Spam Detection Of User And Streaming Tweets On Twitter

Anamika Sharma
Data Science
IIIT-B
Bangalore, India
anamika.sharma@iiitb.org

Antriksh Shah
Data Science
IIIT-B
Bangalore, India
shah.antriksh@iiitb.org

***Abstract*—The virtual world of social networks has been growing exponentially over the last few years. A similar trend of increasing spam is also seen in these social networks[1]. Our approach to detect a spam user on Twitter is by analyzing the content attributes of the users tweet. We show with 99% accuracy that we can classify a user as spammer or not, by looking at tweets in the users timeline and identifying features such as number of spam words and censored words used, number of tweets having sensitive content shortened URLs and the ratio of number of retweets per total tweet count in the time line of a Twitter user. We have also performed spam detection on live streaming tweets provided by the streaming API of Twitter with 87.6% accuracy with the help of both content and user behavior attributes. For both the above cases, we have collected a training dataset having sufficiently large number of spam, non spam users and spam, non spam tweets. We manually labeled them and extracted values of the above mentioned features. On the trained dataset we analyzed the F-measure of various classifiers. From the various classifiers, we conclude Random Forest to be the best classifier for detection of spam users and spam tweets.**

***Keywords*-Spam Detection; Streaming Tweets; Spam User; Twitter; Social Media;**

I. Introduction

Twitter is a social networking website where registered users can post upto 140 character long message called 'tweet'. A tweet is considered a spam if it is highly irrelevant to the user who sees it. There is a chance that a tweet considered spam by one user may not be spam for another user. Hence to standardize the definition of spam we consider a user as spam if: the user posts- duplicate tweets with the same content for the purpose of promotion, tweets with malicious, censored or spam content. The user is also said to be a spam user if the user aggressively follows or unfollows accounts. Our definition of spam is in sync with the definition followed by Twitter [2].

Previous researchers have proposed various features based on the user behavior attributes of a Twitter user [3]. In this approach, spam user is detected based on features such as average age of the account, fraction of tweets containing URLs, average number of hashtag per tweet etc. Clearly these features are in control of the spammer. User behavior based features could be easily tricked by any modern day spammer. Another proposed approach was using relational features between sender and receiver [4].In this approach, the Twitter world is considered to be a directed graph. The features used are number of followers, number of friends and reputation of a user. This approach firstly is dependent on details of both the spam sender and some audience receiver. Moreover in this approach a new user or a user having network similar to that of a spammer may be falsely classified as spammer.These reasons motivated us to understand the mindset of a spammer in depth and design a different mechanism to detect spam.

A common strategy of a spammer interested in promoting a brand is to increase the count of his following (number of users who follow the particular spam user). In 2008 in an experiment it was found that on Facebook 41% of users accepted friend request from a random person [5] and 45% of users click on links posted by any friend [6]. The statistics show there is a good chance, a well constructed spam account would have high spam penetration among normal users. Moreover a spam user is most likely to be a part of a large spam user network that is, a spammer is usually followed by multiple spammers belonging to a common spam network. On social network just like real world, a user having large number of followers or larger number of contacts would appear more credible. Hence a spammer having higher follower count would appear more reliable and trustworthy.

Increase in spam can diminish the user experience on Twitter and can damage the reputation of the network [2]. We are addressing spam detection on Twitter at two levels:

- detecting a spam user given a screen name.
- detecting spam tweets from a live stream of tweets.

We observed a total of 6819 spam, non spam accounts. For each user observed we stored their profile details and maximum of upto 3000 recent tweets. We extracted relevant features based on user profile and tweet content and verified their F-measure with help of various classifiers. We did the same process for detecting spam tweets by collecting a set of 2125 tweets. We then show the F-measures of various classifiers and prove Random Forest as the best classifier.

Rest of the paper is organized as follows. The next section contains our work divided into two sections detecting whether a given user is a spam user or not and detecting if a tweet from live stream of tweets is spam or not. Each of these two topics are divided into following parts. Pre-processing talks about steps taken before the actual analysis. Web Crawler talks about how we obtained data for training. Next we show how we built

a labeled collection. Subsequently we have attribute selection and feature extraction. In the end we compare the F-measure of various classifiers. Section 3 highlights the conclusion of our work.

II. Our Work

We have worked on two problems: Identifying whether a given user is a spam user or not. Second, given a set of filter words for stream of live tweets, classifying each tweet as spam tweet or not spam tweet.

A. Detecting whether a given user is a spam user or not

1) Pre-Processing: Our work involves analyzing whether a given user name is a spam user or not by obtaining two sets of attributes, user behavior attributes of the given user name and content attributes of the tweets posted. The user behavior attributes include features such as profile description, number of tweets posted by the user, number of users as followers, number of users following, number of likes, number of lists the user is present in, the date of joining Twitter and many others. The content attributes are essentially the tweets posted by the user. It contains number of spam words and censored words present in a tweet, the URL in a tweet if it is shortened or not and others. Our work involves analyzing the tweets for shortened URLs, number of retweets, content of the tweet and few other features.

In performing content analysis, we are using NLTK-3.0 Natural Language Toolkit for Python and TTP-Twitter text Parser for python. The first step of pre-processing was identifying all tweet words and hashtag that belong to either stop words, spam words, censored words or other normal words. To identify relevant words, we created three dictionaries one each for stop words, spam words and censored words.

Stop words are the common words which do not reveal any information which can help detect spam. Spam words are the words most frequently used by promotional spammers. Censored words are those which are either profane or derogatory in normal context. To create a stop word list, NLTK offers basic 128 English stop words that are general purpose for any content analysis assuming the spelling of the word for analysis would be correct as per a dictionary. But since Twitter has a limit of 140 characters, it is a common sight to see users using SMS lingo or short forms for words, so the stop word list had to be enriched. We collected a list of all emoticons, relevant SMS-lingo words, and other frequent words and made a corpus of total 2517 words. To create a spam word dictionary we first gathered popular spam promoter profiles. We considered a basic assumption which is true for any social network, the users who follow a spammers are most likely to be spammers. We also observed the same when we looked at the followers list of some popular spam promoter profiles; most of their followers were by spam users. This could be easily understood as on Twitter a user profile with more following is assumed to be more credible. Moreover, ideally no normal user would be following a spam user. From the most popular spam promoters, we collected the user names of their followers. This set of spam promoter user names were sent to the web crawler which would obtain their user behavior attributes and content attributes from the recent up to 3000 tweets. For each tweet posted by every user in the spam user set, we extracted the tweet text using TTP. Next the tweet text is converted from Unicode encoding to ASCII. The punctuation inside the tweet were removed. The refined text was split on words. And each word was stored in a common word dictionary along with the frequency of occurrence. Next we filter out words belonging to the stop words dictionary. Once every word was analyzed, we manually filtered the top 150 words into a spam word dictionary. The top spam words are : follow, followers, retweet, que, teamfollowback, gain, stats, unfollowers, followback, retweets, followed, free, follower, followtrick.

To create a censored word dictionary, we first identified Twitter profiles such as @FakeOuter, which have a collection of users who post censored content. All such users were identified and were sent to the web crawler. We followed the same process as done for spam words, and thus created a censored word dictionary.

Apart from words a tweet typically contains a URL. We created a white list of URL domains which we know would not help us detect a spam URL. To create a white list of URLs, we first identified a few normal users which were known not to be spam. We collected all the followers of that user, and obtained all tweets containing URLs. Twitter does shorten every URL to t.co/, but it also provides an option called the expanded URL which is the URL in the tweet at the time of posting. For each tweet containing a URL, we obtained the expanded URL of the tweet. Next using regular expression, we filtered out the domain of the tweet. If the domain length was less than 8, it typically denoted the URL was generated by a URL shortening website. We created a dictionary of the domains of the URLs, and the frequency of occurring. We manually filtered that list, and collected a list of domains, which would not help in identifying if a user is a spammer or not. In this manner we have obtained dictionaries to identify stop words, spam words, censored words and a dictionary of white listed URLs.

2) Web Crawler: Twitter network is similar to a typical follower following network where in the users following a given user share some common traits. A spammer is more likely to have a following of spammers, and a non spammer is more likely to have a following of non spammers. Hence, given a user name which we can ascertain to be a spammer, exploring the list of followers, we can gradually begin to explore a network of spammers.

We built a web crawler where in given a screen name, the crawler first obtained a list of screen names of its 5000 followers. Then for each user name the crawler obtained the user behavior attributes and the recent up to 3000 tweets of that user. The data is stored in flat files. The directory name corresponds to the given screen name for analysis. The folder contains files corresponding to each of his follower. The

file contains first the user behavior attributes followed by the tweets.

3) Data set Collection: To classify a user as spammer or not, we need labeled set of users indicating either spam user or not spam user. There is no such corpus readily available in the public. The first step of supervised learning was to create a labeled training set where each user is manually classified as spammer or non-spammer.

4) Building a labeled collection: We now look at how we built our training data set with desired properties. The desired properties are:

- The data set should have equal amounts of various kinds of users found on Twitter so as to ensure the data set is not biased. Various kind includes users who aggressively promote a company, user who posts censored or sensitive tweets, users who retweet a lot, dormant users who do not post tweets along with normal user who do not fall under the definition of a spam user.
- The training set should be large enough so as to get the best results from the supervised learning classifiers.

In order to ensure these requirements are met, we browsed around the Twitter network and identified certain profiles such as @Follow2gain__ which had a vast network of promotional spammers, @FakeOuter which has a list of users who tweet censored or malicious content. Exploring the followers network of such users, we found it had sufficient number of various types of users. This approach is different and better than the ones used in most of the previous research work, where user set was obtained from collecting users who tweeted on a trending topic. In the previous approach we could not account for dormant users and those users who are inactive for some time.

To label the data, two volunteers looked at the user profile description, profile photo, profile header photo, random tweets, media links and URLs posted by the user. We ensured the two volunteers independently labeled the tweets and when in doubt were asked to label them as non spammers. We found out there was no tie in the classification, implying the human classification was a 100% success.

In total we collected 6819 users out of which 2859 were labeled spammers and 3960 were labeled non spammer.

5) Selecting User Behavior Attributes: On Twitter, intention of spam users is generally to promote products or enhance the credibility of other users. A common users intention would differ from the spam users hence it is natural to believe there would be a difference in behavior among the two. Intuitively we expect a spammer to have less followers and high following, more tweets containing URLs, periodic tweeting time etc. In order to verify intuitive features, we created two attribute sets namely content and user behavior attributes.

6) Obtaining Features: For finding whether a given user is spam user or not we first find the different features for classification. We classify the features into 2 groups: user behavior attribute features and content attribute features.

1) Content attributes: Content attribute features are not under the control of the user. We obtained the following features:
   - Censored word count: Here we use the dictionary of censored words that we created earlier and count the number of words that belong to the censored word dictionary.
   - Spam word count: Similarly we find the spam word count as the count of number of words that belong to the spam word dictionary.
   - URL shortening ratio: Any Tweet containing a URL is shortened by Twitter automatically. While collecting tweets we get an expanded URL field which provides the actual link at the time of posting. We check if this expanded URL is in turn a shortened URL or not and then calculate the ratio as number of tweets having shortened URLs by total number of tweets having URLs.
   - Retweet ratio: We calculate the ratio as the number of tweets that were retweeted to the total number of tweets.
   - Media URL ratio: Twitter provides the media URL as a part of the tweet if the tweet contains media content like an image or a video. Spammers can promote a product or share sensitive content using media. We calculate the ratio as number of tweets with media URL by total number of tweets.
   - Sensitive tweets count: Twitter provides a field sensitive count which is set true if their detection algorithm finds the content of the tweet possibly sensitive. We calculate the ratio as number of tweets where sensitive count is true by total number of tweets.
2) User behavior attributes: User attributes are under the control of the user. The different user behavior features that we analyzed are: profile description, number of tweets, number of following, number of followers, number of likes, age of the account.
   - Follower following ratio: We observed that for spammers the number of following is much more than the number of followers. We calculate the ratio as number of following by sum of the number of follower and following.

But unlike other previous research work we conclude later, user behavior attributes are not needed to classify correctly a spam user from other. Content attributes can help detect spam users very efficiently. This is in stark contrast with some of the earlier research work which have large number of user behavior attribute features to classify spam users.
Figure:1-4 show weka analysis of some attributes used for user classification.

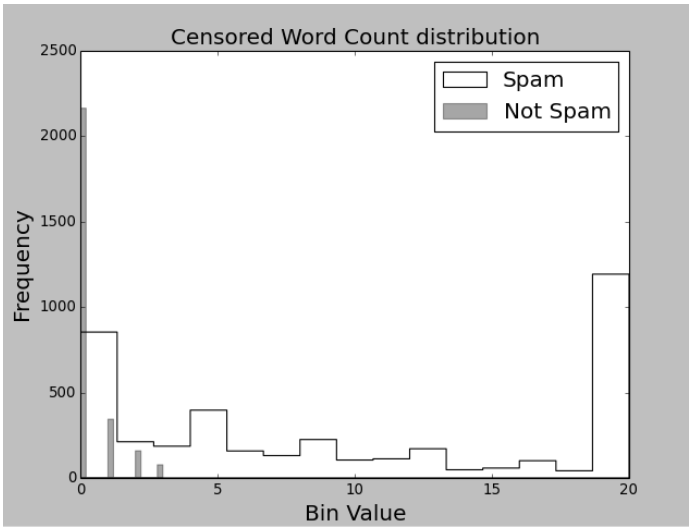

Fig. 1: The histogram shows the amount of censored words used by Spammers and NonSpammers.

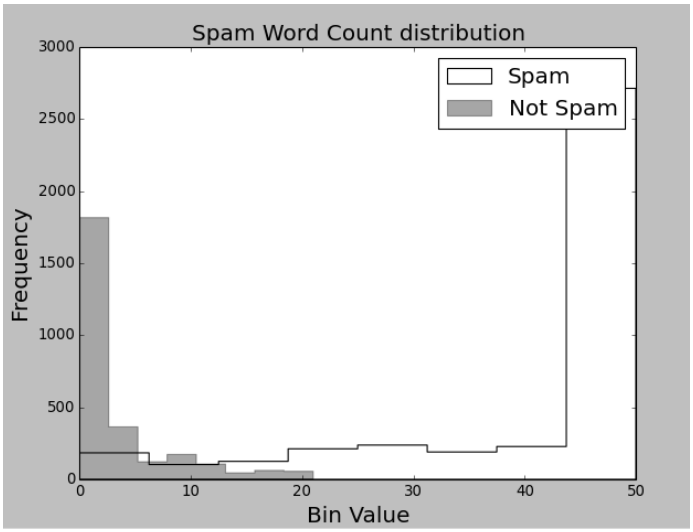

Fig. 2: The histogram shows the amount of Spam words used by Spammers and NonSpammers.

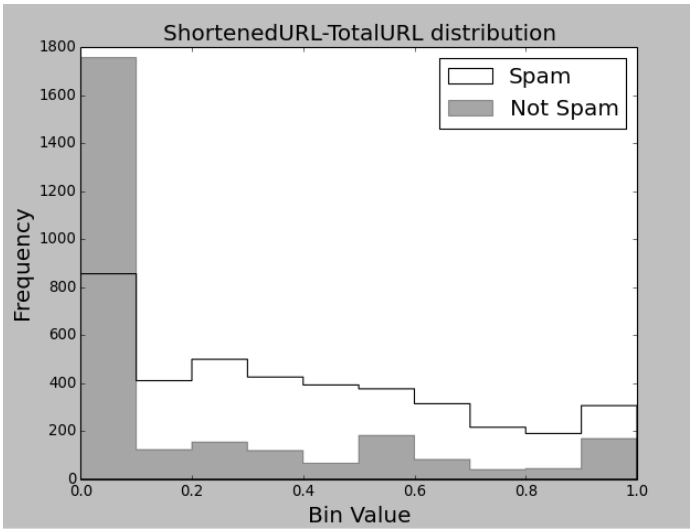

Fig. 3: The histogram shows the ratio of shortened URLs to Total number of URLs posted by a user.

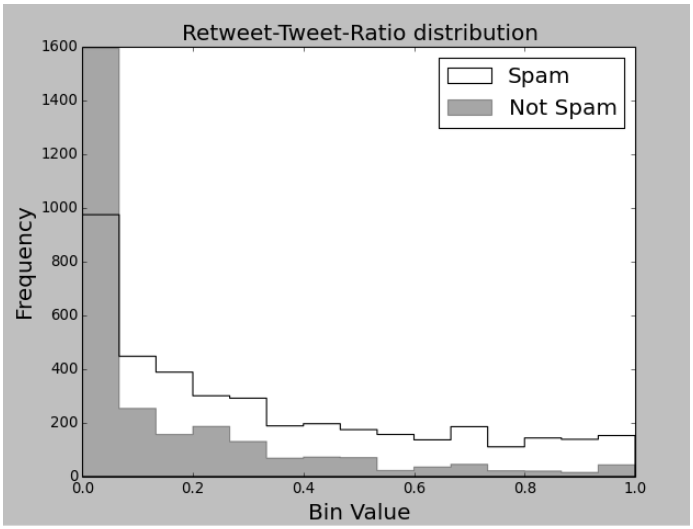

Fig. 4: The histogram shows the ratio of number of retweets to total tweets posted by a user.

7) Classification: Using traditional classifiers and the above features, we see which classifier works best. In our work we have used Random Forest, Support Vector Machine, Naive Bayes, Linear Regression and the J48 Decision Tree. Random Forest gives an accurate estimate about which features are relevant in the classification, and can handle large number of features without rejecting them in the analysis. It also has capabilities to balance errors in classes having unbalanced data. Naive Bayes assumes independence among every feature pair. This assumption is relevant in our work as intuitively the features are independent. Advantage of Naive Bayes is, it simplifies the process and works much faster than other sophisticated algorithms. Support Vector Machine is another commonly used classifier. It is advantageous when the data is not regularly distributed or has an unknown distribution. SVM is more efficient in email categorization with large features than Naives Bayesian [7]. Linear Regression and Decision Tree are also well known and standard classifiers which are implemented in Weka as Regression classifier and J48 Decision Tree respectively.

Evaluations: Using Content based features, our confusion matrix is as shown below

		Prediction	
		Spam	**NotSpam**
Actual	Spam	a	b
	NotSpam	c	d

a represents true Positive count for spam, b represents the false negative count for spam, c represents false positive for

not spam and d represents true negative for not spam

To compare output of various classifiers we used precision, recall and F-measure for comparison.

$$\begin{aligned} Precision = (P) &= a/(a+c) \\ Recall = (R) &= a/(a+b) \\ F - measure = (F) &= 2PR/(P+R) \end{aligned} \tag{1}$$

In our analysis we have considered the recent upto 3000 tweets of a user for analysis. Our results are as shown below (Table I). We see that Random Forest and Decision Tree

TABLE I: Results of classifiers for user classification

Classifier	Precision	Recall	F-measure
SMO	0.806	0.807	0.807
Naive Bayes	0.957	0.953	0.953
Regression	0.993	0.993	0.993
Random Forest	0.999	0.995	0.995
Decision Tree	0.999	0.999	0.999

have the maximum F-measure amongst all other classifiers. Our findings are in sync with earlier research work done in comparing classifiers for detecting spam on Twitter [8].

B. Detecting whether a given tweet is spam or not

To our best of knowledge, this is the first successful attempt to classify tweets in real-time. We present our data with sufficient statistics to back our claim. One of the fundamental difference between tweet classification and user classification is with user classification we have the luxury of time to perform our analysis and also we have past data to look into. In contrast, streaming tweets we neither have the time nor past data for analysis. Hence it is has more complexity than spam detection at a user level.

1) Pre Processing: Our work involves analyzing if a given tweet is spam or not by obtaining two sets of features, user behavior attributes and content attributes. We are using the same spam, censored, stop word dictionary and dictionary of white listed URLs as mentioned in the previous section A.1 .

2) Stream Connection: We used the Twitter streaming API filter functionality where given some keywords we would obtain a live stream of tweets corresponding to those keywords. One of the main reasons why significant work is not accomplished in streaming tweets is because of the unavailability of complete tweet data. With the current stream function, we can obtain a sample of only 1-40% of the live tweets. Moreover Twitter has discontinued its firehose API where in it allowed full 100% access to live streaming tweets at a fixed cost.

3) Data set Collection: Since no corpus is readily available for a list of tweets along with a classified label as spam or not spam tweet, we manually collected tweets and labeled them.

4) Building a labeled collection: We kept our filter set containing two trending topic word, two spam words and two censored words. With trending topic filter words, we ensured the tweets themselves won't be spam. With spam words and censored words we ensured the training data set has representation of all types of users tweeting all types of content.

To label the data, two volunteers looked at only the tweet content and the user profile information and labeled them as either spam or not spam. We ensured the two volunteers independently labeled the tweets and when in doubt were asked to label them as non spammers. We found out there was no tie in the classification, implying the human classification was a 100% success. In total we collected 2125 tweets out of which 693 were labeled as spam and 1432 were labeled as not spam.

5) Obtaining Features: For finding whether a tweet is spam or not we first find the different features for classification. We classify the features into 2 groups: user behavior attribute features and content attribute features.

Content attributes

- spam count: Count of number of words present in the tweet which belong to the spam word dictionary.
- censored count: Count of number of words present in the tweet which belong to the censored word dictionary.
- hashtags count: Count of number of hashtags present in the tweet.
- URL shortening: Number of shortened URLs present in the tweet.

User behavior attributes

- statuses count: Number of tweets tweeted by the user.
- favorites count: Number of people who 'like' the user who posted the tweet.
- listed count: Number of times the user, who posted the tweet, has been listed.
- followers count: Number of people who are following the user who posted the tweet.
- following count: Number of people the user, who posted the tweet, is following.
- follower to following count: Ratio of number of following to the sum of number of following and number of followers.
- age of account: Difference between present time and the time at which the user's account was created.

Figure:5-7 show weka analysis of some attributes used for tweet classification.

6) Classification: With the help of traditional classifiers and the above features, we see which classifier works best. In our

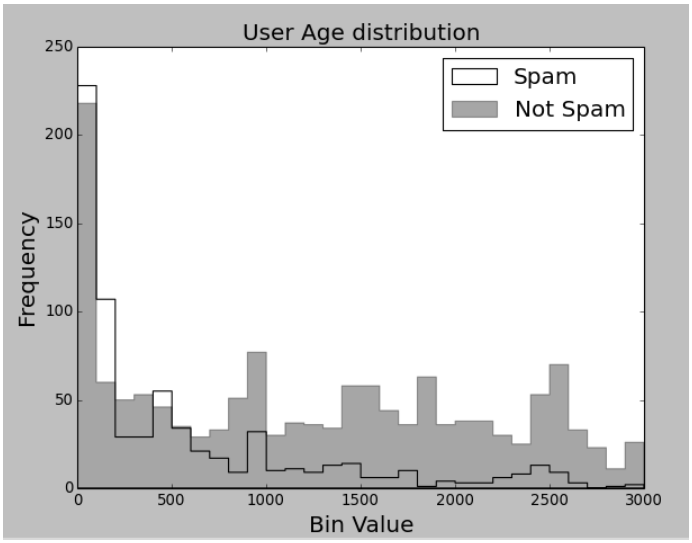

Fig. 5: The histogram shows the age of the account in unit of days.

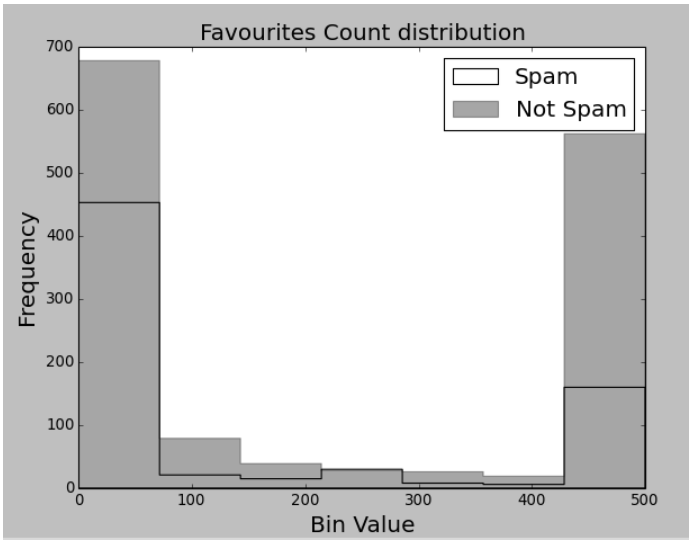

Fig. 6: The histogram shows the number of favorites of a user.

work we have used Random Forest, Support Vector Machine, Naive Bayes, Linear Regression and the J48 Decision Tree.

Evaluations: The output of the classifiers are shown below (Table II).

TABLE II: Results of classifiers for tweet classification

Classifier	Precision	Recall	F-measure
SMO	0.837	0.836	0.836
Naive Bayes	0.817	0.809	0.809
Regression	0.874	0.872	0.872
Random Forest	0.876	0.877	0.877
Decision Tree	0.867	0.868	0.868

We see that Random Forest has the maximum F-measure

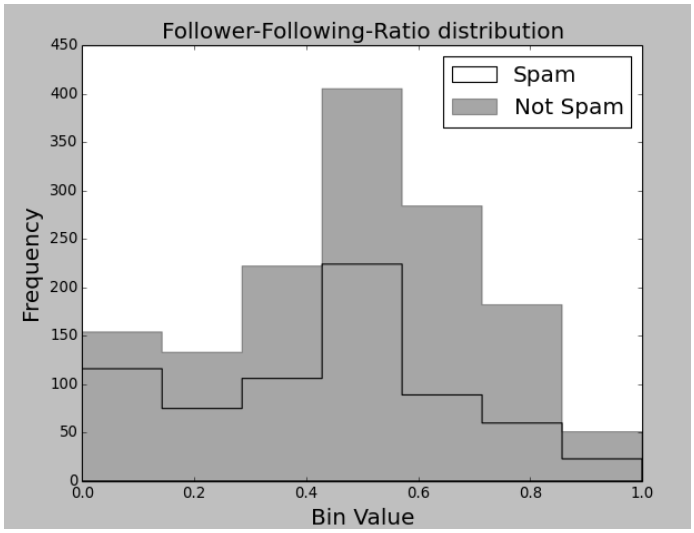

Fig. 7: The histogram shows the ratio of number of followers to total of followers and following of a given user.

among all other classifiers. We find that 87.6% is a good number for streaming tweets considering that the algorithm can work in real-time without having any additional lookup calls. Moreover we find that, both content and user behavior based features are needed for the classification of streaming tweets.

III. Conclusion

In our work we have shown how we detect spam on Twitter at two levels. One at a user level, detecting whether a given user is spam user or not and another at tweet level, detecting whether a given tweet is spam tweet or not.

User level spam detection has been done by various other researchers in the past, but our work enhances the classification F-measure to almost 99.9% by Random Forest classifier. We obtained such a high F-measure because we conclude, content attributes of a tweet are more important than user behavior attributes. Earlier research work focused on both user behavior attributes and content attributes. But any modern spam user can easily trick his user behavior attributes to match with a non-spam user. Hence to detect a spam user content based attributes are needed, and Random Forest is the best classifier. On tweet level classification, not many significant attempts have been made. To the best of our knowledge, ours is the most significant work. We obtain 87.4% F-measure by considering both user behavior attributes and content attributes. In order to do real-time classification of tweets, it is important to perform analysis with the given data without additional lookup calls. We find content attributes alone are not sufficient to classify a tweet, we use user behavior attributes with Random Forest as the best classifier.

ACKNOWLEDGMENT

We are grateful to our mentor Prof. Shrisha Rao of IIIT-B, for his continual guidance and support.

REFERENCES

[1] G. Trends, https://www.google.co.in/trends/, 2016.

[2] U. SEC. (2013) Form 10-k. [Online]. Available: https://www.sec.gov/Archives/edgar/data/1418091/000095012314003031/-twtr-10k_20131231.htm

[3] Benevenuto *et al.*, "Detecting spammers on twitter," in *Seventh annual Collaboration, Electronic messaging, Anti- Abuse and Spam Conference*. CEAS 2010, July 2010.

[4] A. H. Wang, "Don't follow me: Spam detection in twitter," in *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, July 2010, pp. 1–10.

[5] S. P. Release. (2007) Sophos facebook id probe. [Online]. Available: https://www.sophos.com/en-us/press-office/press-releases/2007/08/facebook.aspx

[6] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda, "All your contacts are belong to us: Automated identity theft attacks on social networks," in *Proceedings of the 18th International Conference on World Wide Web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 551–560. [Online]. Available: http://doi.acm.org/10.1145/1526709.1526784

[7] H. Berger, M. Kohle, and D. Merkl, "On the impact of document representation on classifier performance in e-mailcategorization introduction," 2005.

[8] M. McCord and M. Chuah, "Spam detection on twitter using traditional classifiers," in *Proceedings of the 8th International Conference on Autonomic and Trusted Computing*, ser. ATC'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 175–186. [Online]. Available: http://dl.acm.org/citation.cfm?id=2035700.2035717

[9] Sysomos. (2009) An in-depth look inside the twitter world. [Online]. Available: http://sysomos.com/inside-twitter

[10] C. D. M. . A. Stroppa, "Twitter and the underground market," 11th Nexa Lunch Seminar, May 2013.

[11] J. O. et. al. (2013) An in-depth analysis of abuse on twitter. Trend Micro. [Online]. Available: http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-an-in-depth-analysis-of-abuse-on-twitter.pdf

[12] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts, "Who says what to whom on twitter," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 705–714.

[13] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring user influence in twitter: The million follower fallacy." 2010.

[14] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

[15] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai, "Uncovering social network sybils in the wild," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 1, p. 2, 2014.