Real-Time Detection of Changing Spam Tweets in Twitter

P. Jayashree[1], M. Manikandan[2], M. Jayanthi[3]

[1]*Student, Department of CSE, Vidyaa Vikas college of Engineering and Technology, Tiruchengode, India*

[2]*Professor & HoD, Dept. of CSE, Vidyaa Vikas college of Engineering and Technology, Tiruchengode, India*

[3]*Assistant Professor, Department of CSE, K.S.R Institute of Technology, Tiruchengode, India*

***Abstract*: Now-a-days, there is hike in usage of social networks as Twitter, Facebook, Google and so on. The Twitter has a critical problem like Twitter Spams as spam tweets, spammers accounts etc. These Twitter Spams are detected by using existing Machine Learning Algorithms. This issue is known to be as "Twitter Spam Drifts". Here, analysing the tweets as spam tweets and non – spam tweets with the statistical features. The experiments performed to evaluate the proposed scheme and the result shows improvement in Spam Detection accuracy in real – world.**

***Keywords*: Machine learning, Twitter Spam Drifts**

1. Introduction

Among Social Networks, Twitter became most popular in the society. It has number of users from youngsters, politicians, public figures, society etc., The growth of the Twitter has increased spamming activities that thefts the users' data and information with malicious link contains external sites as malware downloads, phishing or scams. There are different spams with pictures of stars, unsecured external links or file downloads [12].

In Twitter itself, for detecting spams or spammers a Spam – Free Platform is introduced. It will block or filter the tweets that are unrelated. Twitter implemented blacklist filtering known to be "BotMaker", to protect from victims of spammers or spam tweets. Mostly the 90% of victims visit a new spam link in blacklist. There are some Statistical features for detecting Spammers or Spam tweets without checking URLs by the Machine Learning [5][6][15]. In Machine Learning (ML) based detection method involve several steps. First, Statistical features can differentiate spam from non-spam that are extracted from tweets or twitter users by using account age, number of followers or friends and number of characters in a tweet. For training data a small set of samples are labelled with class (i.e spam/non-spam).After the machine learning based classifiers are trained by the labelled samples, it is used for spam detection.

The characteristics of spam tweets are varying over time this issue is referred as Twitter spam drift. ML based classifiers are not updated with the changed spam tweets by using spam Drift the new incoming spam tweets. spam tweets drift over time, because spammers struggling with security companies and researchers. spammers are trying to avoid being detected by the researchers. This leads the spammer to avoid being detected by the researchers through using more tweets or creating spam with similar semantic meaning but using different text. Here LFS (Learning from spam tweets) approach is used it will update the classifiers with the spam samples from the unlabelled incoming tweets. Here real-world dataset is used which contains 10 consecutive days tweets with spam and non-spam tweets. LFS approach which learns from the unlabelled tweets to deal with "Twitter spam Drift". Lfs can effectively detect twitter spam by reducing the impact of "spam Drift" issue.

Table 1
Feature and Description

Feature No.	Feature Name	Description
f1	Account_age	The age of an account
f2	No_follower	The number of followers
f3	No_following	The number of followings
f4	No_userfavorites	The number of favourites this user received
f5	No_lists	The number of lists the user is member of
f6	No_tweets	The number of user posted tweets
f7	No_retweets	The number of times this tweet has been retweeted
F8	No_hashtag	The number of hashtags in this tweet
F9	No_usermention	The number of times this tweet being mentioned
F10	No_urls	The number of URLs contained in this tweet
f11	No_char	The number of characters in this tweet
f12	No_digits	The number of digits in this tweet

2. Problem of twitter spam drift

A. Data collection

Twitter's Streaming API is used to collect tweets with URLs in a period of 10 consecutive days. It is also possible to send spam without embedding URLs on twitter, but majority of the spam contain URLs. Spammers use embedded. URLs because it is more convenient to Direct victims to external sits for scams, phishing and malware downloading [1]. Now-a-days researchers use ground-truth, manual inspection and blacklist filtering. Manual inspection uses small amount of labelled training data. Human intelligence task websites help in labelling the tweets but it is costly. Blacklisting service such as Google Safe Browsing and URIBL to label spam tweets.

B. Problem statement

The spam tweets features are changing in unpredicted ways over time. The result of machine learning algorithm becomes inaccurate. It is referred to as "spam drift" problem.

C. Problem justification

To recognize the changing of statistical features in a dataset, a natural approach is to model the spreading of the data. Two approaches are used: parametric and non-parametric. parametric method is utilized when the particular distribution of the dataset is known. In non-parametric Methods, such as statistical tests, which make no rules of the dataset distributions used by the researchers. The statistical tests are to calculate the distance of two distributions to determine the change. One of the most common measures is to compute the distance of two distributions. Common measure to compute the distance of distributions is kullback Leibler(KL) Divergence. KL Divergence, which is also known as relative entropy is defined as

$$D_{kl}(p\,||Q) = (x+a)^n = \sum_{i} p(i) log \frac{p(i)}{Q(i)}$$

It is utilized to compare two probability distributions. To plot data points into distributions to apply the formula. Let s = { x_1,x_2, x_3………x_n } be a multi-set from a finite set F containing numerical feature values, and denote N(x|s) the number of presences of x ∈ s, thus the relative proportion of each x is denoted by

$$P_s(x) = \frac{N(x|s)}{n}$$

The ratio of p/q is undefined if Q(i)= 0. The estimate P_s is replaced as,

$$P_s(x) = \frac{N(x|s) + 0.5}{n + |F|/2}$$

|F| is the number of elements in the finite set F. The distance between two day's tweet, D1 and D2 is,

$$\mathrm{D(D1||D2)} = \sum_{x\varepsilon Fi} P_{D1}(X) log \frac{P_{D1}(x)}{P_{D2}(x)}$$

3. Proposed scheme: LFS

Machine Learning algorithm has the difficult of "spam drift" due to the change of statistical features of spam tweets. In "spam drifts" old classification model is not updated with "changed" spam samples so the result became incorrect. This problem can be solved by update the classification model of "changed samples".

The components in this framework: LDT is to learn from spotted spam tweets and LHL is to learn from human labelling. In "Drifted Spam Detection" state, we have already got a small quantity of labelled spam and non-spam tweets. However, there are not sufficient samples of "changed" spam. It is costly to have human label a large amount of "changed" tweets. we make use of the above stated two components to mechanically extract "changed" spam tweets from a set of unlabelled tweets, which are very easy to collected from Twitter. Once getting enough labelled "changed" spam tweets, we implement the scheme which employs a sufficiently powerful algorithm, Random Forest, to perform classification. Our LFS scheme is summarised in Algorithm 1.

They are two main components in this framework:

1) LDT is to learn from detected spam tweets
2) LHL is to learn from human labelling

A. Learning from detected spam tweets

LDT is used to deal with a classification scenario where there is a sufficiently robust algorithm, but in want of more data. By learning from a large number of unlabelled data, LDT can obtain sufficient new information, which can be used to update the classification model.

In a LDT learning scenario, we are given a labelled data set T_1= {(x_1,y_1) , x_2,y_2) ,…….,(x_m,y_m)}, , containing m labelled tweets, where $x_i \in R^k (i = 1,2, \dots \dots . . m)$ is the feature vector of a tweet, y_i ∈ {spam, non-spam} is the category label of a tweet. Then a classifier φ is trained by T_1. φ can be used to divide T_u into spam T_{spam} and non-spam $T_{non-spam}$. Labelled spam tweets from T_u will be added into the labelled data set T_1 to form a new training data set.

The basic of LDT is to find a function: $R^k \rightarrow$ {spam, non-spam} to predict the label y ∈ {spam, non-spam} of new tweets when trained by T_{1+spam}, which is the combination of the labelled data set T_1 and spam tweets T_{spam} identified from T_u. Mainly, the unlabelled data set T_u used in LDT does not have to share the similar distribution with the labelled data set T_1 . In addition, only identified spam tweets will be added into the training data. The reason is that, we've already gained adequate information of non-spam tweets, as the statistical properties are not varying for non-spam tweets. It is not necessary for us to increase more information about non-spam tweets.

However, the spam tweets identified by the classifier that is trained using T_1 also have the same or similar distribution of old spam. We need samples from "changed spam" to adjust the classifier.

B. *Learning from human labelling*

In a supervised spam detection system, a learning algorithm, such as Random Forest, must be trained by enough labelled data to find more accurate detection results. However, labelled occurrences are very expensive and time-consuming to get. Fortunately, we have a huge number of unlabelled tweets which can be easily collected. The LHL in our LFS is best suitable where there are many unlabelled data cases, and human annotator anticipating to label many of them to train an accurate system. LHL goal is to reduce the labelling cost by using different learning principles to select most useful samples from unlabelled data to be labelled by a human annotator. We also import active learning in our LFS scheme.

Now let us define our learning component in a formal way. In supervised Twitter spam detection, we are given a labelled training data set $T_{training}$={(x_1,y_1) , $x_2,y_2)$,…….,(x_m,y_m)}, , containing m labelled tweets, where $x_i \in R^k(i = 1,2, \dots \dots . . m)$ is the feature vector of a tweet, y_i ∈ {spam,non-spam} is the category label of a tweet. The label $y_i$of a tweet x_i is donated as y = f(x). The task is then to learn a function $\hat{f}$ which can correctly classify a tweet to spam or non-spam.

We use generalisation error to measure the accuracy of the learned function:

Error ($\hat{f}$) =$\sum_{x \in T_{training}} L(f(x)\,, f(\hat{x})$ P(x).

f(x) is not available for testing data instances. Therefore, it is usual to estimate the generalisation error by the test error:

Error ($\hat{f}$) =$\sum_{x \in T_{testing}} L(f(x)\,, f(\hat{x})$ P(x),

where $T_{testing}$ refers to the testing tweets, and prediction error can be measured by a loss function L, such as mean squared error (MSE)

$$L_{MSE}\big(f(x), f(\hat{x})\big) = (f(x) - f(\hat{x}))^2$$

The learning criteria is set to select the most useful instances $X_{selected}$ and add them to the training set $T_{training}$ for achieving some certain ideas. Let us consider this objective as the minimization of generation error of a learned function trained by $T_{training}$. So the learning criteria can be donated as

Error($T_{training}$ U {$X_{selected}$}).

The goal of this kind of learning is to select instances $X_{selected}$ which can reduce the generalisation error:

Error($X_{selected}$):argmin Error ($X_{selected}$).

In LFS scheme, we apply the selection criteria, called "Probability Threshold Filter Model", to select the most useful tweets to tackle "Spam Drift". In order to attain this, Random Forest (RF) is used to determine the possibility of a tweet whether it belongs to spam or not.

Random Forest can yield many classification trees after being trained with T_{ex} from as a result, good selection criteria must be estimated to minimize the error. Asymmetric Self-Learning. When classifying a new arriving tweet, each tree in the forest will give a class estimate. Then forest selects the classification result which has the most votes. In our case, we set the number of trees to m, if n trees vote for the class "spam", the probability of the tweet to be classified as "spam" is Pr = $\frac{n}{m}$. thus set the threshold τ to Pr ∈ [0:4; 0:7].

After we pre-filter some candidate tweets to be labelled using the "Probability Threshold Filter Model", the number of tweets is still too many. We then randomly select a smaller number of tweets from the candidate tweets (we set it to be 100 in our experiments) to be manually labelled. As shown in Fig. 3, the manually labelled tweets, along with T_{ex} will be used to train a new classifier, which can "Spam Drift" problem.

C. *Performance Benefit Justification*

We use three normal distributions (listed below) to simulate this: ω_0 represents the distribution of non-spam, while ω_1 and ω_2 represents the distribution of spam before and after using our LFS approach, respectively.

$$\begin{cases} \omega_0 \ \sim N(\mu_{0,}\sigma_0^2) \\ \omega_1 \ \sim N(\mu_{1,}\sigma_{12}^2) \\ \omega_2 \ \sim N(\mu_{2,}\sigma_{12}^2) \end{cases}$$

ω_0 represents the distribution of non-spam

ω_1 and ω_2 represents the distribution of spam before and after using LS approach

$$m - c_2 = \mu_1 - \mu_2$$
$$p_1(m) = p_2(c_2)$$

As $c_2 < c_1$, we have

$$p_0(c_2) < p_0(c_1)$$

We also have

$p_0(c_2) = p_1(c_1),\ p_0(c_2) = p_2(c_2)$.

From Equation, 3 and Equation 4, we get

$$p_1(c_1) > p_2(c_2)$$

From Equation, 2 and Equation 5, we get

$$p_1(c_1) > p_1(m)$$

As a result

$$m > c_1$$

Taking into account Equation. 7 and Equation 1, we can have $c_1 - c_2 < \mu_1 - \mu_2$. So,

$$c_2 - \mu_2 > c_1 - \mu_1.$$

The error rate of classification before LFS,

$P_1(error) = P(x > c_1) + p(x < c_2)$

=$\int_{c_1}^{\infty} p_1(t)\,dt + \int_{-\infty}^{c_1} p_0(t)dt$

=1 - $\varphi(\frac{c_1-\mu_1}{\sigma_{12}}) + \varphi(\frac{c_1-\mu_0}{\sigma_0})$

Similarly, we have the error rate after using LFS

$P_2(error) = 1 - \varphi(\frac{c_2-\mu_2}{\sigma_{12}}) + \varphi(\frac{c_2-\mu_0}{\sigma_0})$

The difference of P1(error) and P2(error),

$P_1(error) - P_2(error) = \left[\varphi\left(\frac{c_2-\mu_2}{\sigma_{12}}\right) - \varphi\left(\frac{c_1-\mu_1}{\sigma_{12}}\right)\right] + \left[\varphi\left(\frac{c_1-\mu_0}{\sigma_0}\right) - \varphi\left(\frac{c_2-\mu_0}{\sigma_0}\right)\right]$,

while

$$\varphi(x) = \frac{1}{\sqrt{2\pi}}\int_0^x e^{-t^2/_2}\,dt$$

The differentiation of Equation 10 is $\varphi'(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ >0. So, we can have Ø(a) >Ø(b) when a > b. From Equation. 8, we know$\frac{c_2-c_2}{\sigma_{12}} > \frac{c_1-c_1}{\sigma_{12}}$. Consequently,

$$\varphi\left(\frac{c_2-\mu_2}{\sigma_{12}}\right) > \varphi\left(\frac{c_1-\mu_1}{\sigma_{12}}\right).$$

As $c_1 - c_2$, we have $\frac{c_1-\mu_0}{\sigma_0} > \frac{c_2-\mu_0}{\sigma_0}$ .Then, we know

$$\varphi\left(\frac{c_1-\mu_0}{\sigma_0}\right) > \varphi\left(\frac{c_2-\mu_0}{\sigma_0}\right).$$

Substitute Equation. 11 and 12 into 9, we will have

$P_1(error)$ - $P_2(error)$ > 0

Obviously, our proposed approach can effectively reduce the probability of error from Equation 13.

4. Performance evaluation

We calculate the performance of the proposed LFS scheme in detecting "drifted" Twitter spam. To measure the performance by using F-measure and detection rate. F-measure is an evaluation metric which consolidates precision and recall to measure the per-class performance of classification or detection algorithms. It can be calculated by,

F-measure = $\frac{2*precision*Recall}{Precision+Recall}$

Detection rate is defined as the ratio of number of tweets correctly classified as belonging to class spam to the total number of tweets in class spam, it can be calculated by

Detection Rate = $\frac{TP}{TP+FN}$

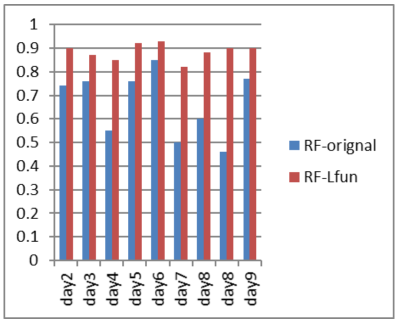
To show the impacts of spam drift we have designed three sets of experiments.

A. *Impacts of spam drift*

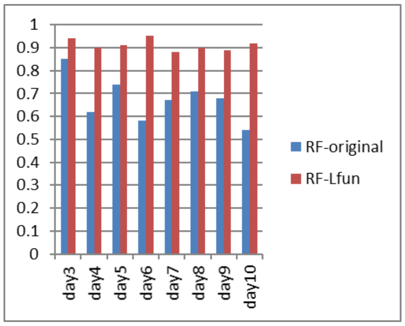
The performance of a traditional classifier for instance c4.5 Decision tree, fluctuates over time when "spam drift" exists. In these examinations, Day 1 information is separated into two sections half to training set, and half to testing set. A supervised classification algorithm is trained with both spam and non-spam tweets from the training pool.

LFS algorithm:

```
Require: labelled training set $\{\varphi_1.....\varphi_N\}$,
unlabelled tweets $T_{\text{unlabelled}}$,
a binary classification algorithm Ø,
Ensure: manually labelled selected tweets $T_m$ ,
$T_{labelled} \leftarrow U^{N}_{i=1}\ \varphi_i$
// Use Ø to create a classifier Cls from $T_{labelled}$:
Cls $\leftarrow \varphi : T_{labelled}$
// $T_{labelled}$ is classified as $T_{spam}$ and $T_{non-spam}$:
$T_{spam} + T_{non-spam} \leftarrow T_{unlabelled}$
// Merge spam tweets $T_{spam}$ classified by Cls into $T_{labelled}$ :
$T_{ex} \leftarrow T_{labelled} + T_{spam}$
// use $T_{ex}$ to re-train the classifier Cls :
Cls $\leftarrow \varphi : T_{ex}$
// determine the incoming tweet's suitability for selection:
U ← $\varphi$
for i = 1 to k do
    if $U_i$ meet the selection criteria S then
        U ← (U $U$ $U_i$)
    end if
end for
// manually labelling each $U_i$ in U
$T_m \leftarrow \varphi$ ;
for i = 1 to k do
    manually label each $U_i$
    $T_m \leftarrow (T_m\ U\ U_i)$
end for
```

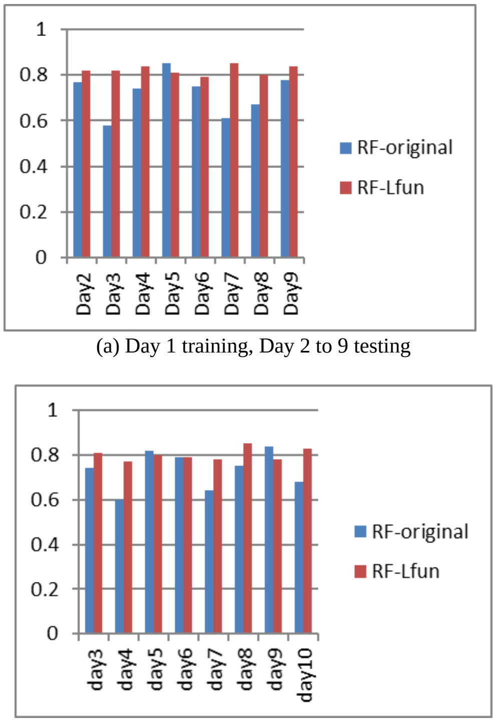


(a) Day 1 training, Day 2 to 9 testing



(b) Day 2 training, Day 3 to 10 testing

Fig. 2. Detection Rate of LFS



(a) Day 1 training, Day 2 to 9 testing

(b) Day 2 training, Day 3 to 10 testing

Fig. 3. F-measure of LFS

B. *Performance of LFS*

Performance can be calculated using F-measure and Detection Rate. Fig. 2 shows the detection rate when day1 or day 2 is used for training and the rest days is used for testing. From the Fig. 6a detection rate for random forest is low as compared to LFS. RF-LFS achieves 80% detection rate but random forest achieves 45% to 80%. same result get when training the data is from Day 2 and testing data is from Day 3 to Day 10. The highest detection rate of random forest is around 85% but RF-Lfs is about 95%.Fig. 3 shows the F-measure of original random forest and Random Forest using Lfs. From Fig. 7 shows that Lfs becomes stable, which is always greater than 80%, except on Day 6. But F-measure of Lfs-RF is not fluctuating as shown in Fig. 3 (b).

C. *Comparisons with other Algorithms*

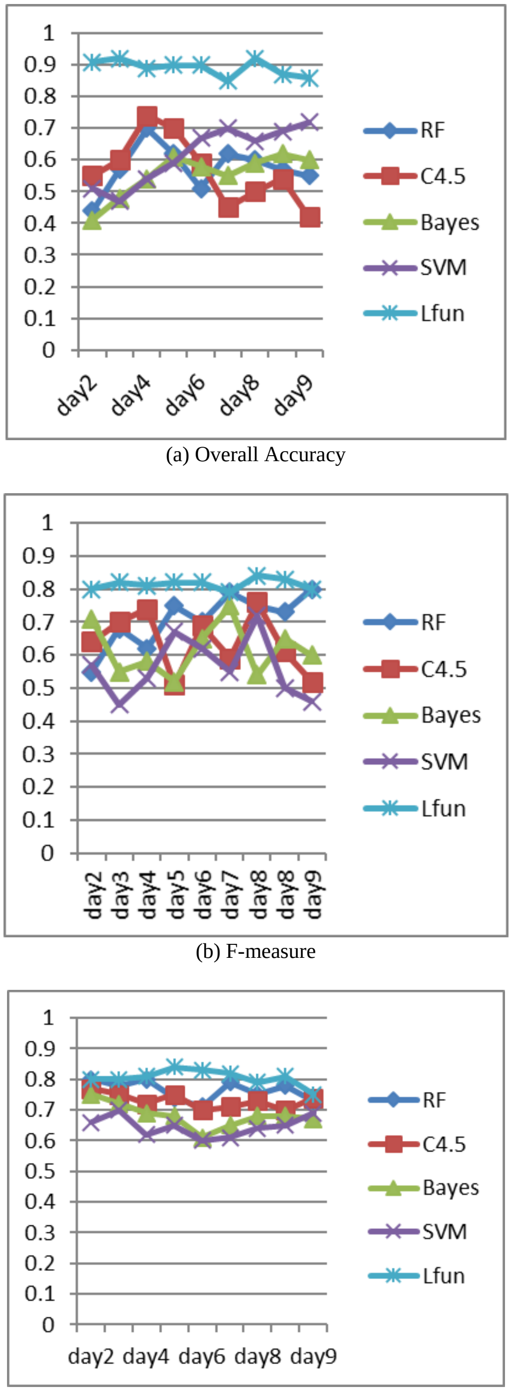
In this section, Lfs approach is compared with four traditional machine learning algorithms such as Random Forest, C4.5, Decision Tree, Bayes Network and SVM, to detect spam tweets in the “drift” scenario. In this two set of experiments carried out.one set is to evaluate the performance while training data is from Day 1, and testing data are varying from Day 2 to Day 9. Another set is to evaluate the performance when training and testing data are from two specified days.

1) Comparisons with changing Days

Fig. 4, shows the experimental results in terms of accuracy, F-measure and detection rate of Lfs compared to other algorithms, for different testing days. Fig 4 (a). shows the overall accuracy of Lfs, Random Forest, C4.5, Decision Tree, Bayes Network and SVM. From Fig. 4 (b) shows that Lfs is best among all algorithms. From Fig. 8c shows the detection rate of Lfs is above 85%. The detection rate of all other algorithms is below 80%.

2) Comparisons with changing Labelled Training Samples

Fig. 5, shows the training and testing data is from Day 1 and Day 5. Fig. 5 shows the training and testing data is from Day 4 and Day 8. Lfs increases from 70% to 80% with the increase of labelled training samples. In F-measure the performance of Lfs is best as compared to other algorithms.

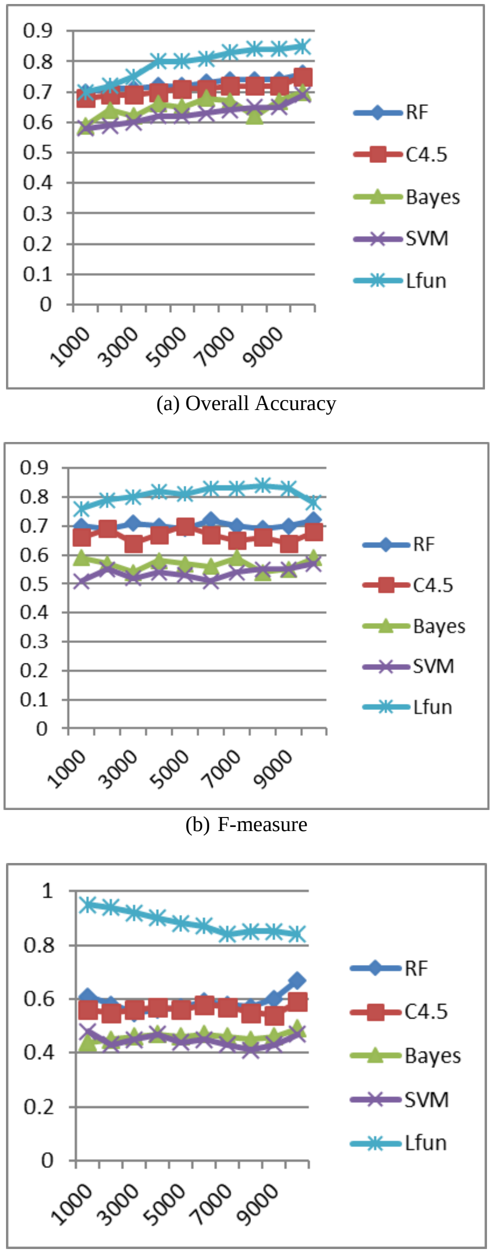


(a) Overall Accuracy

(b) F-measure

(c) Detection Rate

Fig. 4. Comparisons with other Algorithms (changing testing days)



(a) Overall Accuracy

(b) F-measure

(c) Detection Rate

Fig. 5. Comparisons with other Algorithms (training on Day 1 and testing on Day 5)

5. Discussions

They are some machine learning approaches associated to our proposed work. E.g.: online learning and incremental learning. They are common machine learning algorithms, can be update with training data for better result. They can generate a classification model with less training data at first, and update the model by adding new training data.

LFS also has the advantage of online learning and incremental learning. That is at the beginning it can be deployed without much training set later to be updated with the new training data. The LDT component learns from the detected spam tweets. It is automatically updated with the detected spam tweets without any human effort. To improve the performance of prediction model, by using LHL component, which learns from human labelling. To minimize human effort, LHL samples only small number of tweets for labelling. In LHL it does not randomly pick up, it pick up by selection criteria called "Probability Threshold Filter Model" it choose only useful tweets.

6. Conclusion and future work

In this paper, first identify the "spam Drift" problem in statistical features-based Twitter spam detection. To solve this problem, we introduce Lfs approach. In Lfs scheme, classifier is re-trained by the added "changed spam" tweets which are learnt from unlabelled samples, thus it reduces the spam drift problem significantly. The performance of Lfs is evaluate using Detection rate and F-measure. By using LFS scheme detection rate and F-measure result is improved.

The limitation of Lfs scheme is too dropping the "too old" samples from the training set after a certain time. By drooping this it not only eliminate the unusual information sin the training data it also make it faster to train the model.

References

[1] Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In Proceedings of the 20th international conference on World wide web,'11, pages 675–684, New York, NY, USA, 2011.ACM

[2] E. 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. Journal of Computational Science, 16:1 – 7, 2016.

[3] I. Csiszar and J. K¨orner. Information theory: coding theorems for discrete memoryless systems. Cambridge University Press, 2011.

[4] C. Chen, J. Zhang, Y. Xiang, and W. Zhou. Asymmetric Self-Learning for tackling twitter spam drift. In The Third International Workshop on Security and Privacy in Big Data (BigSecurity 2015), pages 237–242, Hong Kong, Hong Kong, Apr. 2015.

[5] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammer on twitter. In Seventh Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference, July 2010.

[6] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In Proceedings of the 26th Annual Computer Security Applications Conference, ACSAC '10, pages 1–9, New York, NY, USA,2010. ACM.

[7] S. Yardi, D. Romero, G. Schoenebeck, and D. Boyd. Detecting spam in a twitter network. First Monday, 15(1-4), January 2010.

[8] J. a. Gama, I. ˇ Zliobait˙e, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. ACM Comput. Surv., 46(4):44:1–44:37, Mar. 2014.

[9] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, IMC '10, pages 35–47, New York, NY, USA, 2010. ACM.

[10] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier. TweetCred: Real-Time Credibility Assessment of Content on Twitter, pages 228–243.Springer International Publishing, Cham, 2014.

[11] R. Jeyaraman. Fighting spam with botmaker. Twitter Engineering Blog, August 2014.

[12] C. Yang, R. Harkreader, J. Zhang, S. Shin, and 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, pages 71–80, New York, NY, USA, 2012. ACM.

[13] S. Yardi, D. Romero, G. Schoenebeck, and D. Boyd. Detecting spam in a twitter network. First Monday, 15(1-4), January 2010.

[14] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and Y. Xiang. Internet traffic classification by aggregating correlated naive bayes predictions. Information Forensics and Security, IEEE Transactions on, 8(1):5–15, Jan. 2013.

[15] X. Zhang, S. Zhu, and W. Liang. Detecting spam and promoting campaigns in the twitter social network. In Data Mining (ICDM), 2012 IEEE 12th International Conference on, pages 1194–1199, 2012.

[16] I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes. Active learning with drifting streaming data. IEEE transactions on neural networks and learning systems, 25(1):27–39, 2014.

[17] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In Proceedings of the 24th international conference on Machine learning, pages 759–766.ACM, 2007.

[18] J. Oliver, P. Pajares, C. Ke, C. Chen, and Y. Xiang. An in-depth analysis of abuse on twitter. Technical report, Trend Micro, 225 E. John Carpenter Freeway, Suite 1500 Irving, Texas 75062 U.S.A., September 2014.

[19] J. Martinez-Romo and L. Araujo. Detecting malicious tweets in trending topics using a statistical analysis of language. Expert Systems with Applications, 40(8):2992 – 3000, 2013.

[20] C. Pash. The lure of naked hollywood star photos sent the internet into meltdown in new zealand. Business Insider, September 2014.

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

[22] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and evaluation of a real-time url spam filtering service. In Proceedings of the 2011 IEEE Symposium on Security and Privacy, SP '11, pages 447–462, Washington, DC, USA, 2011. IEEE Computer Society.

[23] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. Choudhary. Towards online spam filtering in social networks. In NDSS, 2012.

[24] J. Song, S. Lee, and J. Kim. Spam filtering in twitter using sender receiver relationship. In Proceedings of the 14th international conference on Recent Advances in Intrusion Detection, RAID'11, pages 301–317, Berlin, Heidelberg, 2011. Springer-Verlag.

[25] B. Settles. Active learning literature survey, University of Wisconsin, Madison, 52(55-66):11, 2010.

[26] Yang, R. Harkreader, and G. Gu. Empirical evaluation and new design for fighting evolving twitter spammers. Information Forensics and Security, IEEE Transactions on, 8(8):1280–1293, 2013.