Twitter Spam Detection using Lfun Approach based on Real -Time Statistical Features

Ms. Rucha. B. Kadam[1], Ms. Shital Y. Gaikwad[2]
[1]*ME Student,* [2]*Professor,*
[12]*Department of computer science, Matoshri Pratishthan Group of Institution Nanded, India*

Abstract- Online social networking is very vast growing growth today's world but attacks on it is more common, amongst them one of the attack is twitter attack in this Spammers spread various malicious tweets which may have form like as links or hash tags on the website and online services, which are too harmful to real users. Late works concentrate on applying machine learning strategies for Twitter spam detection, which make utilization of the measurable highlights of tweets. In the labeled tweets dataset, monitor that the statistical properties of spam tweets differ after some time, and the performance of existing machine learning-based classifiers diminishes. This issue is indicating to as "Twitter Spam Drift". So as to handle this issue, to first study out a profound analysis on the statistical of one million spam tweets and one million non-spam tweets, and after that propose a novel Lfun strategy. The proposed strategy can find "changed" spam tweets from unlabeled tweets and fuse them into classifier's training process. Various experiments are performed to assess the proposed strategy. The outcomes appear that our proposed Lfun strategy can fundamentally enhance the spam detection accuracy in offline as well as online scenarios.

Keywords- Feature Extraction, Machine Learning, Feature Discretization, Social network security, Twitter spam detection

I. INTRODUCTION

Social networking sites such as Twitter, Facebook, Instagram and some enterprise of online social network have become extremely popular in the last few years. Individuals spend vast amounts of time in OSNs making friends with people who they are familiar with or interested in. Twitter, which was founded in 2006, has become one of the most popular micro blogging service sites. Around 200 million users create around the 400 million new tweets per day the growth of spam. Twitter spam, which is referred as unsolicited tweets containing malicious links that directs victims to external sites containing malware spreading, malicious link spreading etc. has not only affected a number of legitimate users but also polluted the whole platform. Machine Learning (ML) based detection strategies involve several steps. First, statistical features, which can differentiate spam from non-spam, are extracted from tweets or Twitter users (such as account age, number of followers or friends and number of characters in a tweet). Then a small set of samples are labeled with class, i.e.spam or non-spam, as training data. After that, machine learning based classifiers are trained by the labeled samples, and finally the trained classifiers can be used to detect spam. However, the observation in our collected data set shows that the characteristics of spam tweets are varying over time. We refer to this issue as "Twitter Spam Drift". As previous ML based classifiers are not updated with the "changed" spam tweets, the performance of such classifiers are dramatically influenced by "Spam Drift" when detecting new coming spam tweets.

The "Twitter spam drift" problem through analyzing the statistical properties of Twitter spam in the collected dataset and then its impact on detection performance of several classifiers. By observing that there are "changed" spam samples in the coming tweets, propose a novel Lfun (Learning from unlabeled tweets) approach, which updates classifiers with the spam samples from the unlabeled incoming tweets.

Motivation:

The proposed system, to identify the "Spam Drift" problem in statistical features based Twitter spam detection. As a result, prior related works by organizing them into two categories:

1. Characterizing Twitter Spam
2. Detecting Twitter Spam

Objectives:

1. To categories the Spam and Non-spam tweets.
2. To work on a performance evaluation such as Precision, Recall, F-measure.
3. To categorize the tag based tweets and link based tweets.
4. To learn from detected spam tweets.
5. To minimize the labeling cost by using different learning criteria to select most informative samples from unlabeled data to be labeled by a human annotator.

II. REVIEW OF LITERATURE

The paper [1] proposes the system analyzes how spammers who target social networking sites operate. To collect the data about spamming activity, system created a large set of "honey-profiles" on three large social networking sites. Advantages are: The deployment of social Honey pots for harvesting deceptive spam profiles from social networking. Statistical analysis of these spam's profiles. Disadvantages are: Mainly time consuming and resource consuming for the system. The paper [2] proposes a spam filtering method for social networks using

relation information between users. System use distance and connectivity as the features which are hard to manipulate by spammers and effective to classify spammers. Advantages are: The spam filtering system will be more powerful. The accuracy is better. Caching technique will help both client-side and server-side to reduce computing overhead. Disadvantages are: The relation feature approach is very difficult to calculate.

The paper [3] represents the behaviors of spammers on Twitter by analyzing the tweets sent by suspended users in retrospect. An emerging spam-as-a-service market that includes reputable and not-so-reputable affiliate programs, ad-based shorteners, and Twitter account sellers. Advantages are: Fledgling spam-as-a-service market - Affiliate programs and Account providers. Disadvantages are: Low barrier to creating accounts,weak defenses and slow response. The paper [4] proposes Monarch is a real-time system for filtering scam, phishing, and malware URLs as they are submitted to web services. Monarch's architecture generalizes to many web services being targeted by URL spam, accurate classification hinges on having an intimate understanding of the spam campaigns abusing a service. Advantages are: It provides 90.78% accuracy for identifying web service spam. Run-time performance is high as 5.54 seconds. Disadvantages are: This system is very expensive.

In the paper [5] presents a methodology based on two new aspects: the detection of spam tweets in isolation and without previous information of the user; and the application of a statistical analysis of language to detect spam in trending topics. In addition, because of growing micro blogging phenomenon and trending topics, spammers can disseminate malicious tweets quickly and massively. Advantages are: Reduced set of features, which used in conjunction with our machine learning system. To divert traffic from legitimate users to spam websites. Disadvantages are: To select the most appropriate features for use in a detection system in real time or reduce the cost even more. The paper [6] presents a theoretically supported framework for active learning from drifting data streams and develops three active learning strategies for streaming data that explicitly handle concept drift. They are based on uncertainty, dynamic allocation of labeling efforts over time, and randomization of the search space. Advantages are: Active strategies reduce the time and space needed for learning. The results suggest that these strategies may be a good way not only to save labeling costs but also to speed up the training process of classifiers while maintaining good accuracy.

In [7] paper, first contribution is to reveal the OSN spam generation techniques according to spam textual patterns. Second contribution is to propose Tangram, a system that performs effective template generation to combat OSN spam. Advantages are: Tangram is the first accurate online OSN spam detection system that detects spam with or without URLs. Tangram, a template based system for accurate and fast OSN spam detection. Tangram automatically divides OSN spam into segments and uses the segments to construct templates to filter future spam. In [8] paper, extracted light-weight features which are able to differentiate spam tweets and non-spam tweets from the labeled dataset.Used CDF figures to illustrate the characteristics of extracted features. Advantages are: The classifiers have much better performance in detection spam tweets on the continuous datasets. Increase the Twitter spam detection accuracy. Disadvantages are: Naïve Bayes and SVM cannot work well in big dataset.

The paper [9] proposes a novel asymmetric learning approach (ASL) to deal with "Twitter Spam Drift". Through our evaluations, we show that our proposed ASL can effectively detect Twitter spam by reducing the impact of "Spam Drift" issue. There are three components: Training Stage, Online Detection, and ASL. Advantages are: To improve detection rate and F-measure of ASL approach. It can reduce the impact of "Spam Drift" significantly. Disadvantages are: It does not facilitate ASL approach for real-time Twitter spam detection. The paper [10] proposes a classification algorithm that operates using three linguistic attributes of a user's text. The algorithm analyzes (i) the average URL count per tweet, (ii) the average pairwise lexical dissimilarity between a user's tweets, and (iii) the word introduction rate decay parameter of the user for various proportions of time-ordered tweets. A flexible and transparent classification scheme, we have demonstrated the potential of using linguistic features as a means of classifying automated activity on Twitter. Advantages are: The accuracy of the classifier increases with the number of collected tweets. It is flexible. Disadvantages are: It does not analyze multilingual classification scheme.

III. OPEN ISSUES

The existing system, to detect Twitter spam, made use of account and content features, such as account age, number of followers or followings, URL ratio, and the length of tweet to distinguish spammers and nonspammers. To make Twitter as a clean social platform, security companies and researchers are working hard to eliminate spam. Security companies mainly rely on blacklists to filter spam links. However, blacklists fail to protect users on time due to the time lag. To avoid the limitation of blacklists, some early works proposed by researchers use heuristic rules to filter Twitter spam. A simple algorithm used to detect spam in #robotpickupline (the hashtag was created by them) through these three rules: suspicious URL searching, username pattern matching and keyword detection. Remove all the tweets which contained more than three hashtags to filter spam in their dataset to eliminate the impact of spam.

Although there are a few works, such as and which are suitable to detect streaming spam tweets, there lacks of a performance evaluation of existing machine learning-based streaming spam detection methods. In this paper, we aim to bridge the gap by carrying out a performance evaluation, which was from three different aspects of data, feature, and model. Others apply existing blacklisting service, such as Google Safe Browsing to

label spam tweets. Nevertheless, these services' API limits make it impossible to label a large amount of tweets. However, Twitter has around 5% spam tweets of all existing tweets in the real world.

Disadvantages:

1. The lack of a performance evaluation of existing machine learning-based streaming spam detection methods.
2. The existing machine learning-based spam detection methods suffer "Spam Drift" problem.
3. The detection accuracy will decrease as time goes on, since spammers are changing strategies to avoid being detected.

IV. SYSTEM OVERVIEW

Existing machine learning based spam detection methods suffer from the problem of "Spam Drift" due to the change of statistical features of spam tweets as time goes on. To solve this problem, obtaining the "changed" samples to update the classification model is very important. By observing that there are such samples in the unlabeled incoming tweets which are very easy to collect, proposed a scheme called "Lfun" to address "Spam Drift" problem.

The framework of the proposed scheme shows in Fig.1. There are two main components in this framework: LDT is to learn from detected spam tweets and LHL is to learn from human labeling. Consequently, the research community, as well as Twitter itself, has proposed some spam detection schemes to make Twitter as a spam-free platform. For instance, Twitter has applied some "Twitter rules" to suspend accounts if they behave abnormally. Those accounts, which are frequently requesting to be friends with others, sending duplicate content, mentioning others users, or posting URL-only content, will be suspended by Twitter. Twitter users can also report a spammer to the official @spam account. To automatically detect spam, machine learning algorithms have been applied by researchers to make spam detection as a classification problem. Most of these works classify a user is spammer or not by relying on the features which need historical information of the user or the exiting social graph. For example, the feature, "the fraction of tweets of the user containing URL" used in must be retrieved from the users' tweets list; features such as, "average neighbors' tweets" in and "distance" in cannot be extracted without the built social graph. However, Twitter data are in the form of stream, and tweets arrive at very high speed. Despite that these methods are effective in detecting Twitter spam, they are not applicable in detecting streaming spam tweets as each streaming tweet does not contain the historical information or social graph that are needed in detection.

A. *Architecture*

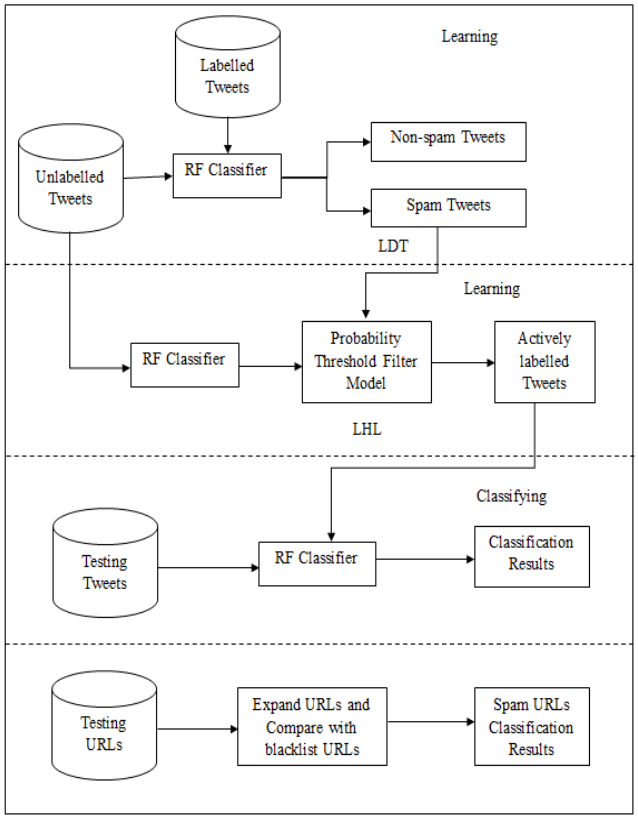

Fig.1: Proposed System Architecture

Advantages:

1. Extraction of 12 features and categories as Tag based features and URL based features.
2. The Lfun approach can be deployed without much training data at the beginning, but to be updated when new training data comes.
3. Automatically updated with detected spam tweets with no human effort.
4. The system implements a method which will use spot filter mechanism to detect whether the post is spam or not.
5. The system implements application can also be hosted online for its use and the data will be stored and fetched from server.
6. User with maximum number of spam can be blocked from the system.
7. Performance evaluation done on Dataset by using TPR, FPR, Precision, Recall and F-measure.

B. *Mathematical Model*

1. Learning from Detected Spam Tweets

In a LDT learning method, given a labeled data set $T_l = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, containing$m$ labelled tweets, where $x_i \in \square^k (i = 1, 2, \dots, m)$is thefeature vector of a tweet,

$y_i \in$ {*spam, non-spam*} is thecategory label of a tweet. Also given a large data set$T_u = \{(x_{m+1}, y_{m+1}), (x_{m+2}, y_{m+2}), \ldots, (x_{m+n}, y_{m+n})\}$containing$n$ unlabelled tweets ($n >> m$). Then a classifier φistrained by T_l. φcan be used to divide $T_u$into spam $T_{spam}$and non-spam $T_{non-spam}$. The LDT function is calculated by

$$\varphi : \square^k \rightarrow \{spam, non-spam\} \quad (1)$$

2. Learning from Human Labelling

In supervised Twitter spam detection, we are given a labelledtraining data set $T_{training} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$,containing m labelled tweets, where $x_i \in \square^k (i = 1, 2, \ldots, m)$is the feature vector of a tweet, $y_i \in$ {*spam, non-spam*} isthe category label of a tweet. The label $y_i$of a tweet *xi* isdonated as *y=f(x)*. The task is then to learn a function *f* which can correctly classify a tweet to spam or non-spam.

C. Algorithms

1. Random Forest Algorithm

Step 1: Let the number of training cases be *N*, and the number of variables in the classifier be *M*.

Step 2: The number *m* of input variables to be used to determine the decision at a node of the tree; *m* should be much less than *M*.

Step 3: Choose a training set for this tree by choosing *n* times with replacement from all *N* available training cases (i.e. take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes.

Step 4: For each node of the tree, randomly choose *m* variables on which to base the decision at that node. Calculate the best split based on these *m* variables in the training set.

Step 5: Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier).

For prediction a new sample is pushed down the tree. It is assigned the label of the training sample in the terminal node it ends up in. This procedure is iterated over all trees in the ensemble, and the average vote of all trees is reported as random forest prediction.

2. Lfun Algorithm

Input: labelled training set$\{\psi_1, \ldots, \psi_N\}$, unlabelled tweets $T_{unlabelled}$, a binary classification algorithm Φ

Output: manually labelled selected tweets T_m

Process:

Step 1: $T_{labelled} \leftarrow \bigcup_{i=1}^{N} \psi_i$

// Use Φto create a classifier *Cls*from $T_{labelled}$

Step 2: *Cls*←Φ: $T_{labelled}$

// $T_{unlabelled}$is classified as $T_{spam}$and $T_{non-spam}$

Step 3: $T_{spam} + T_{non-spam} \leftarrow T_{unlabelled}$

// Merge spam tweets T_{spam}classified by *Cls*into $T_{labelled}$

Step 4: $T_{ex} \leftarrow T_{labelled} + T_{spam}$

// use $T_{ex}$to re-train the classifier *Cls*

Step 5: *Cls*←Φ: T_{ex}

// determine the incoming tweet's suitability for selection

Step 6: $U \leftarrow \phi$

Step 7: **for** *i* = 1 **to** *k* **do**

Step 8: **if** $U_i$meet the selection criteria *S* **then**

Step 9: $U \leftarrow (U \cup U_i)$

Step 10: **end if**

Step 11: **end for**

// manually labelling each $\square_{\square}$in *U*

Step 12: *Tm* ←∅

Step 13: **for** *i* = 1 **to** *k* **do**

Step 14: manually label each $\square_{\square}$

Step 15: $\square_{\square} \leftarrow \square_{\square} \cup \square_{\square}$

Step 16: **end for**

V. RESULTS AND DISCUSSIONS

Experimental evaluation results shows the offline and real-time tweets dataset with higher percentage of spam tweets have better performance because when fraction of spam tweets increases, probability for a tweet to be a spam tweet increases and as a result more spam tweets will be labeled as spam tweets. The results evaluate the performance of the proposed Lfun scheme in detecting "drifted" Twitter spam, by using F-measure and detection rate with accuracy.

Each classifier in this set of experiments was trained with a dataset of 5000 spam tweets and 5000 nonspam tweets. Then, these trained classifiers were used to detect spam in the two sampled datasets. We also used TPR, FPR, and F-measure to evaluate the performance of these classifiers. The fig. 2 shows the spam and nonspam count of real-time testing dataset. Finally, the Fig. 3 displays the performance graph of Random Forest using Lfun scheme.

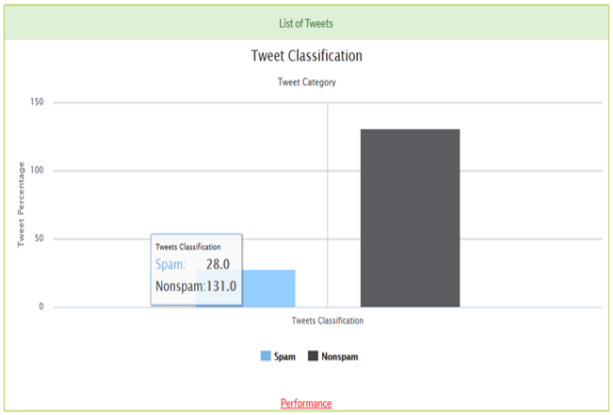

Fig.2: Real-time tweets classified result using Lfun Approach

Fig.3: Performance analysis of Lfun Approach

VI. CONCLUSION

In this paper, provide a fundamental evaluation of ML algorithms on the detection of streaming spam tweets. In the evaluation, found that classifiers' ability to detect Twitter spam reduced when in a near real-world scenario since the imbalanced data brings bias. Also identified that Feature discretization was an important preprocesses to ML-based spam detection. Second, increasing training data only cannot bring more benefits to detect Twitter spam after a certain number of training samples. In this paper, firstly identify the "Spam Drift" problem in statistical features based Twitter spam detection. In order to solve this problem, proposes an Lfun approach. In the Lfun scheme, classifiers will be re-trained by the added "changed spam" tweets which are learnt from unlabelled samples, thus it can reduce the impact of "Spam Drift" significantly. There is also a limitation in our Lfun scheme. The benefit of "old" labelled spam is to eliminate the impact of "spam drift" to classify more accurate spam tweets in future days. The effectiveness of "old" spam has been proved by our experiments during a short period. However, the effectiveness will decrease as the correlation of "very old" spam becomes less with the new spam in the long term run. In the future, incorporate incremental adjustment to adjust the training data, such as dropping the "too old" samples after a certain time. It can not only eliminate unuseful information in the training data but also make it faster to train the model as the number of training samples decrease.

VII. REFERENCES

[1]. K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: social honeypots + machine learning," in *Proc. 33rd Int. ACM SIGIR Conf. Res.Develop. Inf. Retrieval*, 2010, pp. 435–442.

[2]. J. Song, S. Lee, and J. Kim, "Spam filtering in Twitter using sender receiver relationship," in *Proc. 14th Int. Conf. Recent Adv. Intrusion Detection*, 2011, pp. 301–317.

[3]. K. Thomas, C. Grier, D. Song, and V. Paxson, "Suspended accounts in retrospect: An analysis of Twitter spam," in *Proc. ACM SIGCOMM Conf. Internet Meas.*, 2011, pp. 243–258.

[4]. K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time URL spam filtering service," in *Proc. IEEE Symp. Sec. Privacy*, 2011, pp. 447–462.

[5]. J. M. Romo and L. Araujo, "Detecting malicious tweets in trending topics using a statistical analysis of language," *Expert Syst. Appl.*, vol. 40, no. 8, p. 2992–3000, 2013.

[6]. I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with drifting streaming data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 1, pp. 27–39, Jan. 2014.

[7]. H. Gao*et al.*, "Spam ain't as diverse as it seems: Throttling OSN spam with templates underneath," in *Proc. 30th Annu. Comput. Security Appl. Conf.*, 2014, pp. 76–85.

[8]. Chen, J. Zhang, X. Chen, Y. Xiang, and W. Zhou, "6 million spam tweets: A large ground truth for timely twitter spam detection," in *Proc. IEEE Commun. Inf. Syst. Security Symp. (ICCCISS)*, Jun. 2015, pp. 8689–8694.

[9]. Chen, J. Zhang, Y. Xiang, and W. Zhou, "Asymmetric self-learning for tackling twitter spam drift," in *Proc. 3rd Int. Workshop Security Privacy Big Data (BigSecurity)*, Apr. 2015, pp. 237–242.

[10]. M. Clark, J. R. Williams, C. A. Jones, R. A. Galbraith, C. M. Danforth, and P. S. Dodds, "Sifting robotic from organic text: A natural language approach for detecting automation on twitter," *J. Comput. Sci.*, vol. 16, p. 1–7, Sep. 2016.