A Framework for Real-Time Spam Detection in Twitter

Himank Gupta, Mohd. Saalim Jamal, Sreekanth Madisetty and Maunendra Sankar Desarkar
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad, India
Email: [cs16mtech01001, cs16mtech11024, cs15resch11006, maunendra]@iith.ac.in

***Abstract*—With the increased popularity of online social networks, spammers find these platforms easily accessible to trap users in malicious activities by posting spam messages. In this work, we have taken Twitter platform and performed spam tweets detection. To stop spammers, Google SafeBrowsing and Twitter's BotMaker tools detect and block spam tweets. These tools can block malicious links, however they cannot protect the user in real-time as early as possible. Thus, industries and researchers have applied different approaches to make spam free social network platform. Some of them are only based on user-based features while others are based on tweet based features only. However, there is no comprehensive solution that can consolidate tweet's text information along with the user based features. To solve this issue, we propose a framework which takes the user and tweet based features along with the tweet text feature to classify the tweets. The benefit of using tweet text feature is that we can identify the spam tweets even if the spammer creates a new account which was not possible only with the user and tweet based features. We have evaluated our solution with four different machine learning algorithms namely - *Support Vector Machine*, *Neural Network*, *Random Forest* and *Gradient Boosting*. With *Neural Network*, we are able to achieve an accuracy of 91.65% and surpassed the existing solution [1] by approximately 18%.**

I. Introduction

In the past few years, online social networks like Facebook and Twitter have become increasingly prevailing platforms which are integral part of peoples daily life. People spend lot of time in microblogging websites to post their messages, share their ideas and make friends around the world. Due to this growing trend, these platforms attract a large number of users as well as spammers to broadcast their messages to the world. Twitter is rated as the most popular social network among teenagers [2].

However, exponential growth of Twitter also invites more unsolicited activities on this platform. Nowadays, 200 million users generate 400 million new tweets per day [3]. This rapid expansion of Twitter platform influences more number of spammers to generate spam tweets which contain malicious links that direct a user to external sites containing malware downloads, phishing, drug sales, or scams [4]. These types of attacks not only interfere with the user experience but also damage the whole internet which may also possibly cause temporary shutdown of internet services all over the world [5].

As a consequence, researchers as well as Twitter came up with various spam detection solutions to make spam-free online social network platform. Twitter built BotMaker [6] to fight spam on Twitter platform. They have seen a 40% reduction in critical spam metrics since launching BotMaker. But one of the weak aspects of BotMaker is that it fails to protect a victim from new spam, i.e. it is not an efficient tool for real-time spam tweets detection. K. Thomas [7] had observed that 90% users might visit a new spam link before it gets blocked by the blacklist.

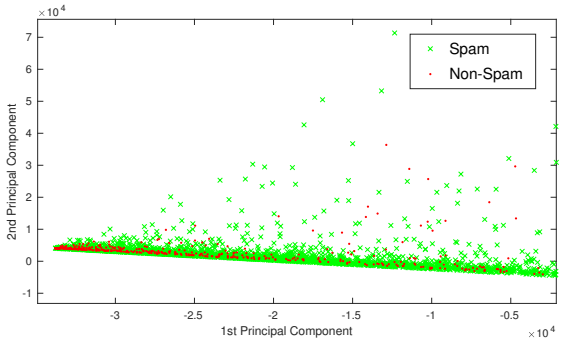

Fig. 1: Scatter plot of dataset showing distribution of two classes namely, spam(x) and non-spam(.)

Tingmin Wu [8] performed spam tweet detection based on deep learning. They used word vector to train their model, but they have not explored user or tweet based features to address the problem. On the other side, Chao Chen [1] used lightweight features (user's and tweet's specific feature) that are suitable for real-time spam tweet detection. As Twitter has increased their character limit to 280 characters, it is essential to scrutinize the tweet's text along with the user-specific features. Despite many existing solutions, there are very few comprehensive solutions that can be used for blocking spam tweets in real-time.

In this paper, we give a framework based on different machine learning approach that deals with various problems including accuracy shortage, time lag(BotMaker) and high processing time to handle thousands of tweets in 1 sec. Firstly, we have collected 400,000 tweets from HSpam14 [9] dataset. Then we further characterize the 150,000 spam tweets and 250,000 non-spam tweets. We also derived some lightweight features along with the Top-30 words that are providing highest information gain from Bag-of-Words model. This approach has been detailed in section III. This technique is proficient

for spam detection in real-time. We also performed various experiments for detecting Twitter spam using our processed dataset.

II. Motivation

Spam in Twitter is different from spam in other online social networks primarily because Twitter exposes developer APIs to make it easy to interact with the platform. Due to this constraint spammers know almost everything about Twitter's anti-spam system through the APIs. So we need a robust system that can mitigate the challenges in Twitter spam detection.

Next challenge in real-time Twitter spam detection is to choose lightweight features that should be feasible to process a large number of tweets in very less time and detect the spam tweets as early as possible. Because the longer a spam tweet remains in the system, the easier it is for users to be affected by it. Chao Chen [10] proposed the novel Lfun algorithm using twelve features to deal with a problem of Spam-Drift in Twitter. In Fig. 1 we present the graphical representation of dataset given by Chao Chen [1].As observed in Fig. 1 that distribution of the two classes namely, spam and non-spam have significantly overlapped that makes difficult to classify the dataset into two classes. Moreover, after Twitter has incremented character limit to 280, we must consider tweet's text as one of the features.

To address these challenges, we incorporate information gain from Bag-of-Words model along with user-based features in Twitter platform. In summary, our contributions are listed below:

- We collect real-world tweets from tweet ids given in HSpam14 dataset. We then extract user based features from 150,000 spam tweets and 250,000 tweets.
- From above 400,000 tweets' text, we collect around 100,000 unique words, out of which we identify 30 words that are possibly strong indicators for marking a tweet as spam or non-spam.
- On this processed dataset, we train our model using on various machine learning algorithms.

III. Proposed Work

We prepare our dataset by collecting tweets corresponding to 400,000 tweet ids from HSpam14 [9]. We then created the features set mentioned in Table I on our dataset. In order to get information from tweets' text, we want to extract those words that can be strong indicators to classify the tweets in one of the classes: spam or non-spam.

A. Information Gain from Bag-of-Word Model

After characterizing the spam and non-spam tweets' text into two separate documents, we construct the following sets:

$\mathcal{U}_{\mathcal{S}}$ = Collection of unique words in the spam tweets' text.
$\mathcal{U}_{\mathcal{NS}}$ = Collection of unique words in the non-spam tweets' text.

For each word $\mathcal{T}$ in $\mathcal{U}_{\mathcal{S}}$ and $\mathcal{U}_{\mathcal{NS}}$ we calculate the following probablity values:

$$\mathsf{P}(\mathcal{T}|\mathcal{U}_{\mathcal{S}}) = \frac{\text{\# of Spam tweets that contain } \mathcal{T}}{\text{total \# of Spam tweets}} \quad (1)$$

$$\mathsf{P}(\mathcal{T}|\mathcal{U}_{\mathcal{NS}}) = \frac{\text{\# of Non-Spam tweets that contain } \mathcal{T}}{\text{total \# of Non-Spam tweets}} \quad (2)$$

We calculate the information gain $\gamma_{\mathcal{T}}$ for each word $\mathcal{T}$ as follows:

$$\gamma_{\mathcal{T}} = \left| \frac{\mathsf{P}(\mathcal{T}|\mathcal{U}_{\mathcal{S}})}{\mathsf{P}(\mathcal{T}|\mathcal{U}_{\mathcal{NS}})} \times \log_{10}[\frac{\mathsf{P}(\mathcal{T}|\mathcal{U}_{\mathcal{S}})}{\mathsf{P}(\mathcal{T}|\mathcal{U}_{\mathcal{NS}})}] \right| \quad (3)$$

We sort words in decreasing order based on their $\gamma_{\mathcal{T}}$ score calculated in Equ. 3. We take the top 15 words from each of the $\mathcal{U}_{\mathcal{S}}$ and $\mathcal{U}_{\mathcal{NS}}$ using above calculation. Sample top-10 words are illustrated in Table II. We combine these words to form top-30 words that we use in our feature set. The benefit of using these words based on their entropy score in the feature-set is that we were able to reduce uncertainty in the prediction outcome as these words have a different impact of frequency count in spam and non-spam tweets. Hence we expect considering these top 30 words will help us to classify the tweets accurately for each class.

B. Extracting Lightweight Features

After collecting 400,000 labelled tweets, we extracted around 350,000 English tweets. Since we are receiving an arbitrary independent tweet from Twitter API, so we could not obtain the complete social graph of Twitter's users. Consequently, we take the feature set from Chao Chen's work [1] that is more suitable for timely detection of Twitter spam. However, we add one more feature, i.e., no_of_non_ASCII on top of those 12 features. From our analysis, we found that 88% of spam tweets use non-ASCII values to post a tweet in

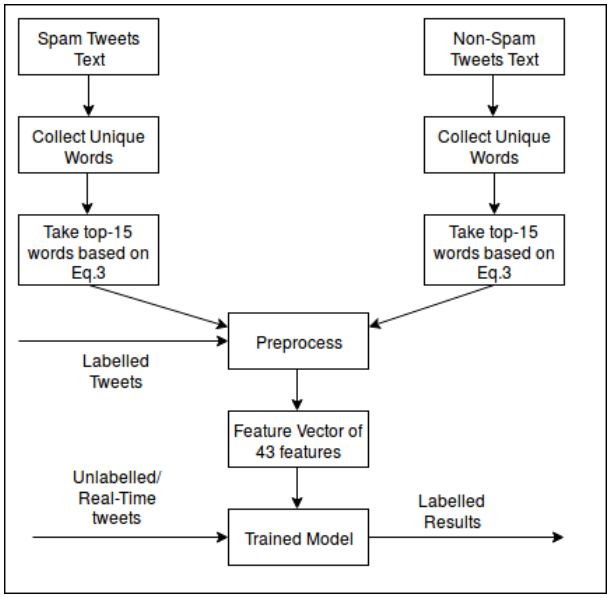

Fig. 2: Flow Diagram to preprocess the dataset for Information gain

TABLE I: Feature Set

Feature Name	Description
account age	The age (days) of an accounl since its creation until the time of sending the most recent tweet
no_follower	The number of followers of this Twitter user
no_following	The number of followings/friends of this Twitter user
no_userfavourites	The number of favourites this Twitter user received
no_lists	The number of lists this Twitter user added
no_tweets	The number of tweets this Twitter user sent
no_retweets	The number of retweets this tweet
no_hashtag	The number of hashtags included in this tweet
no_usermention	The number of user mentions included in this tweet
no_urls	The number of URLs included in this tweet
no_char	The number of characters in this tweet
no_digits	The number of digits in this tweet
no_non-ASCII_characters	The number of non-ASCII characters in this tweet

the form of text. Table I shows the extracted 13 features from dataset.

We sampled our Feature-set-2 with Bag-of-Words model. 100 thousand unique words have been identified from the tweets' text. Using these many words along with extracted 13 features set in Table I, we built our feature set in libsvm format. It is impractical to use Feature-set-2 for other classifiers due to input feature-set size of 100 thousand features. We combine these user and tweet based 13 features along with our top-30 words as extracted in Section III-A based on tweet's text. We then calculate the 43 dimeansional feature vector for every tweet and use them to train our machine learning model. For each top-30 word, the value in feature set corresponds to a frequency of that word in a particular tweet. Fig. 2 represnt the sequence of various steps used in our work.

C. Scaling of Dataset

We classify our feature sets in 3 categories as given in Table IV. We examine that features' values are not in same range that will affect our model training in section IV. So for Feature-set-1 we scale our data as follows:

$\mathcal{D}^1$ = Matix representation of Dataset-1 of size of $\mathcal{M} * \mathcal{N}$, where $\mathcal{M}$ = numbers of tweets, $\mathcal{N}$ = number of features. In our case m = 350,000 and n = 43.

$\mathcal{D}_i^j$ = j^{th} feature of i^{th} tweet.

TABLE II: Sample Top-10 Words

Top 5 Words from Spam Tweets	Top 5 Words from Non-Spam Tweets
harvested	rain
tribez	asleep
coins	rather
collected	college
unfollower	fell
openfollow	follback
inspi	dinos
build	bullshit
smurf	child
brainy	couch

For reprsenting the data using Feature-set-1, we normalize the data so that each of the feature has zero mean and unit standard deviation.

$$\tilde{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \tag{4}$$

Where x_{ij} is the j^{th} feature value in the i^{th} tweet
$\tilde{x}_{ij}$ is the noramlized feature value for j^{th} feature value in the i^{th} tweet
μ_i is the mean value for j^{th} feature over all tweets
σ_j is the standard deviation of value for j^{th} feature in the dataset

While using the feature-set-2, we reprsent each feature using its Bag-of-Word reprsentation. We use libsvm format to store this representation. Here each feature is a word and corresponding value is the frequency of the word in the tweet. We normalize each of the tweet-vectors using l^2-norm.

IV. Experimental Setup and results

In this section, we will measure the Twitter spam detection performance on our dataset by using four machine learning algorithms, *Support Vector Machine with kernel*, *Neural Network*, *Gradient Boosting* and *Random Forest*.We also compare our results with Chao Chen's spam detection technique on their dataset [1].We even patterned three different feature sets for our experiment. The dataset are listed in Table IV. To evaluate the performance of our created classification and make it comparable to current approaches, we use Recall, Precision, F-measure and Accuracy to measure the effectiveness of classifiers. We consider the spam class as a positive class and non-spam class as a negative class. We determine the Recall, Precision, F-measure and Accuracy as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{5}$$

TABLE III: Performance Evaluation on Feature-set-1, Feature-set-2 and Feature-set-3

Unit %	**Feature-set- 1**		**Feature-set- 2**		**Feature-set- 3**	
Classifier	F-Measure	Accuracy	F-Measure	Accuracy	F-Measure	Accuracy
SVM with Kernel	86.18	85.95	84.28	83.88	79.9	79.1
Neural Network	90.56	91.65	-	-	71.25	72.15
Gradient Boosting	75.81	85.84	-	-	81.26	82.69
Random Forest	75.39	86.25	-	-	93.6	92.9

Recall (Sensitivity) is defined as the ratio of correctly classified spam in total actual spam, as

$$\textit{Recall} = \frac{TP}{TP + FN} \tag{6}$$

Precision is defined as true projected spam to classified spam. It can be obtained by

$$\textit{Precision} = \frac{TP}{TP + FP} \tag{7}$$

F-measure is the harmonic mean of Precision and Recall, and it can be calculated as follow:

$$\textit{F-measure} = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{8}$$

Table III shows the comparison of different feature sets for various classifiers. From Table III we can infer that with Feature-set-1 *Neural Network* gives the finest accuracy i.e. 91.65% among all classifier. Also, our approach of using top-30 words for features set outperformed Chen Chon's [10] approach by 18%. However, for Feature-set-2 we cannot use different classifier other than *Support Vector Machine* because for other classifiers it is impractical to give input vector having dimensions of 100 thousand features. So we evaluate Feature-set-2 for *Support Vector Machine* only.

Table III shows that *Random Forest* for Feature-set-3 is 2% better than a neural network for Dataset-1, but Feature-set-3 is more based on user-based (eg, account age, # of followers) feature so it cannot detect Twitter spam if a spammer creates new user account. But we incorporate user based features with Top-30 words then based on tweet's text we can predict it as spam. Thus, it is significant to detect Twitter spam as soon as possible to mitigate the loss caused by spam. Because of that property our approach gives compelling contribution to detect Twitter spam in real-time.

V. Conclusion & Future Work

In this paper, we present a novel framework for real-time spam detection in Twitter. We collected a large number of 400,000 public tweets. Based on tweet's text we extract top-30 words which are able to give the highest information gain in order to classify the tweets. We have also tested our approach with real-time tweet detection that has outperformed existing approach [1] by 18%. As Twitter API is available to all users, spammers may change their behavior over the time. In the real world, spam tweet's feature keeps on changing in an unanticipated way. This problem is referred as "Spam Drift."

TABLE IV: Sampled Dataset

Feature-Set	Sampling Method	Ratio (Spam:Non-Spam)
1	Use 43 features to train a model	1:2
2	Use Bag-of-Word to select features in libsvm format	1:2
3	Use Chao Chen's [1] dataset for comparison	1:2

In the future, we will keep on updating our Bag-of-Words model based on new spam tweets by implementing self-learning algorithm. Also, we observe in our dataset that 79% of spam tweets contain a malicious link. So we will also perform the URL crawl mechanism to detect Twitter spam. *Frequent Pattern Mining* of tweets' text can also be the vital aspect to distinguish Twitter spam in real-time. We will consolidate these three approaches to handle Spam Drift problem.

References

[1] C. Chen, J. Zhang, X. Chen, Y. Xiang, and W. Zhou, "6 million spam tweets: A large ground truth for timely twitter spam detection," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 7065–7070.

[2] A. Greig, "Twitter Overtakes Facebook as the Most Popular Social Network for Teens,According to Study, DailyMail, accessed on Aug. 1, 2015 ," http://www.dailymail.co.uk/news/article-2475591/Twitter-overtakes-Facebook-popular-socialnetwork-teens-according-study.html, 2015, [Online].

[3] H. Tsukayama, "Twitter turns 7: Users send over 400 million tweets per day," https://tinyurl.com/ybsaq7e7, 2013, [Online].

[4] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on twitter," in *In Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS*, 2010.

[5] C. Pash., "The lure of Naked Hollywood Star Photos Sent the Internet into Meltdown in New Zealand, Bus. Insider, accessed on Aug. 1, 2015 ," https://tinyurl.com/yc93ssj4, 2014, [Online].

[6] "BotMaker," https://blog.twitter.com/engineering/en_us/a/2014/fighting-spam-with-botmaker.html, [Online].

[7] K. Thomas, C. Grier, D. Song, and V. Paxson, "Suspended accounts in retrospect: An analysis of twitter spam," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC '11. New York, NY, USA: ACM, 2011, pp. 243–258. [Online]. Available: http://doi.acm.org/10.1145/2068816.2068840

[8] T. Wu, S. Liu, J. Zhang, and Y. Xiang, "Twitter spam detection based on deep learning," in *Proceedings of the Australasian Computer Science Week Multiconference*, ser. ACSW '17. New York, NY, USA: ACM, 2017, pp. 3:1–3:8. [Online]. Available: http://doi.acm.org/10.1145/3014812.3014815

[9] "HSpam14 Dataset," http://www.ntu.edu.sg/home/axsun/datasets.html, [Online].

[10] C. Chen, Y. Wang, J. Zhang, Y. Xiang, W. Zhou, and G. Min, "Statistical features-based real-time detection of drifted twitter spam," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 914–925, April 2017.