitimate or spam. Tweets posted by users whose accounts have been verified as *Legitimate* by Twitter were marked as legitimate. Tweets that contained at least two of the most representative keywords in the *Polluter* part of the *HoneyPot* dataset were marked as spam.

Keywords, both for querying Twitter and validating spam, are obtained by applying Latent Semantic Analysis (LSA) on the *Honeypot* dataset [37]. LSA is useful in capturing the semantics and relevance of terms to a document [38]. Prevalent keywords from LSA concepts include *free, new, lots, win, follow, trade, good, great, make, create, twitter, followers, check,gain, buy, account, get, making, online, want.* See Table A.2 for full list. A block diagram of the collection and validation process is shown in Figure 1. Table 2 shows some example tweets that satisfy this criterion.

4.1. Validation of $SPD_{automated}$

Labelling data in $SPD_{automated}$ as spam is based on the hypothesis that spam users are more likely to use at least two of the terms obtained via LSA on the part of the *Honeypot* dataset that is known to be spam [9]. To validate this, we compute and compare in the legitimate and the spam part of $SPD_{automated}$:

- the distribution of the co-occurring keywords
- lexical richness and lexical density

Id	**Tweet**
T1	RT @user: Retweet to **win** up to 121+ **followers** must be **following** me 💍
T2	Retweet this for 81+ **free follows** ⚜️ ♻️
T3	Retweet for 125 **free follows** 🌈 '\n'Retweet and Fav 💕 this if you have my post notifications on! 🐠 For 125 **free followers** ❣️😱
T4	Watch and like this video for **free** 80 **followers** 💕 url
T5	Retweet to win up to 130+ **free followers** 🍬 ' '@user
T6	RT @user: Retweet this to **gain followers** faster 😍 🍭 🍬
T7	**Follow everyone** who FAV this 🍅
T8	@user @user @user @user @user @user **follow everyone** who likes this 🌸 #SolarEclipse2017 \

Table 2: Examples of collocational bigrams from the spam part of $SPD_{automated}$. Keywords returned by *LSA* on the *Honeypot* dataset are shown in bold face. Actual users mention were replaced with the *'user'* placeholder to preserve anonymity.

	N-gram proportions		
Dataset	**bigrams**	**trigrams**	**four-grams**
$\text{Honeypot}_{\text{spam}}$	B1: 1.26×10^{-3} B2: 3.51×10^{-4}	T1: 1.83×10^{-3} T2: 0.0	F1: 4.40×10^{-4} F2: 0.0
$\text{Honeypot}_{\text{non-spam}}$	B1: 4.07×10^{-4} B2: 3.3×10^{-3}	T1: 2.50×10^{-4} T2: 0.0	F1: : 0.0 F2: 0.0
SPD_{spam}	B1: 6.04×10^{-2} B2: 2.21×10^{-2}	T1: 1.05×10^{-2} T2: 2.87×10^{-2}	F1: 6.42×10^{-3} F2: 4.74×10^{-3}
$\text{SPD}_{\text{non-spam}}$	B1: 2.34×10^{-7} B2: 0.0	T1: 0.0 T2: 0.0	F1: 0.0 F2: 0.0

Table 3: Relative frequencies of n-grams that consist of some spammy words in the dataset; in particular the n-grams B1, B2, T1, T2, F1 and F2, that are shown in bold face in table A.1.

- the distributions of user mentions and URLs

Table 4 shows the results.

4.1.1. Distribution of co-occurring keywords

Spammers heavily leverage certain deceptive words to lure users [27]. Words normally preceded by *free*, *follow* and *gain* have high probability of occurrence in spam tweets [26].

In this study, we aim to capture important *n-grams* used by spammers by leveraging a public spam dataset. To select the best *n-grams* as well as the number of co-occurring terms sufficient for identifying spam tweets, we first apply Latent Semantic Analysis (LSA) as a decomposition technique to discover the most representative keywords in the corpus and compared

Data	**% Name Similarity**	**% digits in names**	**% containing spam bigrams**	**% LexRich unfiltered**	**% LexRich filtered**
Legitimate	82.59	14.07	1.05	97.43	86.74
Spam	26.27	88.84	89.51	90.94	49.46

Table 4: Percentage distributions of relevant metrics computed in the two parts of $SPD_{automated}$, i.e. legitimate and spam

with a list of known spammy words[3]. Based on the list of spammy words, we compute the relative frequencies of various spammy *n-grams* (bigrams, trigrams and four-grams) in the corpus. Table 3 shows the relative frequencies of spammy *n-grams* in various datasets. Figure A.3 shows an example of common *spammy n-grams.* In table 3 we observe that bigrams have higher relative frequencies in spam datasets and the individual terms that they consist of occur in the spammy list, in table A.1. Accordingly, a tweet is highly probable to be a spam if it contains at least bigrams of spammy words and has low lexical richness.

We observe in Table 4 that only 1.05% of the tweets in the legitimate part of $SPD_{automated}$ contain two or more keywords, extracted using *LSA* from the *Polluter* part of the *HoneyPot* dataset. In contrast, more than 89.5% of the tweets in the spam part of $SPD_{automated}$ contain keyword pairs. This distribution is a strong indicator of a probable spam tweet and also minimises the risk of labelling legitimate users as spammers. Table 2 shows examples of frequent co-occurring keywords sampled from $SPD_{automated}$.

4.1.2. Lexical richness and density

In quantitative linguistics, lexical richness measures the wealth of vocabulary in a given text [39]. Basic measures, such as *Type Token Ratio (TTR)* and *Mean Word Frequency*, are utilised to assess the quality of lexicons in spam and non-spam corpora. We hypothesise that spam users will have low lexical diversity and sophistication compared to genuine users. Legitimate users are expected to use rich and diverse lexicons in tweets depending on the discussion topic. In contrast, spam users focus on specific targets such as promoting a certain product or marketing to increase the number of their followers. Users engaging with this behaviour are highly likely to recycle specific sets of similar words.

Type-token ratio (TTR) measures the richness of a lexicon in a document [40]. It is useful in understanding how distinct words are utilised in the legitimate and the spam part of $SPD_{automated}$. For a dataset *D*, *TTR* can be

[3] Compiled by [26] and shown in Table A.1, in the Appendix.

computed as follows:

$$TTR = \frac{\text{unique tokens in D}}{\text{tokens in D}} \tag{1}$$

We also compute *lexical density (LD)* [40] as follows:

$$LD = \frac{\text{words in D excluding stopwords}}{\text{tokens in D}} \tag{2}$$

Table 4 shows the result of computing these metrics in both datasets.

However, lexical richness is insufficient for the purpose, because it does not capture term semantics [41]. Some spammy words are not exclusive to spammers,as non-spam users may also use them in different context. To capture the semantics of words in spam and non-spam datasets, we experimented with word embeddings as classification features. Table 10 shows evaluation results of various classifiers trained on word embedding features and features without word embeddings and tested on $SPD_{automated}$. Table A.3 summarises the datasets used in training our Word2Vec model [31].

4.1.3. Users mention

Random mentioning of users [42] is a common tactic employed by spammers in an effort to expand the visibility or their network of followers [9, 23]. In Table 4, lexical richness, i.e. *%LexRich (unfiltered)*, in the spam set is marginally higher than expected. Noting the high proportion of user mentions in spam data, lexical richness *(% LexRich (filtered)) or lexical density* is computed without considering the *user mentions* and *URLs* in both datasets. The computation in the spam dataset led to a very low score suggesting that the large number of *user mentions* and *URLs* are responsible for the relatively high TTR score in the spam dataset.

TTR in the legitimate dataset is not affected by filtering out *user mentions* and *URLs* and is indicative of the richness and diversity of the lexicon used by genuine users. The low TTR score in the spam dataset indicates that the same words are being used repetitively usually not really matching the discussion topic. Table 4 also shows metrics related to naming conventions by computing the degree of similarity between the username and the screenname of each user and the proportion of digits in their names. This topic is discussed further in Section 5.

5. Features

The Twitter platform facilitates global connections and interactions of diverse users [43]. Figure 2 presents an overview of the platform and its relevant attributes that enable users to connect and form the basis of our feature extraction.

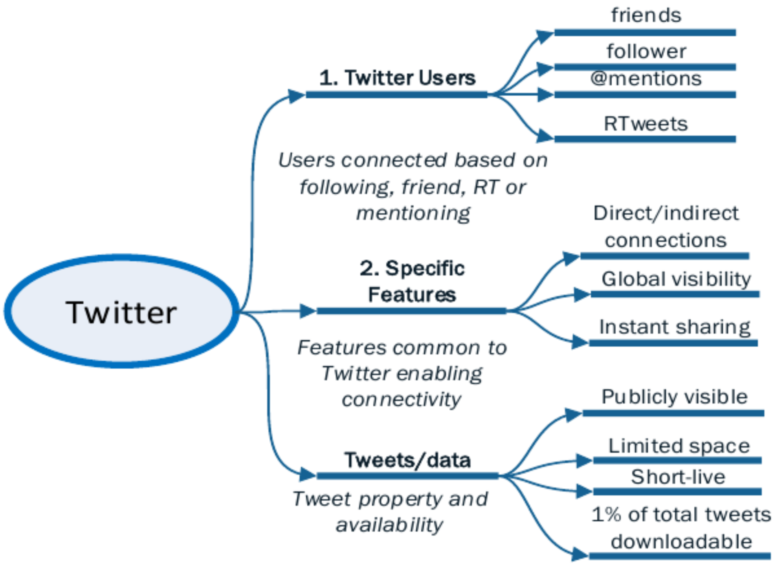

Twitter
1. Twitter Users
Users connected based on following, friend, RT or mentioning
friends
follower
@mentions
RTweets
2. Specific Features
Features common to Twitter enabling connectivity
Direct/indirect connections
Global visibility
Instant sharing
Tweets/data
Tweet property and availability
Publicly visible
Limited space
Short-live
1% of total tweets downloadable

Figure 2: An overview of Twitter: three different categories of attributes that support global interconnectivity of users are shown. The features utilised in this study are derived from these categories directly or indirectly.

5.1. Accessibility, dynamism and categorisations of features

Tweets are available only for a short time, approximately seven days, after being posted. Many real time spam detection systems that rely on historical features from past tweets, are affected by this constraint and may be practically less effective. Readily available, dynamic features offer an enhanced opportunity to distinguish spam from non-spam tweets in real-time. To leverage this potential, features are categorised as follows:

- *User Profile Features (UPF)* include information about the user, such as their *user name*, *screen name*, *location* and *description*
- *Account Information Features (AIF)* consist of information such as *account creation time (account age)* and *account verification flag (verified or not verified)*
- *Pairwise engagement features* subcategorised into:
  - *Engage-with Features (EwF)* include features that describe user activities on Twitter and users can influence or choose how to alter their values. Features under this group include *friends count*, *statuses count*, *tweet type*, *tweet creation time*, *tweet creation frequency*, etc.

– *Engaged-by Features (EbF)* are similar to features in the EwF group . The main difference is that features under this group cannot be influenced by users directly. For instance, a user relies on other users to increase their *favourites count* or to attract more *followers*. Features in this group include *followers count*, *favourites count*, *number of retweet (RT)*, etc.

Furthermore, features can be classified as *basic features* or *derived features*. The aforementioned features, i.e. under *UPF*, *AIF*, *EwF* and *EbF*, are basic features, whereas derived features are computed using two or more basic features or are based on further analysis, e.g. sentiment analysis or entropy computation on textual data. Features can also be characterised as *static or dynamic*. Static features cannot be changed once the account is created e.g. *user ID* and *account creation time*, whereas dynamic features keep changing depending on the user's level of engagements on Twitter e.g. *statuses count*. All features and their properties are shown in Table 5.

5.2. Feature selection

The early work of [43] categorises features for Twitter-based study into *content-based, network-based* and *Twitter specific memes*. These categorisations are further expanded in 2 and were utilised directly or indirectly in previous related studies [15, 20, 21, 22, 8]. Statistical properties of tweet metadata in relation to user, accounts and URLs usage have been effective in spam detection systems [22]. Based on this categorisation, basic features have been analysed for various Twitter-related tasks. For instance, basic features on Twitter have been analysed to detect simple social bots accounts which lack or repeat basic account information such screen names, profile picture [23]. Retweets, user mentions and low reciprocity of friendship [8] or the dynamism of follower-following networks overtime [9] have also been investigated. The sophistication level of automated accounts on Twitter varies from random following and re-tweeting to advanced social bots that actually generate content. Studies that focus on the detection of such accounts rely on basic features on Twitter to define complex ones [13]. Varol et al. [8] developed a detection framework by leveraging numerous features. Our study takes a similar direction by defining a novel set of additional features derived from the basic ones, that have been discussed and exploited in many studies concerning Twitter. The choice of features for experimentation is informed by insights gained from a series of exploratory analysis to understand the distribution of textual features, the composition of data, and the dynamism of features, such as *statuses count*, *friends count*, *followers count*, *favourites* *count*, *naming conventions* and *tweeting patterns*. Figures A.1 and A.2 in the Appendix present further exploratory results.

Account age is useful in capturing the frequency of user activity. From our analysis, accounts with very high statuses and friends count but low favourites count and followers count at young age are likely to be automated spam posting accounts. For example, Figure 3 shows huge amounts of content generated within short period. We utilised these observations in deriving features, such as *Activeness*, *Interestingness* and *Followership*, as shown in Table 5.

Naming conventions: The *Username* and *screenname* of a Twitter user usually exhibit a high degree of similarity. Normally, *screennames* of legitimate users contain segments of the *username*, are not very lengthy and rarely begin with a digit. In some cases, *usernames* of legitimate users contain a reasonably small number of digits in the middle or at the end. In spam accounts, the mix of letters, digits, special characters and unusual symbols is much more widespread. Often, names begin with digits or email addresses and, as shown in Table 4, there is high discrepancy between *usernames* and the corresponding *screenname*. Features, such as *NameSim* and *NamesRatio* in 5, are inspired by this analysis. Other static features in the metadata of a user account on Twitter, such as the *Language* and *Location* fields, may be useful to some extent for identifying spam accounts, due to the fact that most of these fields are either vacant or populated with meaningless content for spam users. Genuine users often report a real location name, but spam posting accounts often return irrelevant content or lengthy and unintelligible sequences of characters or just email addresses.

Tweeting activity and posting behaviour: In an earlier study, spam posting users have been observed to post four tweets per day on average [9]. We observed that an automated spam posting account posts on average at least 12 tweets per day at well-defined periods. Usually, activity levels remain constant within approximately four long-lasting periods. Figures 3 and 4 show examples of spam and legitimate user activity patterns from our June 2017 collection, respectively.

In contrast to automated spam-posting users, a legitimate user of Twitter often follows random usage patterns and takes long breaks of inactivity. Figure 4 represents the activity patterns of two different users with relatively low traffic generation within the same period as the users in Figure 3. Table 5 shows the features proposed for prediction model training, the corresponding feature groups and definitions.

The *VerifiedAccount* feature, labelled as f_{22}, takes on binary values, '1' for verified accounts or '0' otherwise. These values reflect the target labels in the user profile meta-data. The feature was used in the feature set for

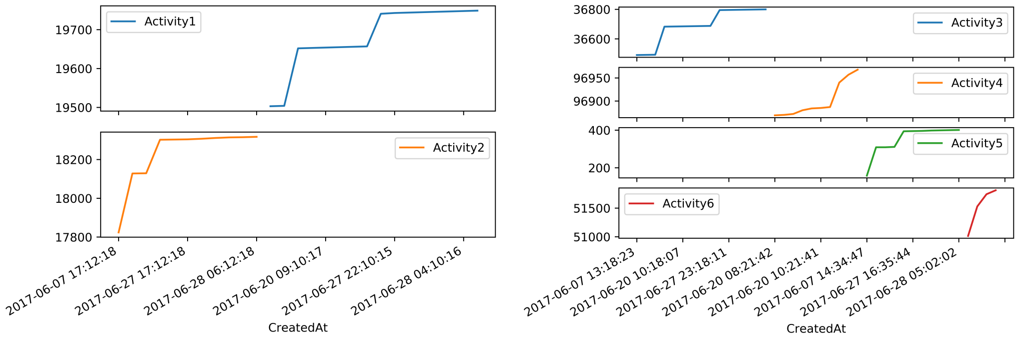


Figure 3: Example of activity patterns of spam-posting social bot accounts. All sub-figures depict hyperactive automated users that generated very high traffic within a short period. The activity distribution over time for most users resembles the *staircase* function. Some users generate much higher traffic than other, e.g. *Activity4* and *Activity6* represent many times more tweets than *Activity5*.

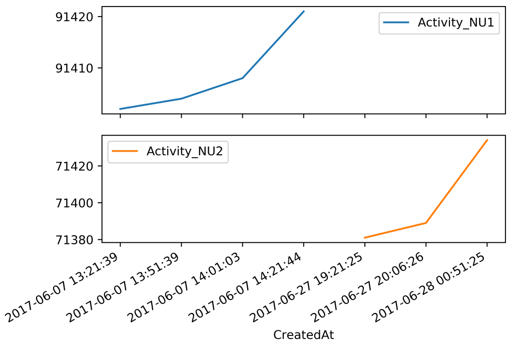


Figure 4: Example of activity patterns of two legitimate users

training classification models during our early experiments. The resulting model overfitted the training data and, for this reason, the feature was later removed due to its role in leaking the correct prediction into the test data [44].

It is crucial for detection models to be able to continuously and automatically learn features strong enough to distinguish spam from non-spam, avoiding handcrafted features. Wu et al. [30] report good performance of a spam detection system that learns suitable features using Word2Vec. However, such methods rely on textual information, only. Social media, including Twitter, offer a wealth of information other than the textual content that are important to draw the distinction between a habitual spam posting account and a non-spam posting account. To improve the classification, we define and experiment with a set of handcrafted features, including features about

Id	Features	Groups	Status	Description/Definition
f_1	*AccountAge*	AIF	static	days since account creation to date of collection
f_2	*FollowersCount*	EbF	dynamic	in user profile meta-data
f_3	*FriendsCount*	EwF	dynamic	in user profile meta-data
f_4	*StatusesCount*	EwF	dynamic	in user profile meta-data
f_5	*DigitsCountInName*	UPF	static	number of digits in screen name
f_6	*TweetLen*	EwF	dynamic	number of characters in tweet
f_7	*UserNameLen*	UPF	static	number of characters in user name
f_8	*ScreenNameLen*	UPF	static	number of characters in screen name
$f_{9,10,11,12}$	*Metric entropy* for all textual features: tweet, user profile description, user name and screen name, respectively	UPF	dynamic	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: $\frac{\sum_{i=1..k} p_i \log_2 p_i}{\lvert x\rvert}$
f_{13}	*URIsRatio*	EwF	dynamic	$\frac{\lvert\text{characters in URLs}\rvert}{\lvert\text{tweet length}\rvert}$
f_{14}	*MentionsRatio*	EwF	dynamic	$\frac{\lvert\text{characters in user mentions}\rvert}{\lvert\text{tweet length}\rvert}$
f_{15}	*NameSim*	UPF	static	% proportion of similarity in User name and Screen name
f_{16}	*LexRichWithUU*	EwF	dynamic	TTR in tweets: $\frac{\lvert\text{token types}\rvert}{\lvert\text{total tokens}\rvert} * 100$
f_{17}	*Friendship*	EwF	dynamic	$\frac{FriendsCount}{FollowersCount}$
f_{18}	*Followership*	EbF	dynamic	$\frac{FollowersCount}{FriendsCount}$
f_{19}	*Interestingness*	EbF	dynamic	$\frac{FavouritesCount}{StatusesCount}$
f_{20}	*Activeness*	EwF	dynamic	$\frac{StatusesCount}{AccountAge}$
f_{21}	*LexRichWithOutUU*	EwF	dynamic	$\frac{\lvert\text{lexical worlds}\rvert}{\lvert\text{total number of words}\rvert} * 100$
f_{22}	*VerifiedAccount**	AIF	static	in tweet metadata
f_{23}	*FavouritesCount*	EwF	dynamic	in user profile meta-data
f_{24}	*NamesRatio*	UPF	static	$\frac{\lvert\text{screenname length}\rvert}{\lvert\text{username length}\rvert}$

Table 5: Features proposed and used in the current study, the corresponding groups and definitions. The *VerifiedAccount* feature, f_{22}, was excluded form our final feature set, because in preliminary experiments it was shown to cause overfitting.

the account and the user that posted each tweet.

Handcrafted features can be used in tandem with features learn by deep neural networks. Our study follows similar approaches to spam detection systems [30, 28, 26] by adopting the unsupervised paradigm. Unsupervised methods effectively counter the effect of *Twitter Spam Drift*, which affect detection systems [30, 28], by capturing the variability of spammer behaviour effectively. Sedhai and Sun [26] used a semi-supervised framework for spam detection at tweet level, whereas Chao and Chen [28] used both traditional machine learning on handcrafted features and deep learning to automatically learn some key features. We experimented with both handcrafted features and features learnt by deep learning models and compare their performance, as shown in table 10. To account for full variability, the more handcrafted features are used, the better the classification performance and the lower the false positive rate [32]. Significant performance improvements were achieved at different levels in our study.

6. Experimentation and results

This section discusses the experimental procedure and the results obtained. All experiments are conducted using the Scikit-learn toolkit [45].

6.1. Parameter tuning and classification models

An effective classifier should be able to correctly classify previously unseen data by leveraging the experience gained from training on n labelled samples, i.e. data instances and the corresponding class. The target of the classification task at hand is to predict spam-posting users or normal legitimate users correctly, by accessing one of their tweets associated with user account meta-data. Effective hyper-parameter tuning is key for significantly improving the performance of machine learning models [46]. Thus, we tuned the hyper-parameter values of all classification models, used in experiments of our study, via grid search on standard 10-fold cross-validation.

6.2. Feature importance and correlation

During an initial analysis stage, a large number of features have been used for training and some features were discarded due to their relatively low contribution to the overall performance. Figures A.1 and A.2 in the appendix provide more evidence about the feature selection process. A recursive feature elimination approach was adopted to measure the contribution of each feature to the overall performance. The results of this process are graphically illustrated in Figure 5.

Correlation analysis plays a crucial role in achieving optimum performance. Features that correlate perfectly introduce redundancy and do not add extra information into classification models [47]. We conducted univariate feature analysis to understand the relevance of each feature in predicting the target class. The results are shown in Figure 6 formatted as a heat-map to visualise as colour intensity the correlation degree of each feature with the

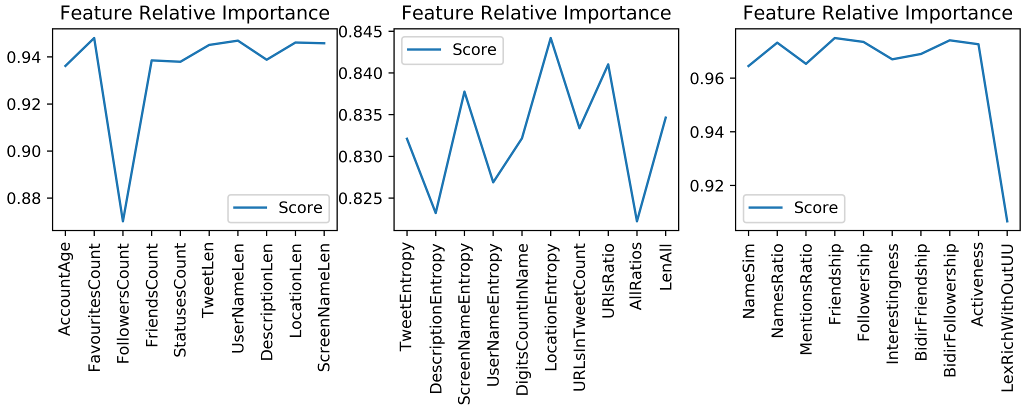


Figure 5: This figure shows the performance of features measured using recursive feature elimination. The most informative feature is the lexical richness of tweets including user mentions and URLs *(LexRichWithUU)*. It contributed significantly to the overall performance, as evidenced from the sharp drop in the figure. The complete set of the features is provided in figures A.1 and A.2 in the appendix.

target class, i.e. *AccountClass*, and with other features. With the exception of *lexical richness*, *LexRichWithUU*, and *lexical density*, *LexRichWithOutUU*, which are derived from same root, there is no other pair of features with perfect correlation. Thus, the features shown in 6 comprise our feature set for all experiments in this paper[4]. The main diagonal of the heatmap matrix represents perfect correlation because each feature is correlated with itself. The column of the target *(AccountClass)* shows the intensity of the correlation of each feature with the target.

6.3. Performance metrics

For evaluation, different metrics are utilised in order to avoid any type of bias towards the majority class, especially when the dataset is imbalanced [48]. In particular, we use the following metrics to summarise experimental results: *F-score*, *Precision*, *Recall*, *Accuracy*, the *Receiver Operating Characteristics (ROC) curve* and the *area under the ROC curve (AUC)*. *F-score*, the geometric mean of *Precision* and *Recall*, captures a model's prediction quality especially in sensitive areas, by requiring both *Precision* and *Recall* to be high. The *AUC* offers a more encompassing metric, insensitive to the imbalance between classes that sometimes provides better evaluation than

[4] In the preliminary stages of this study, we experimented with many more features, mainly derived as combinations of the features in figure 6. Most of these features were discarded due to correlating almost perfectly with others and, thus, not contributing to the accuracy of the model.

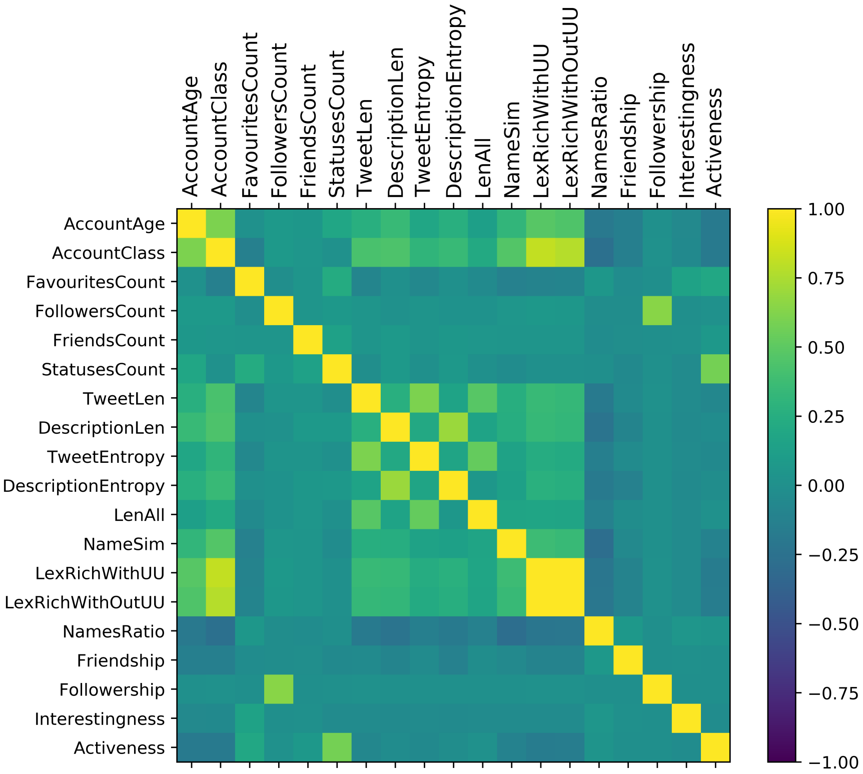


Figure 6: Visual representation of the univariate analysis of correlation of each feature with the target, i.e. *AccountClass* and other features. Correlation magnitudes range from 1 to -1, with 1 denoting perfect positive correlation, 0 no correlation and -1 perfect negative correlation. Features highly correlated with the target constitute the optimum features set.

accuracy [49]. Specifically, the higher the AUC score, the larger the area under the curve, well above the diagonal, e.g. Figure 7.

6.4. Experimental results

We conducted a series of experiments with different classification models and assessed them using various metrics, as discussed in Section 6.3. Our first experiment, aimed to investigate the effectiveness of the proposed features, which we called *Spam Post Detection (SPD)* features, and compared the suitability of different classification models for the task at hand. We trained six different classification models: Maximum-Entropy (MaxEnt), Random Forest, Extremely Randomized Trees (ExtraTrees), C-Support Vector Clas-

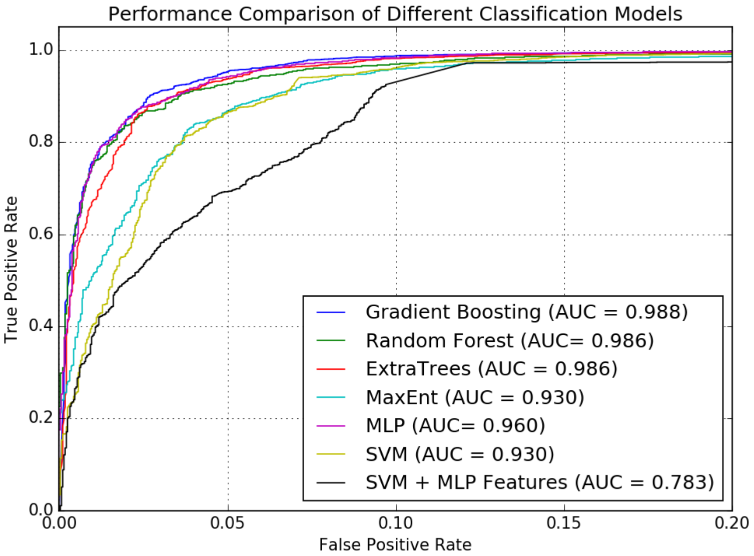


Figure 7: Performance of different classification models evaluated on the $SPD_{automated}$ dataset using 10-fold cross-validation.

sification (SVC[5]), Gradient Boosting and Multi-layer Perceptron (MLP). We also included additional model i.e. *SVM + MLP* which utilises the features learnt by the MLP during training as input for training regime. Figure 7 shows the learning curves and corresponding AUC scores achieved by each model on the best hyper-parameter values, as explained in Section 6.1. All models were trained and evaluated on the $SPD_{automated}$ dataset using 10-fold cross-validation. The chart shows relative consistency in terms of performance across the different classification models, which can be attributed to the effectiveness of the proposed *SPD* features. *Gradient Boosting* is chosen for subsequent use in our next experiments due to its higher performance.

Our second experiment compared the features proposed in this study, *SPD* features, with the *Honeypot* features, proposed in Lee et al. [9]. Since the study of Lee et al. [9] is our main baseline, we compared the two feature sets on the *Honeypot* dataset and the $SPD_{automated}$ dataset, using 10-fold cross validation. The associated learning curves are shown in Figures 8 and 9, respectively. The figures show that *SPD* features perform better than the *Honeypot* features for both datasets. The improvement is small for the *Honeypot* dataset, whereas it is significant for the $SPD_{automated}$ dataset

[5] Which is based on Support Vector Machines (SVM). SVM and SVC used interchangeably in this study.

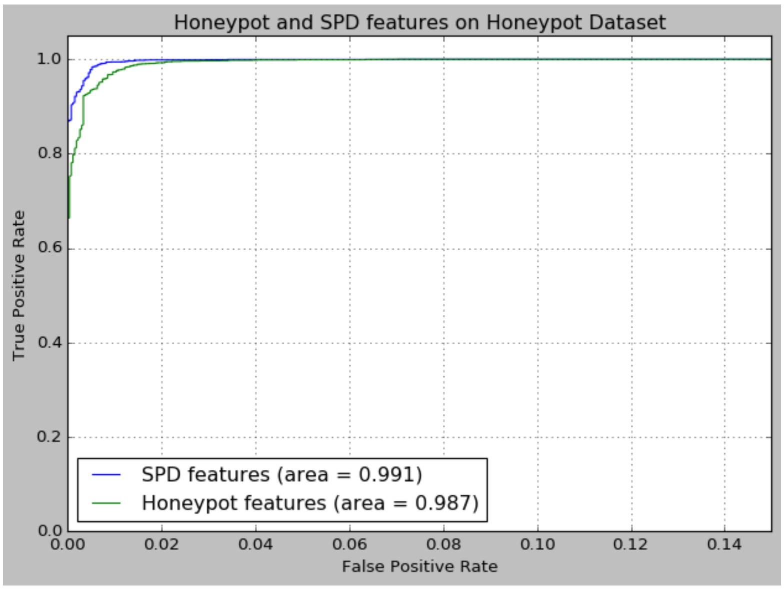


Figure 8: Learning curves of the *SPD* features and the Honeypot features on the *Honeypot* dataset [9]. The *SPD* features achieve a slight improvement in performance.

It should be noted that the *Honeypot* dataset does not provide enough information for computing all *SPD* features. As a result, the *SPD* features line in Figure 8 is based on some *SPD* features, only. Features such as *Interestingness*, *Activeness*, *NameSim* and *Lexical Richness* are not used in this experiment. The lack of these features explains why the improvement in performance is minimal.

In addition to the univariate correlation analysis of features, we investigated the importance of features groups. Table 6 shows the features grouped into three distinct groups: account, users and network features. In the additional experiment with Word2Vec, features learnt by the trained Word2Vec model and some handcrafted features from the study are utilised, in particular lexical richness, activeness and interestingness. Tables 7, 8 and 9 present experiment results for for *Honeypot*, $SPD_{automated}$ *and* SPD_{manual}, respectively on various feature groups. Best performing features in each group constitute the optimum set of features i.e. $SPD_{selected}$) for improved effectiveness and efficiency.

Similarly, Table 10 shows the performance of various classifiers on including or excluding Word2Vec features tested on $SPD_{automated}$. Combining Word2Vec features and lexical richness features performs significantly better than the *Honeypot features* baseline. The combination performs slightly worse that the optimised feature set but uses a much smaller number of features.

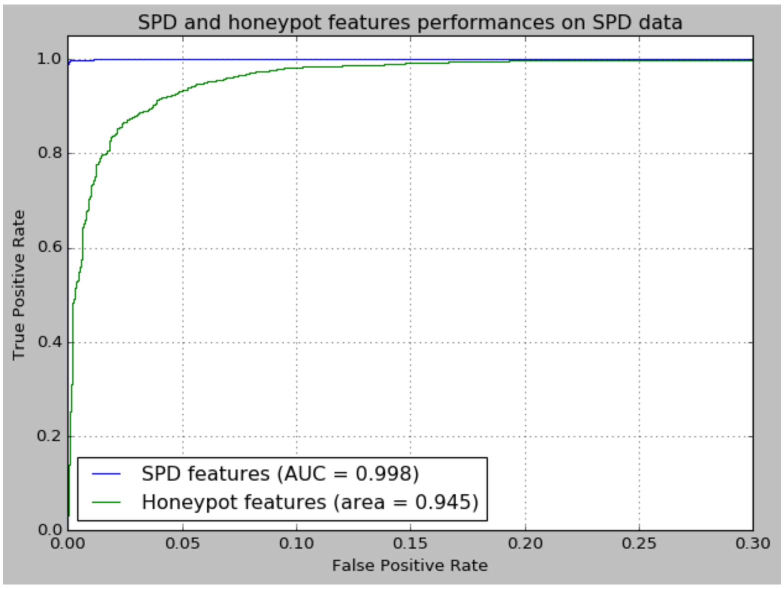


Figure 9: Learning curves of the *SPD* features and the Honeypot features [9] on the $SPD_{automated}$ dataset. The *SPD* improve performance significantly.

Feature Group	**Features**			
Account	AccountAge	DescriptionLen	LocationLen	LocationEntropy
	DescriptionEntropy			
User	UserNameLen	ScreenNameLen	AllRatios	LexicalRichness
	TweetLen	URLsInTweetLen	Activeness	TweetEntropy
	LenAll	StatusesCount	URLsRatio	NamesRatio
	PosSentiment	NegSentiment	OverallSent	DescPosSent
	DescNegSent	DescSentiment	UserNameEntropy	
	SingleHashtagInTweetLen		ScreenNameEntropy	
Pairwise (Network)	*Engaged with:*		*Engaged by:*	
	FriendsCount	MentionsInTweetLen	FollowersCount	Interestingness
	MentionsRatio	HashtagsInTweetLen	Followership	BidirFriendship
	Friendship	HashtagsRatio	BirdirFollowership	
Optimised	AccountAge	FollowersCount	TweetLen	TweetEntropy
	Friendship	StatusesCount	LenAll	FriendsCount
	NamesRatio	Interestingness	NameSim	Followership
	Activeness	LexRichWithUU	DescriptionEntropy	
	BirdirFollowership			

Table 6: All Features and respective feature sets

Classifier	Features	Accuracy %	AUC %	Precision % $(0,1)$	Recall % $(0,1)$	F-score % $(0,1)$
Random Forest	*Honeypot*	**94.70**	96.19	(90, 93)	**(91, 94)**	(92, 92)
	$SPD_{Selected}$	94.68	**96.38**	(93, 91)	**(92, 93)**	**(93, 92)**
ExtraTrees	*Honeypot*	93.74	**96.37**	**(91, 91)**	**(92, 89)**	**(91, 90)**
	$SPD_{Selected}$	**93.86**	95.32	(89, 90)	(91, 89)	(90, 90)
Gradient Boosting	*Honeypot*	98.53	98.55	(99, 98)	(98, 99)	(99, 98)
	$SPD_{Selected}$	**98.93**	**98.94**	**(99, 99)**	**(99, 99)**	**(99, 99)**
MaxEnt	*Honeypot*	83.57	83.62	(86, 81)	(83, 84)	(84, 83)
	$SPD_{Selected}$	**85.99**	**86.21**	**(90, 82)**	**(83, 89)**	**(86, 86)**
MLP	*Honeypot*	89.53	89.54	(91, 88)	(89, 90)	(90, 89)
	$SPD_{Selected}$	**93.70**	**90.28**	**(91, 90)**	**(92, 89)**	**(91, 90)**
SVM	*Honeypot*	86.26	86.29	(88, 84)	(86, 87)	(87, 85)
	$SPD_{Selected}$	**88.13**	**88.21**	**(90, 86)**	**(86, 90)**	**(88, 88)**
SVM + MLP Features	*Honeypot*	87.57	87.62	(90, 85)	(87, 88)	(88, 87)
	$SPD_{Selected}$	**89.08**	**89.09**	**(90, 88)**	**(89, 89)**	**(89, 89)**

Table 7: Evaluation results of all combinations of classifiers and feature sets applied on the *Honeypot* dataset. '(0, 1)' denotes performance on the spam part and the legitimate part of each dataset, respectively.

To address the imbalance in the SPD_{manual} dataset, we utilised the *SMOTE* technique [36], which up-samples the minority class during training the classifier. We observe that the set of features proposed in this paper, *SPD*, performs better than the *Honeypot* [9] on all datasets. The lightweight version of *SPD* features, as computed by the feature selection process in Section 6.2, perform better than the *Honeypot* feature set when applied on $SPD_{automated}$ but worse than the *Honeypot* feature set when applied on *Honeypot* and SPD_{manual}. The lightweight version of *SPD* features consistently perform worse than the full *SPD* feature set, as expected.

6.5. Error analysis

Error analysis is carried-out to investigate cases that were not classified correctly by the classification model. In this section, we discuss the reasons that may have led to misclassification of some representative samples, shown in Figure 11

In the study dataset that was used to design the *SPD* features proposed in this study only tweets in English were considered. As a result, some tweets in the *SPD* dataset, such as tweet #1 in Table 11, were not in English and were misclassified. This can be attributed to the fact that although the original language field in the meta-section of some user profiles was set as English, the actual interaction language in the tweet is not English.

As shown in Figure 5, lexical richness and density are important classification features. The occurrence of irrelevant tokens in a tweet, which were

Classifier	**Features**	**Accuracy** %	**AUC** %	**Precision** % $(0,1)$	**Recall** % $(0,1)$	**F-score** % $(0,1)$
Random Forest	*Honeypot*	90.71	96.28	(91, 91)	(92, 89)	(92, 90)
	$SPD_{Account}$	80.90	85.74	(86, 61)	(90, 51)	(88, 56)
	SPD_{User}	85.81	91.37	(91, 69)	(91, 69)	(91, 69)
	$SPD_{Network}$	92.06	96.70	(95, 83)	(95, 82)	(95, 82)
	$SPD_{Optimised}$	**98.46**	**98.87**	**(99, 99)**	**(99, 99)**	**(99, 99)**
	SPD all	94.41	98.13	(96, 88)	(96, 87)	(96, 88)
ExtraTrees	*Honeypot*	90.57	96.32	(91, 91)	(92, 89)	(91, 90)
	$SPD_{Account}$	80.53	85.85	(86, 60)	(89, 54)	(87, 57)
	SPD_{User}	86.22	91.48	(89, 73)	(94, 61)	(91, 67)
	$SPD_{Network}$	91.99	96.51	(94, 83)	(94, 81)	(94, 82)
	$SPD_{Optimised}$	**98.63**	**99.89**	**(100, 97)**	**(97,100)**	**(99, 98)**
	SPD_{all}	93.78	98.09	(96, 87)	(96, 87)	(96, 87)
Gradient Boosting	*Honeypot*	94.93	94.94	(96, 94)	(95, 95)	(95, 95)
	$SPD_{Account}$	82.17	87.13	(85, 66)	(93, 46)	(89, 54)
	SPD_{User}	85.74	91.82	(88, 76)	(94, 57)	(91, 65)
	$SPD_{Network}$	91.62	96.41	(93, 85)	(96, 77)	(95, 81)
	$SPD_{Optimised}$	**98.97**	**99.93**	**(99, 98)**	**(98, 99)**	**(98, 99)**
	SPD_{all}	93.60	97.96	(96, 88)	(97, 85)	(96, 87)
MaxEnt	*Honeypot*	84.59	84.65	(87, 82)	(84, 86)	(85, 84)
	$SPD_{Account}$	80.93	69.45	(86, 60)	(90, 48)	(88, 54)
	SPD_{User}	81.00	68.56	(85, 61)	(91, 46)	(88, 52)
	$SPD_{Network}$	85.37	72.35	(86, 84)	(97, 48)	(91, 61)
	$SPD_{Optimised}$	**97.12**	**97.13**	**(98, 96)**	**(97, 98)**	**(97, 97)**
	SPD_{all}	91.54	87.67	(94, 82)	(94, 81)	(94, 82)
MLP	*Honeypot*	89.34	89.40	(91, 87)	(89, 90)	(90, 89)
	$SPD_{Account}$	81.85	74.49	(87, 63)	(89, 60)	(88, 62)
	SPD_{User}	85.51	79.62	(91, 69)	(91, 69)	(91, 69)
	$SPD_{Network}$	91.40	86.84	(94, 83)	(95, 78)	(94, 81)
	$SPD_{Optimised}$	**98.42**	**98.43**	**(99, 98)**	**(98, 99)**	**(98, 98)**
	SPD_{all}	94.17	91.83	(96, 87)	(96, 88)	(96, 88)
SVM	*Honeypot*	86.38	86.39	(88, 85)	(86, 87)	(87, 86)
	$SPD_{Account}$	81.22	71.50	(87, 60)	(89, 54)	(88, 57)
	SPD_{User}	82.08	68.54	(85, 68)	(94, 43)	(89, 53)
	$SPD_{Network}$	87.33	75.12	(87, 89)	(98, 52)	(92, 62)
	$SPD_{Optimised}$	**97.35**	**97.38**	**(98, 97)**	**(97, 98)**	**(97, 97)**
	SPD_{all}	91.50	88.82	(95, 79)	(94, 83)	(94, 81)
SVM + MLP Features	*Honeypot*	88.21	88.23	(90, 87)	(88, 89)	(89, 88)
	$SPD_{Account}$	80.69	69.04	(85, 60)	(91, 47)	(88, 53)
	SPD_{User}	84.38	74.99	(88, 70)	(92, 58)	(90, 63)
	$SPD_{Network}$	90.24	83.43	(92, 85)	(96, 71)	(94, 77)
	$SPD_{Optimised}$	**97.71**	**97.74**	**(99, 97)**	**(97, 98)**	**(98, 98)**
	SPD_{all}	93.70	90.95	(96, 85)	(96, 85)	(96, 85)

Table 8: Evaluation results of all combinations of classifiers and feature sets applied on the $SPD_{automated}$ dataset. ‘(0, 1)’ denotes performance on the spam part and the legitimate part of each dataset, respectively.

Classifier	**Features**	**Accuracy** %	**AUC** %	**Precision** % $(0,1)$	**Recall** % $(0,1)$	**F-score** % $(0,1)$
Random Forest	*Honeypot*	93.03	93.11	(91, 89)	(89, 90)	(91, 90)
	$SPD_{Account}$	77.16	79.98	(75, 77)	(74, 78)	(75, 76)
	SPD_{User}	84.29	92.89	(83, 84)	(84, 84)	(85, 85)
	$SPD_{Network}$	95.43	99.74	(92, 94)	(94, 92)	(93, 95)
	$SPD_{Optimised}$	**97.79**	**98.03**	**(94, 98)**	**(98, 94)**	**(97, 97)**
	SPD_{all}	96.29	97.97	(93, 99)	(99, 93)	(96, 96)
ExtraTrees	*Honeypot*	**99.26**	**99.24**	**(99,100)**	**(100, 98)**	**(99, 99)**
	$SPD_{Account}$	75.43	79.49	(73, 78)	(78, 72)	(71, 75)
	SPD_{User}	83.54	91.86	(85, 88)	(84, 90)	(82, 84)
	$SPD_{Network}$	95.80	97.97	(94, 96)	(97, 93)	(96, 96)
	$SPD_{Optimised}$	97.29	99.95	(94, 98)	(98, 93)	(97, 97)
	SPD_{all}	97.90	98.90	(96, 99)	(73, 78)	(98, 98)
Gradient Boosting	*Honeypot*	89.38	59.19	(35, 93)	(23, 96)	(27, 94)
	$SPD_{Account}$	78.13	79.40	(76, 78)	(78, 75)	(77, 76)
	SPD_{User}	87.39	93.45	(85, 90)	(91, 84)	(88, 87)
	$SPD_{Network}$	89.99	95.83	(87, 95)	(96, 83)	(91, 89)
	$SPD_{Optimised}$	**96.08**	**99.88**	**(97, 99)**	**(99, 96)**	**(98, 97)**
	SPD_{all}	93.20	98.22	(89, 98)	(99, 86)	(94, 93)
MaxEnt	*Honeypot*	72.93	73.02	(76, 70)	(69, 78)	(72, 74)
	$SPD_{Account}$	60.82	60.82	(60, 61)	(61, 61)	(61, 61)
	SPD_{User}	67.37	67.39	(77, 63)	(49, 86)	(60, 72)
	$SPD_{Network}$	55.01	56.09	(64, 52)	(32, 80)	(43, 63)
	$SPD_{Optimised}$	**75.40**	**75.51**	**(79, 72)**	**(70, 81)**	**(74, 76)**
	SPD_{all}	75.40	75.51	(79, 72)	(70, 81)	(74, 76)
MLP	*Honeypot*	82.58	82.43	(84, 82)	(81, 80)	(81, 80)
	$SPD_{Account}$	69.72	69.59	(70, 69)	(73, 66)	(71, 68)
	SPD_{User}	80.22	80.25	(82, 78)	(77, 83)	(80, 81)
	$SPD_{Network}$	62.42	62.29	(63, 62)	(57, 68)	(60, 65)
	$SPD_{Optimised}$	**82.94**	**82.95**	**(85, 81)**	**(83, 83)**	**(84, 82)**
	SPD_{all}	92.58	92.65	(89, 97)	(97, 88)	(93, 92)
SVM	*Honeypot*	73.81	73.79	(73, 74)	(73, 74)	(73, 74)
	$SPD_{Account}$	66.01	66.79	(60, 76)	(82, 51)	(70, 61)
	SPD_{User}	73.30	72.92	(80, 69)	(60, 86)	(68, 77)
	$SPD_{Network}$	58.84	58.36	(62, 57)	(77, 40)	(49, 66)
	$SPD_{Optimised}$	**75.65**	**75.58**	**(79, 73)**	**(69, 82)**	**(77, 74)**
	SPD_{all}	80.47	80.46	(81, 80)	(81, 80)	(81, 80)
SVM + MLP Features	*Honeypot*	73.98	74.13	(76, 77)	(77, 75)	(73, 74)
	$SPD_{Account}$	63.54	63.69	(61, 67)	(67, 61)	(63, 64)
	SPD_{User}	71.32	70.96	(77, 68)	(58, 84)	(66, 75)
	$SPD_{Network}$	59.46	59.04	(62, 58)	(43, 75)	(51, 66)
	$SPD_{Optimised}$	**76.64**	**76.58**	**(79, 75)**	**(72, 81)**	**(75, 78)**
	SPD_{all}	87.64	87.49	(85, 91)	(92, 83)	(89, 87)

Table 9: Evaluation results of all combinations of classifiers and feature sets applied on the SPD_{manual} dataset. '(0, 1)' denotes performance on the spam part and the legitimate part of each dataset, respectively.

Classifier	Features	Accuracy %	AUC %	Precision % $(0, 1)$	Recall % $(0, 1)$	F-score % $(0, 1)$
Random Forest	$SPD_{Word2Vec}$	94.95	**99.05**	(95, 95)	(95, 95)	(95, 95)
	$SPD_{Optimised}$	98.46	**99.87**	(99, 99)	(99, 99)	(99, 99)
ExtraTrees	$SPD_{Word2Vec}$	95.47	**99.34**	(96, 95)	(96, 95)	(96, 95)
	$SPD_{Optimised}$	98.63	**99.89**	(100, 97)	(97, 100)	(99, 98)
Gradient Boosting	$SPD_{Word2Vec}$	95.04	**99.09**	(95, 95)	(95 ,95)	(95, 95)
	$SPD_{Optimised}$	98.72	**99.93**	(99, 98)	(98, 99)	(98, 99)
MaxEnt	$SPD_{Word2Vec}$	89.03	**89.14**	(92, 86)	(87, 91)	(89, 89)
	$SPD_{Optimised}$	97.12	**97.13**	(98, 96)	(97, 98)	(97, 97)
MLP	$SPD_{Word2Vec}$	94.40	**94.43**	(96, 93)	(93, 96)	(94, 94)
	$SPD_{Optimised}$	98.42	**98.43**	(99, 98)	(98, 99)	(98, 98)
SVM	$SPD_{Word2Vec}$	89.91	**90.01**	(93, 87)	(87, 93)	(90, 90)
	$SPD_{Optimised}$	97.35	**97.38**	(98, 97)	(97, 98)	(97, 97)
SVM + MLP Features	$SPD_{Word2Vec}$	92.08	**92.24**	(96, 88)	(88, 96)	(92, 92)
	$SPD_{Optimised}$	97.71	**97.74**	(99, 97)	(97, 98)	(98, 98)

Table 10: Evaluation results of Word2Vec features in comparison with our optimised set of features for all classifiers. The Word2Vec feature group contains features learnt by the Word2Vec model, and some handcrafted features *lexical richness*, *activeness* and *interestingness*. '(0, 1)' denotes performance on the spam part and the legitimate part of the dataset, respectively.

Id	Tweet
1	gain followers 👉@ . . . . 🍅💰アメリカとカナダでまっています。\n地域社会に溶けめずにいた民が、ディナ会での出会.
2	retweet this 👍✅
3	like this 👍✅
4	follow like & retweet 👍✅
5	follow back follow you 👍✅
6	gain followers 👍. . . ', 68), gain followers 👉. . . ', this 👀; retweet this 👍✅

Table 11: Sample tokens from misclassified tweets

regarded as unique, leads to a richer lexicon, which in turn increases the chance of classifying the tweet as legitimate. Tweets #2-#6 in Table 11 contain some irrelevant symbols, which were counted as unique, increased the corresponding lexical score and misled the classifier. Emoticons are also a source of confusion for the classifier, especially when computing the lexicon of unique tokens for a tweet and its similarity to lexicons of other tweets.

7. Discussion

This section presents an additional analysis of the manual annotations in the SPD_{manual} dataset, a description of the different user groups and discusses the distribution of relevant features in the dataset.

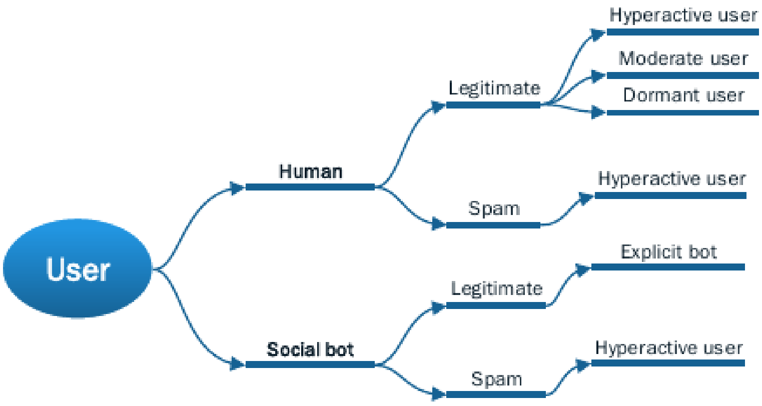


Figure 10: User types in the SPD_{manual} dataset

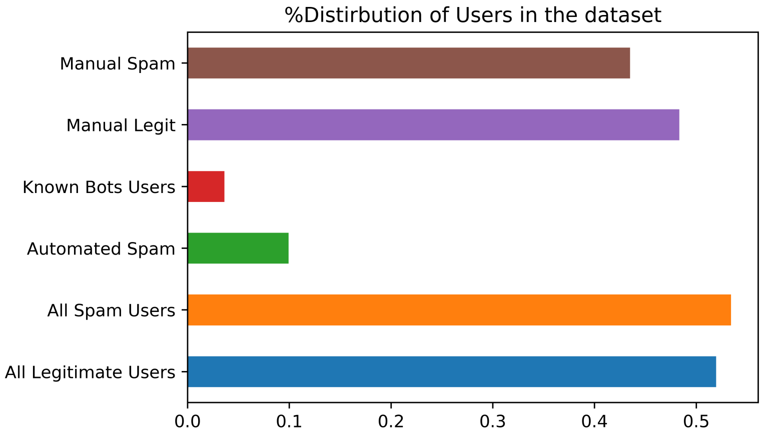


Figure 11: Distribution of different users in the SPD_{manual} dataset. Known bots are accounts that mention the word 'bot' explicitly as part of their name and share some basic features similarities with normal users such as the level of name similarity *(NameSim)*. Known bots in the dataset account for less than 10% of all users.

7.1. Characterising users

A thorough inspection of the tweets in the spam and legitimate parts of the SPD_{manual} dataset suggests that there are two kinds of users on Twitter: *human* users and *social bot (autonomous entity)* users. Each user type consists of a legitimate (non-spam) and a spam part, as depicted in Figure 10, with the following characteristics:

7.1.1. Legitimate users

Legitimate users interact with moderate frequency, within the reasonable and acceptable Twitter usage policy. This user group also contains *genuine* *multiple users*, i.e. accounts managed by organisations or *useful social bots.* Users in this group tend to show a proportionate interaction level and *(activeness)*, i.e. their statuses count matches their account age and the tweets they post are of interest to followers, hence exhibit high *interestingness.* Followers of users in this group often outnumber friends, sometimes even by twice as much. This is expected, since most users subscribe or follow an account due to their interest in it.

The *username* and *screenname* of useful social bot accounts often contain the word *'bot'* as part of name, e.g. *AIBigDataCloudIoTBot* and *Troll Bot.* In some cases, groups of *screennames* share the same suffix separated by the underscore character from a description of the account. Accounts in this group achieve relatively high *interestingness* levels and an almost equal proportion of friends and followers. They also exhibit moderate similarity between their *username* and *screenname* and use a wide variety of words and expressions, i.e. diverse lexicons.

7.1.2. Spam-posting users

Spam-posting users are hyperactive and generate irrelevant content, potentially offensive to other users and in violation of Twitter's terms of use[6]. Accounts in this group exhibit very low *interestingness* and disproportionate *activeness* levels i.e. the statuses count does not match the account age indicating that they employ flooding techniques. Friends of users in this group usually outnumber followers. The interaction patterns of spam-posting social bot accounts are often randomised rather than well-defined, as shown in Figure 3. There is also a high level of inconsistency in naming conventions and a high dissimilarity between *usernames* and corresponding *screennames.* The *screenname* of spam-posting social bot accounts is often unintelligible, mostly containing digits and special characters. Spam-posting users also exhibit low lexicon richness due to the high proportion of URLs, retweets, and user mentions. Spam users generally engage in subscribing to different conversations on Twitter (based on hashtags) and generate tweets not related to the topic of discussion. Figure 11 shows a summary of user groups in Twitter, human and social bot, legitimate and spam-posting. The filtering mechanism developed in this study was succesfully applied in the work of [50] to detect and remove irrelevant posts from spam and automated accounts.

[6] Detailed in [42].

8. Conclusion and future work

This study offers an effective method for spam detection and new insights into the sophisticatedly evolving techniques for spamming on Twitter. The proposed spam detection method utilised an optimised set of readily available features. Being independent of historical tweets which are often unavailable on Twitter makes them suitable for real-time spam detection. The efficacy and robustness of the proposed features set is shown by testing a number of machine learning models and on dataset collected orthogonally from the study data. Performance is consistent across the different models and there is significant improvement over the baseline. It was also shown that automated spam accounts follow a well-defined pattern with surges of intermittent activities. The proposed spam tweet detection approach can be applied in any real-time filtering application. For example, it is applicable to data collection pipelines to filter out irrelevant content at an early pre-processing stage to ensure the quality and representativeness of research data. The combination of handcrafted features and features learnt in an unsupervised manner using word embeddings is shown to significantly improve baseline performance and to perform comparably to the best performing feature set using a smaller number of features.

During the analysis of the data, we observed that spam users tend to be selective in following other users thereby forming enclaves of spammers. This is a high-level observation that we aim to explore further in the future. Additionally, both the two broad user groups, i.e. human users and social bot (autonomous entity) users contain spammers, whose spamming behaviour tends to be similar. The distinction between legitimate human users vs. legitimate social bots as well as human spammers vs. social bot spammers needs to be investigated further. Another interesting dimension for future work is to study the effect of the recent increase in the maximum length of tweets [25] on spamming activity. Intuitively, automated spam accounts will face difficulties in generating lengthier tweets intelligently, thereby making these tweets easier to identify.

Acknowledgements

The authors would like to thank Prof. Francesco Rizzuto for the fruitful discussions and exchange of ideas about a multitude of aspects related to social media, spam content and the motives of spammers. The third author has participated in this research work as part of the CROSSMINER Project, which has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 732223.

References

[1] Social Media Statistics and Facts, Online: www.statista.com/topics/1164/social-networks, Accessed: 18-02-2018.

[2] E. Rojas, J. Munoz-Gama, M. Sepúlveda, D. Capurro, Process mining in healthcare: A literature review, Journal of Biomedical Informatics 61 (2016) 224–236. doi:10.1016/j.jbi.2016.04.007.

[3] K. C. Yee, E. Miils, C. Airey, Perfect Match? Generation Y as Change Agents for Information Communication Technology Implementation in Healthcare, Studies in Health Technology and Informatics 136 (2008) 496–501.

[4] T. Davenport, Analytics in Sports: The New Science of Winning, Tech. rep., International Institute for Analytics, White paper (2014).

[5] Deloitte, Social Analytics in Media Entertainment the Three-minute Guide, Tech. rep., Deloitte Development LLC (2014).

[6] D. Contractor, B. Chawda, S. Mehta, L. Subramaniam, T. A. Faruquie, Tracking Political Elections on Social Media: Applications and Experience, in: Proceedings of the 24th International Conference on Artificial Intelligence, AAAI Press, 2015, pp. 2320—-2326.

[7] NexGate, State of Social Media Spam Research report, Online, Accessed: 18-02-2018 (2013).

[8] O. Varol, E. Ferrara, C. A. Davis, F. Menczer, A. Flammini, Online Human-Bot Interactions: Detection, Estimation, and Characterization, in: International AAAI Conference on Web and Social Media, AAAI Press, 2017, pp. 280–289.

[9] K. Lee, B. D. Eoff, J. Caverlee, Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter, in: Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, 2011, pp. 185–192.

[10] M. Alsaleh, A. Alarifi, F. Al-Quayed, A. S. Al-Salman, Combating Comment Spam with Machine Learning Approaches, in: 14th IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 2015, pp. 295–300.

[11] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, F. Menczer, BotOrNot: A System to Evaluate Social Bots, in: Proceedings of the 25th International Conference on World Wide Web (WWW), Companion Volume, Montreal, Canada, 2016, pp. 273–274.

[12] Y. Yao, B. Viswanath, J. Cryan, H. Zheng, B. Y. Zhao, Automated Crowdturfing Attacks and Defenses in Online Review Systems, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), Dallas, TX, USA, 2017, pp. 1143–1158.

[13] V. S. Subrahmanian, A. Azaria, S. Durst, V. Kagan, A. Galstyan, K. Lerman, L. Zhu, E. Ferrara, A. Flammini, F. Menczer, The DARPA Twitter Bot Challenge, IEEE Computer 49 (6) (2016) 38–46.

[14] S. Vosoughi, D. Roy, S. Aral, The spread of true and false news online, Science 359 (6380) (2018) 1146–1151.

[15] A. H. Wang, Don't follow me: Spam detection in twitter, in: Security and cryptography (SECRYPT), proceedings of the 2010 international conference on, IEEE, 2010, pp. 1–10.

[16] C. Yang, R. Harkreader, J. Zhang, S. Shin, G. Gu, Analyzing Spammers' Social Networks for Fun and Profit: A Case Study of Cyber Criminal Ecosystem on Twitter, in: Proceedings of the 21st International Conference on World Wide Web, WWW '12, ACM, New York, NY, USA, 2012, pp. 71–80.

[17] H. Yu, M. Kaminsky, P. B. Gibbons, A. D. Flaxman, SybilGuard: Defending Against Sybil Attacks via Social Networks, IEEE/ACM Transactions on Networking 16 (3) (2008) 576–589.

[18] G. Danezis, P. Mittal, SybilInfer: Detecting Sybil Nodes using Social Networks, in: Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, California, USA, 2009.

[19] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, B. Y. Zhao, Detecting and characterizing social spam campaigns, in: Proceedings of the 10th ACM SIGCOMM Internet Measurement Conference (IMC), Melbourne, Australia, 2010, pp. 35–47.

[20] K. Thomas, C. Grier, J. Ma, V. Paxson, D. Song, Design and Evaluation of a Real-Time URL Spam Filtering Service, in: 32nd IEEE Symposium on Security and Privacy (S&P), Berkeley, California, USA, 2011, pp. 447–462.

[21] S. Lee, J. Kim, WarningBird: Detecting Suspicious URLs in Twitter Stream, in: 19th Annual Network and Distributed System Security Symposium (NDSS), San Diego, California, USA, 2012, pp. 183–195.

[22] F. Benevenuto, G. Magno, T. Rodrigues, V. Almeida, Detecting Spammers on Twitter, in: In Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS, Vol. 6, 2010.

[23] P. N. Howard, B. Kollanyi, Bots, #StrongerIn, and #Brexit: Computational Propaganda during the UK-EU Referendum, Social Science Research Network (SSRN).

[24] C. Grier, K. Thomas, V. Paxson, C. M. Zhang, @spam: The Underground on 140 Characters or Less, in: Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS), Chicago, Illinois, USA, 2010, pp. 27–37.

[25] T. Blog, Giving you more characters to express yourself, Online: `blog.twitter.com/official/en_us/topics/product/2017/Giving-you-more-characters-to-express-yourself.html`, Accessed: 18-02-2018.

[26] S. Sedhai, A. Sun, Semi-supervised spam detection in twitter stream, IEEE Transactions on Computational Social Systems 5 (1) (2018) 169–175.

[27] C. Chen, S. Wen, J. Zhang, Y. Xiang, J. Oliver, A. Alelaiwi, M. M. Hassan, Investigating the deceptive information in twitter spam, Future Generation Computer Systems 72 (2017) 319–326.

[28] C. Chen, Y. Wang, J. Zhang, Y. Xiang, W. Zhou, G. Min, Statistical features-based real-time detection of drifted twitter spam, IEEE Transactions on Information Forensics and Security 12 (4) (2017) 914–925.

[29] C. Li, S. Liu, A comparative study of the class imbalance problem in twitter spam detection, Concurrency and Computation: Practice and Experience 30 (5) (2018) e4281.

[30] T. Wu, S. Liu, J. Zhang, Y. Xiang, Twitter spam detection based on deep learning, in: Proceedings of the Australasian Computer Science Week Multiconference, ACM, 2017, p. 3.

[31] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, CoRR.

URL http://dblp.uni-trier.de/db/journals/corr/corr1301.html#abs-1301-3781

[32] N. Chavoshi, H. Hamooni, A. Mueen, Temporal patterns in bot activities, in: Proceedings of the 26th International Conference on World Wide Web Companion, International World Wide Web Conferences Steering Committee, 2017, pp. 1601–1606.

[33] B. Wang, A. Zubiaga, M. Liakata, R. Procter, Making the Most of Tweet-Inherent Features for Social Spam Detection on Twitter, in: Proceedings of the the 5th Workshop on Making Sense of Microposts, co-located with the 24th International World Wide Web Conference (WWW), Florence, Italy, 2015, pp. 10–16.

[34] B. Lott, Survey of Keyword Extraction Techniques, Tech. rep., UNM Education (2012).

[35] Twitter, Twitter Streaming APIs, Online: dev.twitter.com/streaming/overview, Accessed: 18-02-2018.

[36] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.

[37] N. P. Halko, P.-G. Martinsson, J. A. Tropp, Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions, Society for Industrial and Applied Mathematics (SIAM) Review 53 (2) (2011) 217–288.

[38] P. Wiemer-Hastings, K. Wiemer-Hastings, A. C. Graesser, Latent Semantic Analysis, in: Proceedings of the 16th international joint conference on Artificial intelligence, 2004, pp. 1–14.

[39] F. J. Tweedie, R. H. Baayen, How variable may a constant be? measures of lexical richness in perspective, Computers and the Humanities 32 (5) (1998) 323–352.

[40] D. Biber, S. C. and Geoffrey Leech, The Longman Student Grammar of Spoken and Written English, Longman, 2002.

[41] Z. Šišková, Lexical richness in efl students' narratives, Language Studies Working Papers 4 (2012) 26–36.

[42] Twitter, The Twitter Rules, Online: help.twitter.com/en/rules-and-policies/twitter-rules, Accessed: 18-02-2018.

[43] V. Qazvinian, E. Rosengren, D. R. Radev, Q. Mei, Rumor has it: Identifying Misinformation in Microblogs, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, (EMNLP), A meeting of SIGDAT, a Special Interest Group of the ACL, Edinburgh, UK, 2011, pp. 1589–1599.

[44] G. Forman, M. Scholz, Apples-to-apples in Cross-validation Studies: Pitfalls in Classifier Performance Measurement, ACM SIGKDD Explorations Newsletter 12 (1) (2010) 49–57.

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

[46] R. S. Olson, W. L. Cava, Z. Mustahsan, A. Varik, J. H. Moore, Data-driven Advice for Applying Machine Learning to Bioinformatics Problems, Pacific Symposium on Biocomputing (PSB) 2018 Online Proceedings 23 (2018) 192–203.

[47] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.

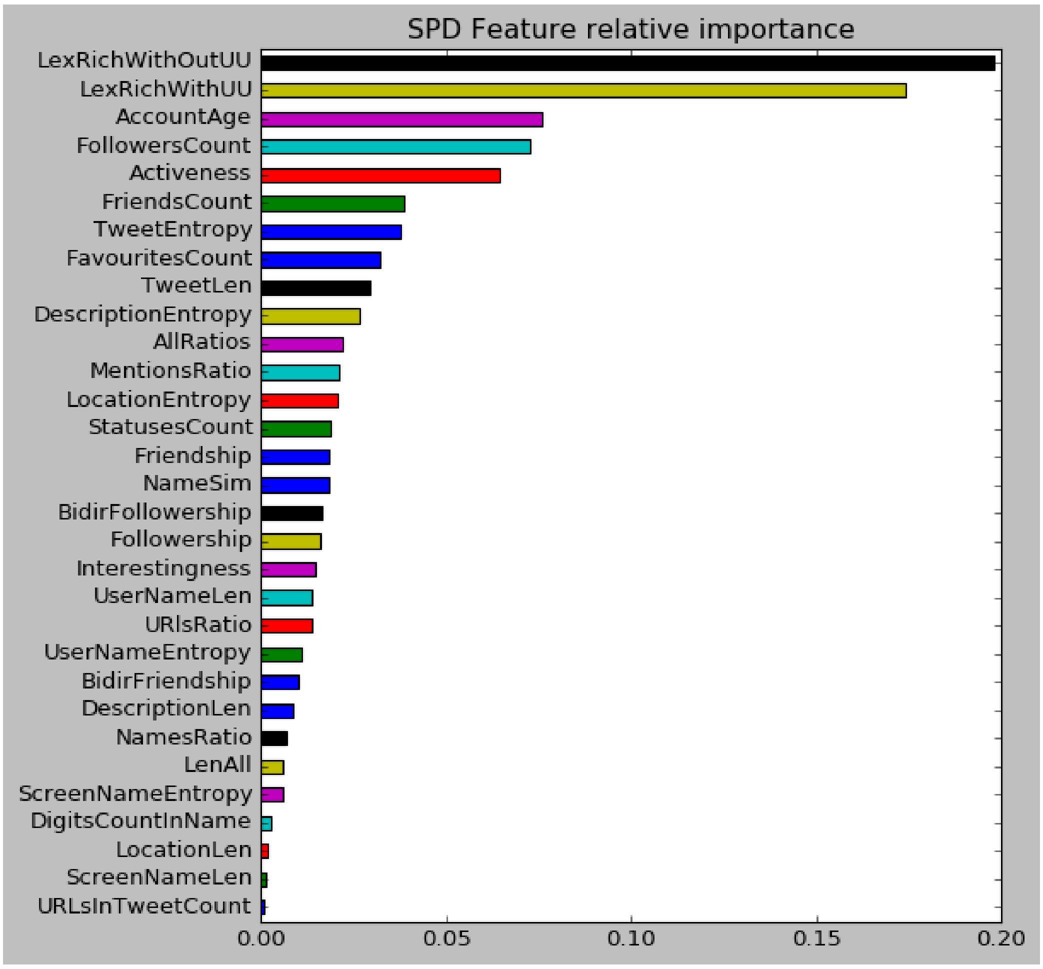
[48] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters 27 (8) (2006) 861–874, rOC Analysis in Pattern Recognition.

[49] N. Japkowicz, The class imbalance problem: Significance and strategies, in: In Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI), 2000, pp. 111–117.

[50] I. Inuwa-Dutse, Modelling formation of online temporal communities, in: Companion of the The Web Conference 2018 on The Web Conference 2018, International World Wide Web Conferences Steering Committee, 2018, pp. 867–871.

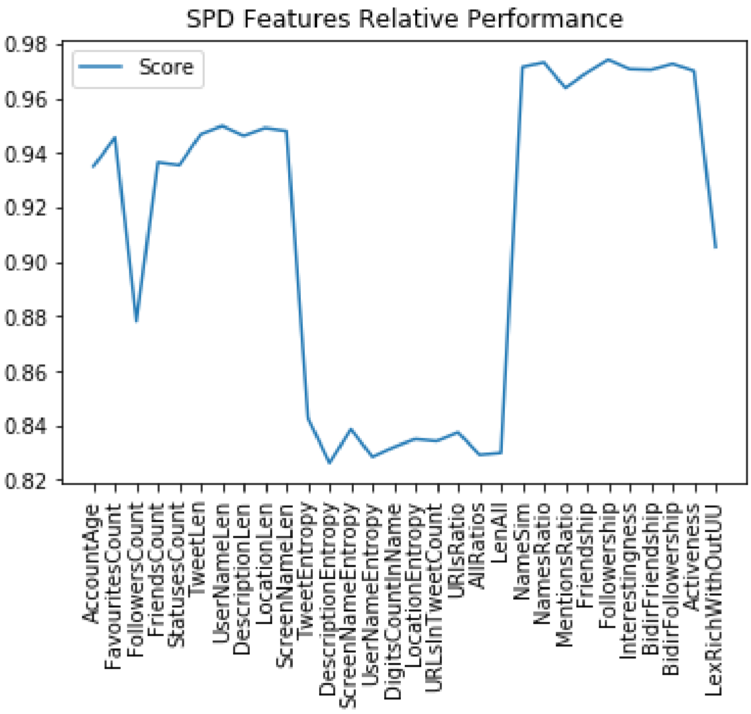
[51] F. Godin, B. Vandersmissen, W. De Neve, R. Van de Walle, Multimedia lab @ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations, in: Proceedings of the Workshop on Noisy User-generated Text, 2015, pp. 146–153.

Appendix



Supplementary Figure A.1: Features and their corresponding relative importance. Importance scores sum to one and *LexRichWithOutUU* scores the highest.

We train the Word2Vec model on various datasets, as shown in table A.3, to learn the semantics and relationships of the spammy words, shown in table A.2, as used by various users. The idea of training on multiple datasets is to understand how the terms are semantically related. We utilised the learned features in a new experiment. Results are shown in table 10.



Supplementary Figure A.2: The complete set of proposed features and their relative importance

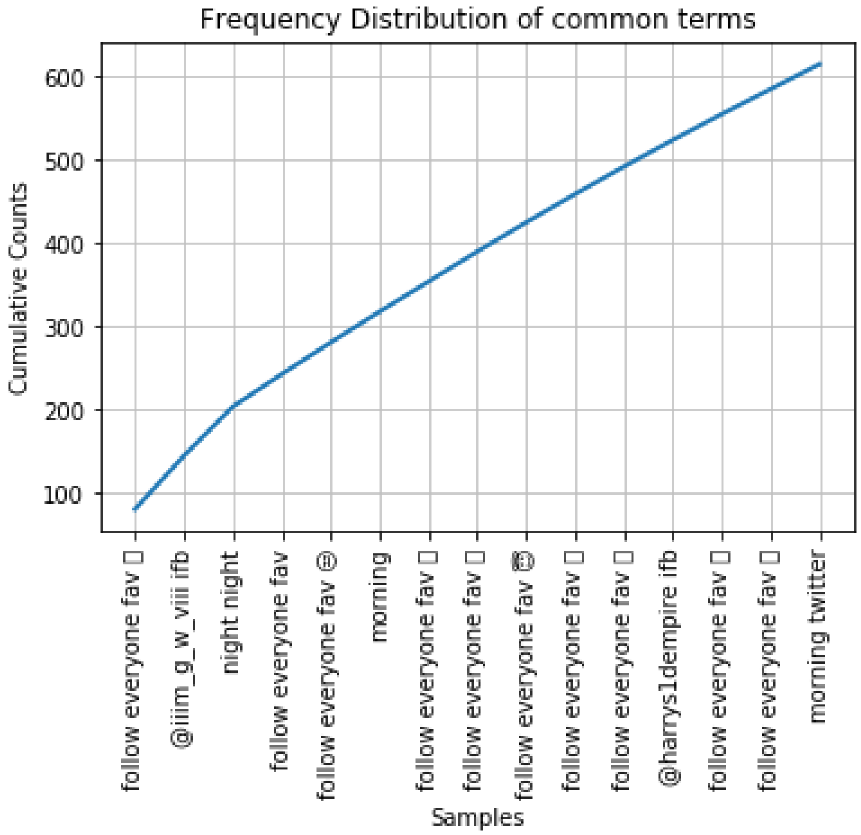
Common spammy words
follow, everyone, fav, ifb, retweet, this, like, you, gain, followers, risingplanet, risegain, trickfollowhp, proximalfollow, thegainfactor, simplegain, dogfather__mgwv, gainhub, teamstallion, trapadrive, gainwithxtiandela, 1m__000 ifb, gaintweet24, comment, want, new, gainaccount98, instant, back, followtrain

Common spammy n-grams		
bigrams	**trigrams**	**four-grams**
follow train	T1: **free new followers**	F1: **follow everyone *who* fav this**
B1: **gain follower**	instant follow back	follow back follow you
free followers	T2: **gain new followers**	F2: **follow everyone *who* likes this**
B2: **follow everyone**	please fav this	retweet for more followers

Supplementary Table A.1: Examples of spammy words and common n-grams in the $SPD_{automated}$ dataset. The distribution of N-grams in bold face in the datasets is reported in Table 3. Italicised words are included in the stop-list and are not counted towards N-gram length.

Study	**Spammy terms**
Sedhai and Sun [26]	Followme, follow, follow back, back, ipad ipadgames, please follow, please, followers, retweet, tfbjp, teamfollowback, ipad, follow me, collected, followback, gameinsight
This study	follow, everyone, fav, ifb, retweet, this, like, you, gain, followers, risingplanet risegain, trickfollowhp, proximalfollow, thegainfactor, simplegain, dogfather__mgwv, gainhub, teamstallion,trapadrive, gainwithxtiandela, 1m__000 ifb, gaintweet24, comment, want, new , this

Supplementary Table A.2: Lists of spammy words.



Supplementary Figure A.3: An example of cummulative frequency distribution of some common *spammy n-grams* in $SPD_{automated}$ dataset

Dataset	**Description**
$\text{Honeypot}_{\text{spam}}$	Spam tweets made publicly available by Honeypot [9]
$\text{Honeypot}_{\text{non-spam}}$	Legitimate users collected by Honeypot [9]
SPD_{spam}	Spam datasets collected for this study
$\text{SPD}_{\text{non-spam}}$	Non-spam dataset from verified genuine users
Pretrained W2V	A pre-trained Word2Vec model1 on a corpus of 15 million tweets [51]

Supplementary Table A.3: Summary of datasets utilised for Word2Vec model training model for feature extraction.