International Journal of Web-Based Learning and Teaching Technologies
Volume 16 • Issue 4 • July-August 2021

Improvised Spam Detection in Twitter Data Using Lightweight Detectors and Classifiers

Velammal B. L., Anna University, India
Aarthy N., Anna University, India

ABSTRACT

Receiving spam messages is one of the most serious issues in social media, especially in Twitter, which is a widely used platform to reflect the opinions and emotions of an individual publicly as well as focused to a specific group of members with similar thoughts or discussion topic. In such focused discussion groups, getting spam message through social media sites is the most annoying issue. In this paper, a system is developed to detect spam tweets by using four lightweight detectors, namely blacklist domain detector, near duplicate detector, reliable ham detector, and multiclass detector. The detected tweets are then classified using ensemble classifiers such as naïve Bayes, logistic regression, and random forest. Voting method is applied to decide the labels for the tweets obtained after classification process. The proposed system has achieved an accuracy of 79% to detect spam tweets with the help of naïve Bayes classifier method and the value seems to be optimizing further with the availability of more sample data.

KEYWORDS

Blacklist, Classifier, Detector, Ensemble, Ham Tweets, Machine Learning, Spam Tweets, Twitter

1. INTRODUCTION

Nowadays social media has become the most unavoidable and the most popular means for communication amongst the individuals. For most of the youngsters, the days won't count without using social media such as Twitter, which was established in 2006 and became an exceptionally good social website amongst the most well-known microblogging administration web applications. Twitter is the most popular micro-blogging site with approximately 200 million users. Twitter has witnessed different kinds of spam attacks. Detecting a spam is the first and very crucial step in the battle of fight against spam (Chu, Widjaja, & Wang, 2012).

Conventional spam detection methods on Twitter mainly check individual tweets or twitter accounts for the existence of spam. The tweet-level detection monitors individual tweets to check whether they contain spam text content or Uniform Resource Locators (URLs). By 6th June2018, around 8.3 million tweets are generated per hour (Lin, Sun, Nepal, Zhang, Xiang &Hassan, 2017) demand near real-time delivery. So, the tweet-level detection would consume too much computing resources and can hardly meet stringent time requirements. The account-level detection works by checking individual accounts for the evidence of posting spam tweets or aggressive automation behavior. Accounts violating the twitter rules of spam and abuse (Meda, Bisio, Gastaldo, Zunino, 2014) will get suspended by the administrators. However, suspending spam accounts is an endless cat and mouse game, as it is easy for spammers to create new accounts as a replacement for the

DOI: 10.4018/IJWLTT.20210701.oa2

suspended ones. The twitter detection should shift from the perspective of individual detection to collective detection and focus on detecting spam campaigns.

A spam campaign is defined as a collection of multiple accounts controlled and manipulated by a spammer to spread spam on twitter for a specific purpose (e.g., advertising a spam site or selling counterfeit goods). Detecting such spam campaigns and prohibiting them can bring two additional benefits. First, improvement in Efficiency by clustering related spam accounts into a campaign and generating a signature for the spammer behind the campaign. With the help of this process, the system can detect multiple existing spam accounts at a given time and also capture future ones, if the spammer maintains the same spamming strategies. Second, Robustness - There are some spamming methods which cannot be detected at an individual level, similar to the behavior of posting duplicate content over multiple accounts, which Twitter do not consider as spamming. By grouping related accounts, the system can be able to detect such a collective spamming behavior and precautionary measures can be taken to restrict such messages. Another way of achieving this phenomenon is by clustering tweets with the same final URL into a campaign using the Twitter dataset and then partitioning the dataset into numerous campaigns based on URLs. Then perform a detailed analysis over the campaign data and generate a set of useful features to classify a campaign into two classes: spam or legitimate.

Internet spam is one or more unsolicited messages sent or posted as a part of larger collection of messages, all having substantially identical content (Giyanani& Desai, 2013). Most spam messages take the form of advertising or promotional materials like debt reduction plans, getting rich quick schemes, gambling opportunities, pornography, online dating, health-related products etc. The major technical disadvantages of spam messages are wastage of network resources (bandwidth), wastage of time, damage to the PC and laptops that may be caused due to viruses. Spammers generally have designed personalized templates to deliver their messages using bulk mailing software. It is widely assumed that most of the spam messages are sent directly from a collection of bots.

On an organizational front, spam effects are likely to be considered as annoyance to individual users, less reliable e-mails, loss of work productivity, misuse of network bandwidth, wastage of file server storage space and computational power. It can also include spreading of viruses, worms, Trojan horses and financial losses through phishing, Denial of Service (DoS), directory harvesting attacks. According to the Text Retrieval Conference (TREC) (Bhowmick& Hazarika, 2013) the term 'spam' is - an unsolicited, unwanted information that was sent indiscriminately. Spams are unsolicited, unratified and usually mass broadcasted to act as a carrier of unsolicited advertisements, fraud schemes, phishing messages, explicit content, promotions of cause, etc.

Spamming becomes an attractive phenomenon for spammers due to below five main reasons (Iqbal, Abid, Ahmed &Khurshid, 2016) which includes financial benefits to be earn from search engines, it sabotages the trust of end user in a search engines, spam websites serve as means of disseminating malware and adult content dissemination and also being used for fishing attacks, search engine may spend large amount of computational and storage resources on spam pages, declining employee productivity is the overwhelming by-product of spam.

It is very unfortunate that the spam keeps growing each year with a fast pace. According to different studies (chu, Widjaja, Wang 2012 & Iqbal, Abid, Ahmed,Khurshid, 2016 &Giyanani, Desai, 2013 &Bhowmick, Hazarika, 2016) the volume of spam worldwide has been increasing in all internet traffic, and is continuously increasing with each passing year. There are different techniques are being implemented by spam detection algorithms to eliminate spam and every detector is achieving different performance levels. The classification task to distinguish spam and ham is complex and constantly changing. Due to this nature of the problem; unfortunately, till date no exact solution has been explored by researchers to eliminate spam traffic. However, majority of spam detectors have two main attributes in common that determine their overall efficiency which is number of spam messages detected and number of ham (legitimate) messages falsely reported as spam.

Online social networks (OSNs) are the extremely popular collaboration and communication tools that have attracted millions of Internet users(Gao, Chen, Palsetia&Choudhary, 2012). Unfortunately,

recent evidence shows that they can also be effective mechanisms for spreading attacks. Popular OSNs are increasingly becoming the target of phishing attacks launched from large botnets (Sangeetha M, nithyanantham S&Jayanthi M,2018&Hirve S, &Kamble, 2016). Two recent studies have confirmed the existence of large-scale spam campaigns in Twitter and Facebook, respectively. Furthermore, the click through rate of OSN spam will be in the orders of magnitude higher than its counterpart (Subramaniam, T, Jalab, H. A, &Taqa, A. Y, (2012), indicating that users are more prone to trust spam messages from their friends in OSNs.

2. BACKGROUND

(Sangeetha M, Nithyanantham S &Jayanthi M, 2018) used different machine learning algorithms in the system they developed, which accurately detected the spam tweets and provided better results in TPR/FPR, Accuracy and F-measure. The different machine learning algorithms used by them are random forest and c5.0 which is a decision tree algorithm and they showed that their system is more stable than other algorithms.

(Goyal S, Chauhan R. K, &Parveen S, 2016)developed the spam detection mechanism based on Decision tree and KNN algorithm. In the proposed mechanism, they applied those algorithms on real datasets of twitter to detect spam messages. The performance metrics like TP Rate, FP Rate, Precision, Recall, F-Measure and Class are used to measure the execution of proposed mechanism.

(Hirve&Kamble, 2016) used KL algorithms to develop a system for detecting a stream of spam tweets. In order to perform this evaluation, they collected a large amount of 600 million public tweets. They have extracted 12 lightweight features which can differentiate between that of a spam and non-spam tweets from the labeled datasets. They have averaged features to machine-learning based spam classification. They sampled four different datasets to simulate various scenarios to investigate the ability of spam detection of different classifiers. They also recognized that Features discretization was an important preprocess to ML- based spam detection. Secondly, increasing training data only cannot bring more benefits to detect Twitter spam after a certain number of training samples. The classifiers can detect more spam tweets when the tweets were sampled continuously rather than randomly selected tweets. They came to the conclusion that the performance decreases due to the fact that the distribution of features changes of later days datasets, whereas the distribution of training datasets stays the same. This problem will exist in streaming spam tweets detection, as the new tweets are coming in the forms of streams, but the training dataset is not updated.

(SurendraSedhai&Aixin, 2017) propose a semi-supervised spam detection framework, named S3D. S3D utilizes four lightweight detectors to detect spam tweets on real-time basis and update the models periodically in batch mode. The experiment results demonstrate the effectiveness of semi-supervised approach in that spam detection framework experiment found that confidently labeled clusters and tweets make the system effective in capturing new spamming patterns. Tweet-level spam detection is a fine grained approach which can be used to detect spam tweets in real time. For a given tweet only limited information can be obtained. In contrast, more discriminative features can be derived from user account, historical tweets of the users, and social graph. However, by the time a malicious user is detected, the user might affect many other users. They believe that tweet-level spam detection complements user-level spam detection.

(M.Mccord&M.chuah, 2011) suggested some user-based and content-based features that can be used to distinguish between spammers and legitimate users on Twitter. These suggested features are influenced by Twitter spam policies and their observations of spammers behaviors. So they use these features to help identify spammers. They evaluate the usefulness of these features in spammer detection using traditional classifiers like Random Forest, Naïve Bayesian, Support Vector Machine, KNN neighbor schemes using the Twitter dataset they have collected. Hence results show that the Random Forest classifier gives the best performance. Using that classifier, suggested features have achieved 95.7% precision and 95.7% F-measure. Based on dataset, features provide slightly better

classification results when compared to those suggested. They evaluated detection scheme using larger Twitter datasets along with wall-post datasets from other online networking sites like Facebook.

(Eshraqi, N, Jalali, M, &Moattar M. H,2015) suggested an approach which can identify 89% of available spam tweets. Although this approach can't identify 11% of spam tweets, but because of high maximum precision and having zero of FPR, they result all determined tweets as spam are certainly spam and any normal tweet is considered as spam wrongly. High accuracy of this approach shows tweets are identified with good quality. According to homogeneity and Purity values, they result each cluster contains data from the same class without impurities into clusters and tweets are placed into related clusters. Also by paying attention to amount of homogeneity and purity they can conclude that each cluster contains data from the same class and impurity is not seen in these clusters. Low SSQ suggests data item's distance from cluster centers is slightly low, each cluster data are centralized around the core and dispersion is low. So they conclude the cluster process has been done with high quality.

(B.L. Velammal, 2019) implemented knowledge based Twitter Sentiment Analysis Classifier. It can generate tweets for a given topic, classify them according to positive, negative and neutral sentiments and provide the results graphically on a webpage. Since this model has been developed using pure machine learning techniques, it classifies tweets based on the frequency of the words and not its semantics. Natural Language Processing can be incorporated into the model to better detect sentiment and sarcasm. The tweets have been classified into positive, negative and neutral and the polarity has been calculated but the actual emotional category such as elation, joy, happiness, dull, sad, depressed etc. have not been identified. Larger knowledge datasets have been used to train the system to provide better accuracy. So they have trained the system using two datasets of approximately 15,000 tweets each. The accuracy of the system has shown improvement in relation to training data set.

(Awad W. A. &ELseuofi, 2011) reviewed some of the most popular machine learning methods and of their applicability to the problem of spam e-mail classification. Descriptions of the algorithms are presented and the comparison of their performance on the Spam Assassin spam corpus revealed very promising results especially in the algorithms that is not popular in the commercial e-mail filtering packages. Spam recall percentage in the six methods has less value among the precision and the accuracy values, while in term of accuracy they show that the Naïve Bayes and rough sets methods have a very satisfying performance among the other methods. Finally hybrid systems look to be the most efficient way to generate a successful anti-spam filter nowadays.

(Lin, G, Sun, N, Nepal, Zhang J, Xiang Y & Hassan H, 2017) used the Naïve Bayesian Classifier and extracting the word using word-count algorithm. After calculation they find that Naïve Bayesian classifier has more accurate the support vector machine. The error rate is very low when they are using the Naïve Bayesian Classifier. So they can say that Naïve Bayesian Classifier produce better result than Support Vector Machine.

(Subramaniam, Jalab&Taqa, 2012) explores the use of Naïve Bayesian technique to combat spam problem for Malay language. An experiment was conducted by them using Naïve Bayesian technique in filtering Malay language spam shows promising results. Further refinement on the stemming and feature selection process needed to improve accuracy rate. In order to avoid over-fitting and biasness training corpus need to be increased. However, the proposed solution was considered only for Malay language and not any other language as well as standard dataset.

(Meda, Bisio, Gastaldo& Zunino, 2014) points out the application of three machine learning algorithms, studying the different performance of these techniques, in order to identify the best algorithm and the best parameters that combine both satisfactory detection results and considerable performance capabilities. Experimental results confirm the effectiveness of Random Forest algorithm compared to the Support Vector Machine and the Extreme Learning Machines: the Random Forest performances increase with the decreasing number of features opposed to the other two techniques. This behavior underlines the advantage to choose few features on behalf of detection and computational cost.

3. EXPERIMENTAL SETUP

In this paper, spam tweets are detected using various classifiers namely Naïve Bayes, logistic regression, and random forest. The whole experiment comprised of two parts. First, preprocess the tweets according to the detectors. Secondly, apply the tweets to the respective detectors for classification between two classes. The two classes are spam and ham, which are the main focus of our work. Here four detectors including three lightweight detectors and three classifiers are implemented. Thus the system can produce the desirable level of accuracy in the classification of tweets.

4. SYSTEM ARCHITECTURE

The architecture of the proposed system has been explained clearly in Figure 1 and the detailed functions of the modules present in the system have been explained below.

5. DATA PREPROCESSING

Pre-processing is considered as an important step in text mining. There are three steps in Pre-processing task for tweet classification, which are tokenization, stop word removal and stemming. First step is tokenization process, in which all symbols, punctuations and numbers will be removed. The remaining strings will be split up into tokens. Second step is stop word removal. Many of the frequently used words in English are useless in Information Retrieval (IR) and text mining process. These words are called 'Stop words', which are language-specific functional words and frequent words that carry no information (i.e., pronouns, prepositions, conjunctions). In this step, the common words, which are the most frequent words that exist in a document are removed. In English language, there are about 400-500 Stop words and this list is based on word frequency. This process will identify which words those match with the stop word lists by comparing both of them. Removing these words will save spaces for storing document contents and reduce time taken during the searching process. Third step is Stemming, which means finding the origin of the words and removing prefixes and postfixes. By using Stemming, forms of a word like adjectives, nouns and verbs are converted to homological like word. For instance, both 'capturing' and 'captured' are converted to a same word – 'capture'.

5.1 Blacklist Domain Detector

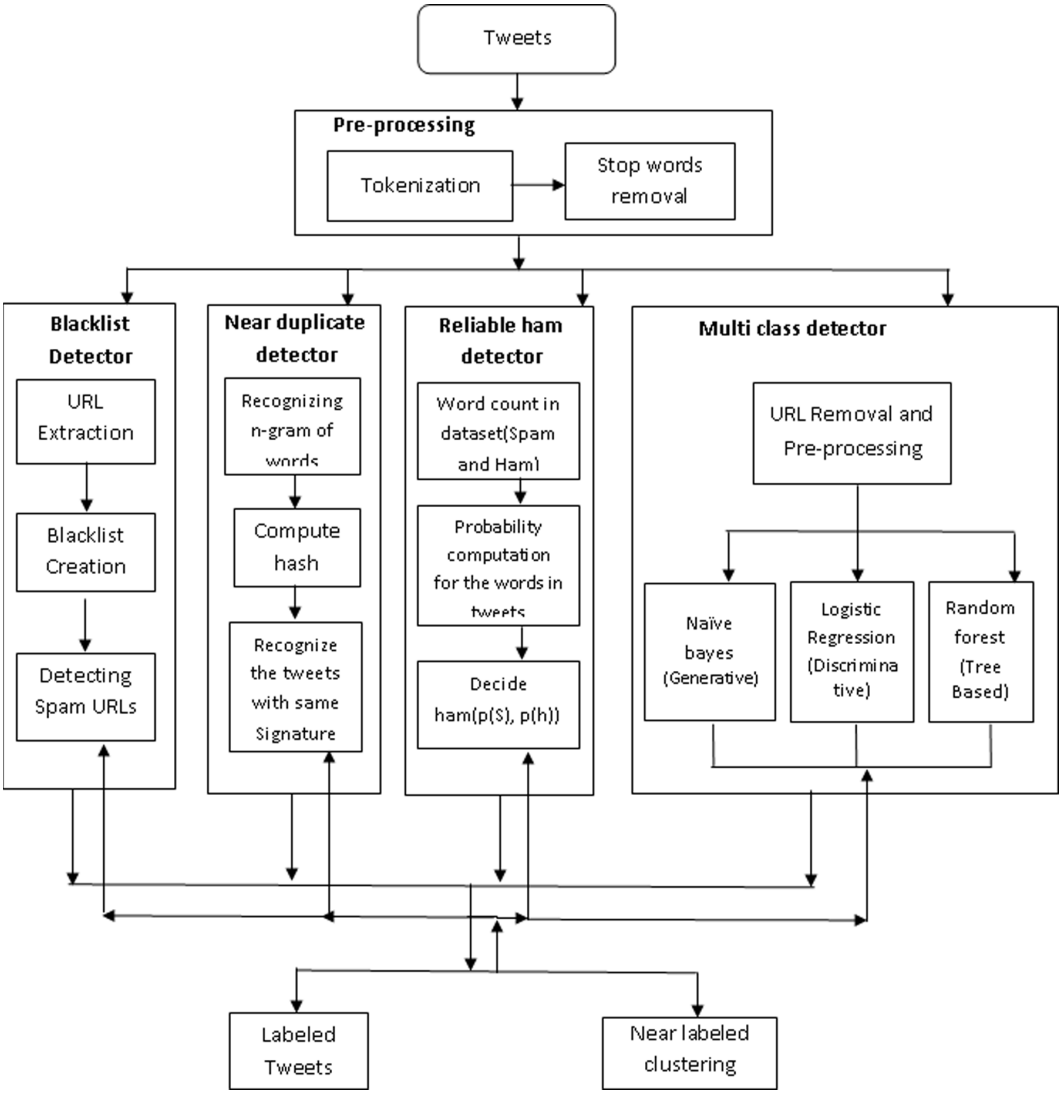
Spam blacklist is a list of domain names that supposedly are source of spam. If your domain name in the spam list, tweets that received will be rejected or marked as spam. The blacklist domains are listed from the tweets contained in the dataset. The content pollutant list contains spam tweets. The URL from those tweets is taken and added to blacklist domain list in order to train the model. If the test tweets are received by the model it will tested against the training data. Thus based on the spam blacklist URL the tweets are classified as spam accordingly. In general, microblog such as twitter has the domain url as short which is shortened url (eg: bit.ly) and it will extend to some domains such as facebook, youtube, etc., such shortened url are taken as spam blacklist which is obtained content pollutant tweets.

5.2 Near Duplicate Detector

The Near Duplicate Detector aims at finding the hash value of Tweets using methods like N-Gram modeling and cluster similar tweets, thereby a set of spam tweets can be clustered together and can be segregated separately from desirable ham tweets.

N gram modeling: An N-gram is an N-character slice of a longer string. Although ideally the term implies that notion of any co-occurring set of characters in a string can be included, but in this paper shall refer the term for contiguous but overlapping slices only. N-grams of several different lengths simultaneously are used in the processing while detecting spam tweets. Even blanks are appended to

Figure 1. System Architecture



the beginning and ending of the string to support pattern matching efficiently. Let us replace blank spaces by the underscore character ("_") for the n grams of the word "STAY". Bi-grams: _S, ST, TA, SY, Y_ Tri-grams: _ _S, _ST, STA, TAY, AY_, Y_, Quad-grams: _ _ _ S, _ _ ST, _STA, STAY, TAY_, AY_ _, Y_ _ _. This approach is extremely beneficial as it allows the use of the same word or phrase in different context when their meanings vary substantially. Thus the tweets are converted as unigram, bigram, trigram of words; followed by application of Min hash algorithm to the n-gram of the tweets, which is a technique for approximating the Jaccard similarity between two different sets.

MinHash makes use of "random hash functions". A random hash function takes, e.g., a 32-bit integer and maps it to a different integer, with no collisions. Put another way, if you took the numbers 0 – (2^32 – 1) and applied this hash function to all of them, you'd get back a list of the same numbers in random order.

The hash functions have the following form:

$h(x) = (ax+b) \% c$

The coefficients 'a' and 'b' are randomly chosen integers less than the maximum value of 'x'. 'c' is a prime number slightly bigger than the maximum value of 'x'. The hash value will generate for all n-gram of words. Then the threshold value will compute based on the training set. By comparing the threshold group the hash values which are nearly close to each other. If two or more tweets have the same signature means the hash values, then the tweets are considered near-duplicates. If a cluster of near-duplicate tweets hashed to the same signature have been labeled as spam or ham tweets, then the new tweet having the same signature receives the same label.

6. RELIABLE HAM DETECTOR

Probability is based on the observations of certain events and it is the ratio of number of observations of the event to the total numbers of observations. An experiment is a situation involving chance or probability that leads to results called outcomes. An outcome is the result of a single trial of an experiment. The probability of an event is the measure of the chance that the event will occur as a result of an experiment. Probability of an event A is symbolized by P(A). Probability of an event A is lies between $0 \leq P(A) \leq 1$. Probability of an Event = Number of Favorable Outcomes / Total Number of Possible Outcomes. Probability of words in each tweets will calculated. If the word is in spam or ham tweets that probability will propagated. Ham means which cannot be spam anymore. Hence there is no chance for a ham tweets to be a spam. Tweets will considered ham if the users did not post any more than 5 spam tweets. Similar to this criterion, many such need to be evolved over period of time to detect the ham accurately. By constantly updating the spam tweet patterns, the probability of detecting ham will increase.

7. MULTICLASS DETECTOR

7.1 Naivebayes

Navïe Bayes filtering technique is a widely used machine learning method. Bayesian filtering is based on the principle that most events are dependent and that the probability of an event occurring in the future can be guessed from the previous occurrences of that event. In Bayesian net (Bayesian filtering), it assigns a score to different tendencies used by spammers. For example, a message with a high percentage of misspelled words sent from a Russian IP address that mentions Viagra (or spelling variations) has more tendencies used by spammers than a message regarding your annual sales forecast. Seeing that the first message fits a specific pattern, that message would be blocked if the score meets the threshold set by the administrator If the word "cheap" occurs in 500 out of 3,500 messages and in 5 out of 300 legitimate messages, for example, then its spam probability would be 0.8955 (that is, [500/3500] divided by [5/300 + 500/35000]).

$Pr(spam/words) = Pr(word/spam) / (Pr(words/ham) + Pr(words \mid spam))$

The updating frequency mechanism of database file for Bayesian filter makes it secure to end users. This spam data file must hold large sample size of known spam and must be constantly updated with the latest spam by the anti-spam software. This will assure that the Bayesian filter is aware of the most recent spam tricks, and ultimately resulting in a high spam detection rate. General Algorithm by using statistical content to filter spam traffic contains the following steps: Following are some important reasons to choose Bayesian Filters to detect spam traffic:

Bayesian filtering apply intelligent approach to filter data because it examines all aspects of a message, as opposed to keyword checking that classifies a mail as spam on the basis of a single word.

A Bayesian filter is constantly self-adapting - By learning from new spam and new valid outbound mails, the Bayesian filter evolves and adapts to new spam techniques. The Bayesian method is multi-lingual and international. Combination of training and classification or testing is described in the following algorithm. Create spam and ham sets by collecting many tweets, retrieving individual tokens strings as feature words and calculating the appearance time of the token to build the feature set $f = \{w1, w2, \ldots, wn\}$, Generate hash tables for both ham and spam for the mapping relation of a feature word tokens, And Compute the class-conditional probability $P(w_t | c_i)$ for feature word w_t

```
Algorithm: Pseudocode for calculating conditional probability
For each word in t do
     Find ngram set
          Calculate appearance of tokens as f
               Generate hash tables for f
     Compute p(w|c)
End for
Classification Phase of the Bayesian method hold the following
probability calculation:
Step 1: Retrieve feature words from new tweets.
Step 2: Calculate the probability P(c(ham)|d) of legitimate
message and P(c(spam)|d) of spam when it satisfies the extracted
feature words d.
Step 3: Classify the incoming tweets based on the results.
When the value of P(cspam |d) is greater than P(cham|d) or the
threshold value λ, this e-mail is tagged as spam.

Algorithm : Naïve bayes

For f in t do
 Calculate p(c(ham)|d) and p(c(spam)|d)
 If p(c(spam)|d) > p(c(ham)|d)
The tweet is spam
Else
The tweet is ham
End for
```

7.2 Random Forest

Random forests are a combination of tree predictors so that all trees depend on the values of a random vector sampled autonomously and with the similar distribution for all trees in the forest. The random forests algorithm for prediction or classification task can be explained as follows:

1. Using original samples data draw n tree bootstrap
2. For every of the bootstrap samples, produce an unpruned classification tree, by following modification: at each node, instead of choosing the best split among all predictors, arbitrarily sample m try of the predictors and select the best split among those variables.
3. Predict new data by aggregating the predictions of the n-tree trees using majority votes for classification.

An estimation of the error rate can be found, based on the training data, by the following steps:

1. At every bootstrap iteration, predict the data not in the bootstrap sample (what Breiman calls "out-of bag", or OOB, data) by considering the tree developed with the bootstrap sample.
2. Cumulate the OOB predictions. (On the average, every data point would be out-of-bag around 36% of the times, so cumulate these predictions.) Calculate the error rate, and call it the OOB estimate of error rate

The Random forest is a meta-learner which consists of many individual trees. Each tree votes on an overall classification for the given set of data and the random forest algorithm chooses the individual classification with the most votes. Each decision tree is built from a random subset of the training dataset, using what is called replacement, in performing this sampling. That is, some entities will be included more than once in the sample, and others won't appear at all. In building each decision tree, a model based on a different random subset of the training dataset and a random subset of the available variables is used to choose how best to partition the dataset at each node. Each decision tree is built to its maximum size, with no pruning performed. Together, the resulting decision tree models of the Random forest represent the final ensemble model where each decision tree votes for the result and the majority wins. Each tree is constructed using the following algorithm: Let the number of training cases be N, and the number of variables in the classifier be M. The number of m input variables to be used to determine the decision at a node of the tree; m should be much less than M. Choose 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. 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. Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier).

```
Algorithm : random forest
Initialize N, m and M
If(m<M)
For i=0;i<=n
Choose Y train[i]
     For each node in T
          Choose root(best split)
          Grow tree
     End for
End for
```

7.3 Logistic Regression

An explanation of logistic regression begins with an explanation of the logistic function (also called the sigmoid function):

$f(z) = 1/1 + e^{-z}$.

The logistic function is useful because it can take as an input, any value from negative infinity to positive infinity, whereas the output is confined to values between 0 where The variable represents the exposure to some set of risk factors, while f(z) represents the probability of a particular outcome, given that set of risk factors. The variable z is a measure of the total contribution of all the risk factors used in the model. The variable z is usually defined as:

$z = \sum w_i x_i$,

where x1..xn are the features and w1..wn are the regression coefficients (weights).

The Logistic Regression algorithm is as given below:

1. Initialize weight vector to zero
2. Train the features by minimizing the logistic loss

while (||gradient||1> precision) do
calculate the new prediction y
vector using:

$Y=1/1+e-W.X$

3. Calculate the gradient vector using gradient and calculate the logistic loss on the test.

```
Algorithm Logistic regression:
Wt=0
While(||gradient||1>precision)
Do
Calculate the new prediction y vector
Calculate gradient vector and logistic loss
End while
```

Logistic Regression is another way to determine a class label, depending on the features. Logistic regression takes features that can be continuous (for example, the count of words in a tweet) and translate them to discrete values (spam or not spam). A logistic regression classifier works in the following way:

1.Fit a linear model to the feature space determined by the training data. This requires finding the best parameters to fit the training data
2.Using the parameters found in step 1, determine the z value for a testing point.
3. Map this z value of the testing point to the range 0 to 1 using the logistic function. This value is one way of determining the probability that these features are associated with a spam.

An example of a linear model has been provided in Figure 2 and the feature space has been depicted for better understanding. Image credit: https://en.wikipedia.org/wiki/File:Linear_regression.svg

The logistic function has been provided in Figure 3. Logistic Curve. Image credit: https://en.wikipedia.org/wiki/File:Logistic-curve.svg

$P(spam|z)= 1/(1+e^{-z})$
$P(Ham|z) = 1-p(Spam|z) = e^{-z}/(1+e^{-z})$

8. EXPERIMENTS AND RESULTS

To evaluate the proposed system the social dataset Honeypot (http://infolab.tamu.edu/data/) is used, which is freely available in the internet. The dataset contains legitimate tweets and content pollutant tweets. The dataset contains 22,223 content polluters, 2 million tweets, and 19,276 legitimate users, and 3 million tweets. The dataset size is 715 megabytes. It has been used separately for testing and training. Among 100 percent of these tweets 20 percent tweets used for training and 80 percent

Figure 2. Linear Regression

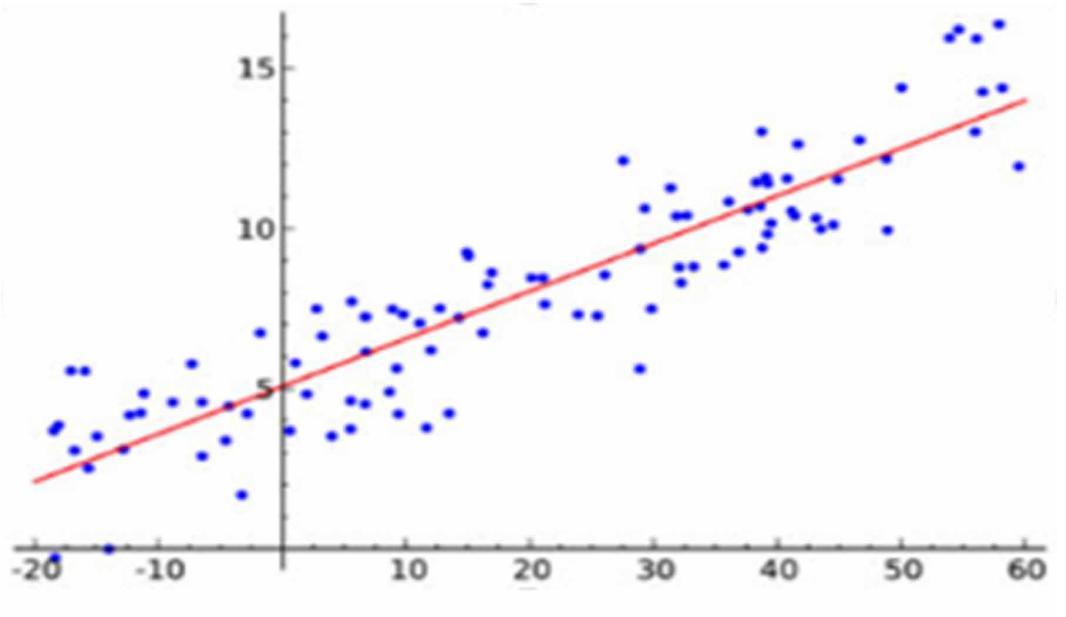
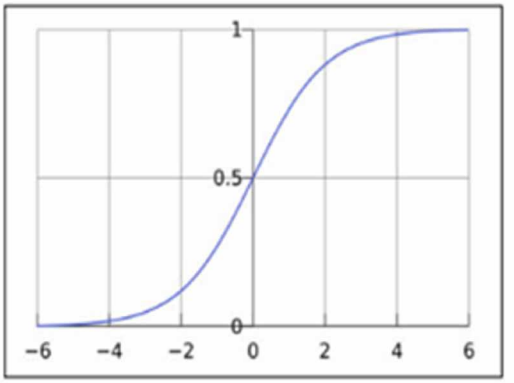


Figure 3. Logistic Curve



for testing. The evaluation metrics considered to evaluate the performance of the application are Precision, Recall and F1 Score. These metrics help to perform comparative performance analysis of various detectors and classifiers used in the proposed system. The details of metrics obtained from the proposed system have been detailed in Table 1. Key description of evaluation metrics.

True Positives (TP): These are the correctly predicted positive values, which means that the value of actual class is yes and the value of predicted class is also yes.

True Negatives (TN): These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. False positives and false negatives, these values occur when your actual class contradicts with the predicted class.

False Positives (FP): When actual class is no and predicted class is yes.
False Negatives (FN): When actual class is yes but predicted class in no.
Accuracy: Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. If there is high accuracy then the model is considered to be the best and provide better expected result. This measure can be used only when there is symmetric datasets where values of false positive and false negatives are almost same. The proposed model has been evaluated with various classification methods like Naïve Bayes, Logistic regression, Random forest and achieved the highest accuracy of 79% using Naïve Bayes method for the dataset under consideration.

Accuracy = TP+TN/TP+FP+FN+TN

Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision relates to the low false positive rate. Here this spam detection system has got nearly 0.95 precision which is pretty good.

Precision measures the percentage of tweets flagged as spam that were correctly classified

Precision = TP/TP+FP

Recall: Recall is the ratio of correctly predicted positive observations to all positive observations in actual class. The proposed system has got a recall of 0.901 which is good for this model.

Recall measures the percentage of actual spam tweets that were correctly classified.

Recall = TP/TP+FN

F1 score: F1 Score is the weighted average score of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall. In our case, F1 score is 0.701.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

8.1 Accuracy Graph For The Classifiers

Naïve Bayes gives better accuracy than Random forest and Logistic regression. But Logistic regression is more or less equal to Naïve Bayes. Random forest has some less value comparatively. The results have been depicted in Figure 4. Accuracy Result and the values are detailed in Table 3. Evaluation result of classifiers. Table 2 depicted that the result of four detector. In that table which is shown that the 60percent of tweets among 100 percent are classified by the multiclass detector, 10 percent tweets which having domains are classified by blacklist domain detector, 10 percent tweets classified the reliable ham detector which is capable to produce the tweets which is not possibly to be a spam tweets. Near duplicate detector groups the possible correlated tweets as both nearly spam and ham

Table 1. Key description of evaluation metrics

Evaluation Metric	Details of the Metric
TP (True Positive)	Tweets that are correctly predicted as spam
TN (True Negative)	Non spam Tweets that are correctly predicted as Legitimate ham tweets
FP (False Positive)	Non spam tweets that are misclassified as spam
FN (False Negative)	Spam tweets that are misclassified as Legitimate ham tweets

Table 2. Classification results of Tweets

No	Detector	Count	Classification of tweets
1.	**Blacklist Domain Detector**	**0.10**	**0.10**
2.	**Near Duplicate Detector**	**0.20**	**0.189**
3.	**Reliable ham Detector**	**0.10**	**0.079**
4.	**Multiclass Detector**	**0.60**	**0.51**

tweets. Thus the light weight detectors classify the tweets lightly which is used to reduce the fully dependence of multiclass detector.

Figure 4 depicted the accuracy values for all four classifiers. Among these four classifiers naïve bayes produce the best result among four.

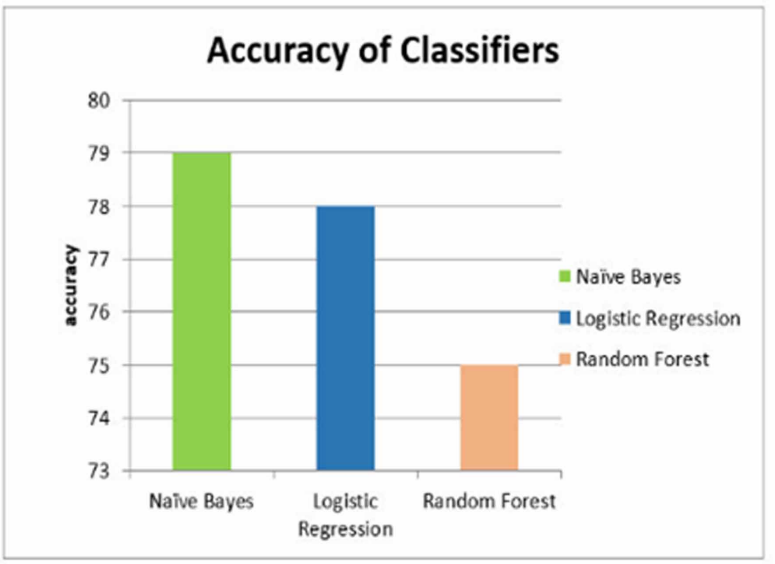
Here detectors utilize classification techniques at two levels, tweet level and cluster level. Two light weight detectors works based on tweet level characteristics. Here, a cluster is a group of tweets with similar characteristics. With this flexible design of the system, any features that maybe effective in spam detection can be easily incorporated into the detection framework. The framework starts with a small set of labeled samples.

Thus lightweight detectors making the framework more robust in identifying spam tweets. Tweets that are labeled by the first three detectors (i.e., blacklisted domain, near-duplicate, and reliable ham tweet) are considered as confidently labeled tweets. For the classifier based detector, we use three classifiers each is based on a different classification technique. Tweets that are labeled as spam by all the three classifiers are considered as confidently labeled spam tweets. The words cannot contain domain names could not detected by blacklist domain detector. The near-duplicate detector computes a signature for each tweet to check if the tweet is a near duplicate of a labeled cluster. If the signature of a tweet does not match any pre labeled cluster, then the tweet is passed to the next level detectors. Similarly, tweets that do not contain any spam words are labeled as ham by reliable ham detector. After all the tweets that do not match prelabeled clusters but having the same signature are obtained, then they are grouped into a new cluster. Specifically, if there are more spam tweets in a cluster than ham tweets, then the cluster is labeled as a spam cluster. However, the majority voting approach solely relies on the predicting power of the detectors Tweets that are not labeled in any of

Table 3. Evaluation result of classifiers

Classifiers	Accuracy
Naïve Bayes	0.79
Logistic regression	0.78
Random Forest	0.75

Figure 4. Accuracy Result



the previous steps are processed and labeled in this step. Here, we develop a spam detector by using three efficient classifiers, namely, Naïve Bayes (NB), logistic regression (LR), and random forest (RF). The three classifiers use different classification techniques, i.e., generative, discriminative, and decision tree-based classification models. Based on the respective technique the classifiers predict the label. By following the majority voting method at least two detectors label is set as final label with the best accuracy. Even the classifiers do their best the most important thing in this work is deploying three light weight detectors.

The tweets are classified as spam and tweets by the detectors as in the screenshots:

Figure 5 represents spam and ham tweets based on the domains which are all blacklisted.

The hash values generated for the tweets it is represented by Figure 6.

Figure 7 represents the threshold calculation to compare the MinHash values.

The ham tweets which are all nearly close to each other is represented by Figure 8.

Figure 9 represents probability values calculated for the words the in the dataset.

The reliable ham tweets which cannot be a spam anymore is represented by Figure 9

The tweets labelled by the naïve bayes classifier which has the highest accuracy is shown in the Figure 11

Figure 12 represents the tweets labeled by the logistic regression classifier.

The tweets labeled by the Random Forest classifier is demonstrated by the Figure 13 .

The tweets are labeled by the voting method which takes the maximum values which are classified by the classifier is shown in the Figure 14.

9. CONCLUSION

In the recent technology world, Twitter is one of the important social media Application, which is used to unite people across regions and provide a common platform to share their personal opinions, thoughts, feelings and much more for a common as well as greater cause. In such a significant medium of communication, it is important that spam mails or comments do not obstruct the users and make

Figure 5. Blacklist Domain Detector

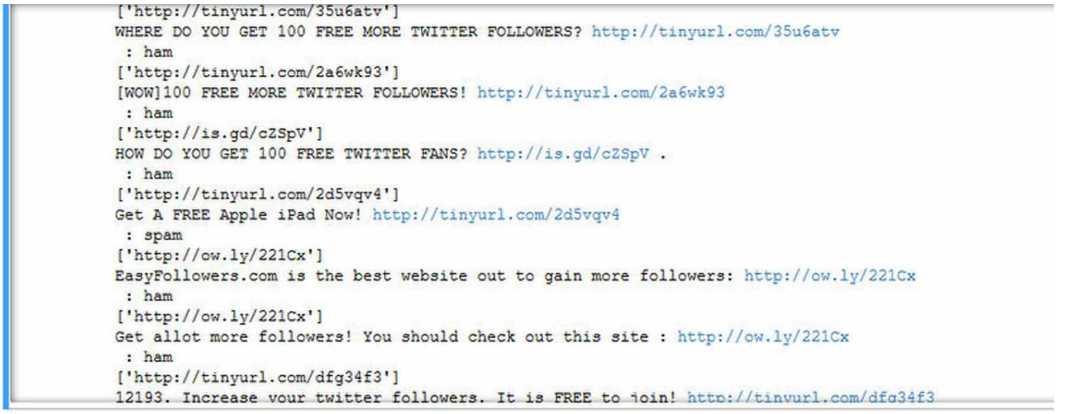


Figure 6. Generated Hash Values

them deviate from the greater cause, as this reduces the primary intention of the application and reduces the efficiency of the infrastructure. In this paper, the spam and ham tweets are labeled using four detectors and the classifiers namely - Naïve Bayes, random forest and Logistic regression. The light weight detectors used here does a weightless detection and the classifiers detect the tweets effectively. As the analysis and results section has revealed, looking at all the parameters collectively, it is found that Naïve Bayes is the best classifier of this kind of operation and gives more accuracy than other two classifiers. F-measure, which represents the accuracy, is the highest for Naïve Bayes and it also has the highest precision score. It will help in classifying the spam and ham tweets respectively. On the other hand, it is also seen that ensemble of classifiers technique helps in learning capability of the classifiers. The time taken to build the model for the best classifier found for this study, which is Naïve Bayes, is also less. However, as already established classifiers used, the light weight detectors did their part. Hence the results are more effective and easy to train the classifiers as well.

Figure 7. Threshold Calculation

```
with open('C:/Users/rthyn/Desktop/project/content_polluters_tweets.txt', 'r',encoding="ISO-8859-1") as f:
    #reader = csv.reader(f)
    lines = f.readlines()
    s=[]
    p=[]
    q=[]
    #print(lines)
    for i in lines[0:100]:
        hsh=minhash_tweet(line1)
        res=jaccard_sim(i,hsh)
        q.append(res)
        #print(q)
        thr=0.05
        if res>thr:
                s.append(i)
        else:
                p.append(i)

a3=min(q)
a4=max(q)
print(a4)
print(a3)
#print(q)
```

```
0.8984375
0.15625
```

Figure 8. Near Duplicate Domain Detector

```
0.03125
0.0703125
```

```
[7]: p
```

```
 'Car needs juice :-) (@ Shell breukelen A2 in Breukelen) http://bit.ly/4mt12X\n',
 '"Foursquare punten vandaag voor Marktplaats, Dauphine (Measure Works), Boondoggle en Tribal DDB (IAB meeting) :-)"\n',
 'Eerste project document voor vandaag alweer verzonden. Nu snel nog wat email voordat de dag van meetings begint!\n',
 '@erikvanderkooij leuk http://nl.wikipedia.org/wiki/Nauru was ooit rijkste per hoofd van de bevolking. Nu heeft 90% over
gewicht....\n',
 '@heinivanbergen :-) ik hoor overigens leuke ontwikkelingen. Ben benieuwd :-)\n',
 'Ziet dat je jezelf nog steeds kunt verzekeren via de DSB bank website http://bit.ly/LwgeG\n',
 '@nouwen lees nu overigens pas dat je de PIBN man bent geworden. Congrats nog!\n',
 '@nouwen html / css development veel geld? Er zijn veel Niet US / EU toppers die dat voor 3 appels en 2 eieren uitvoeren
... Eigen site?\n',
 '"@nouwen niet aan beginnen, uitbesteden :-)"\n',
 '@oli4b veel plezier daar anyways :-)\n',
 "@oli4b ah over service via internet. Als ze nog tips nodig hebben hoe ze hun eigen webpresence (en website) kunnen verb
eteren.. I'm here :-)\n",
 '@oli4b wat voor een event ben je vandaag bij?\n',
 'Ziet veel nieuwe #foursquare requests binnenkomen. Geweldig hoe goed lokale promotie bij je doelgroep werkt :-)\n',
 'sprak net roze wolk @barts :-)\n',
 '"@NuNicole mooi, welkom terug :-)"\n',
 'even bellen met @vangeest\n',
 '@Tim V de bestuurder kan echter niet parkeren :-)\n',
```

```
[8]: s
```

```
t[8]: ['"@SunnySanny ze zoeken 23 nieuwe mensen, dus nog niet alles is bezet :-)"\n',
 '"I\'m at Parnassia (Zee, Bloemendaal). http://bit.ly/6BRyw"\n',
 '"Kreeg vandaag complete web monitoring cadeau van #measureworks , thanks! http://twitti.ms/eYwl"\n',
 '@NuNicole we houden je in de gaten :-)\n',
 'Optimize photos for your digital scrapbook http://bit.ly/8n5dSq\n']
```

Figure 9. Probability for words in Ham and Spam Tweets

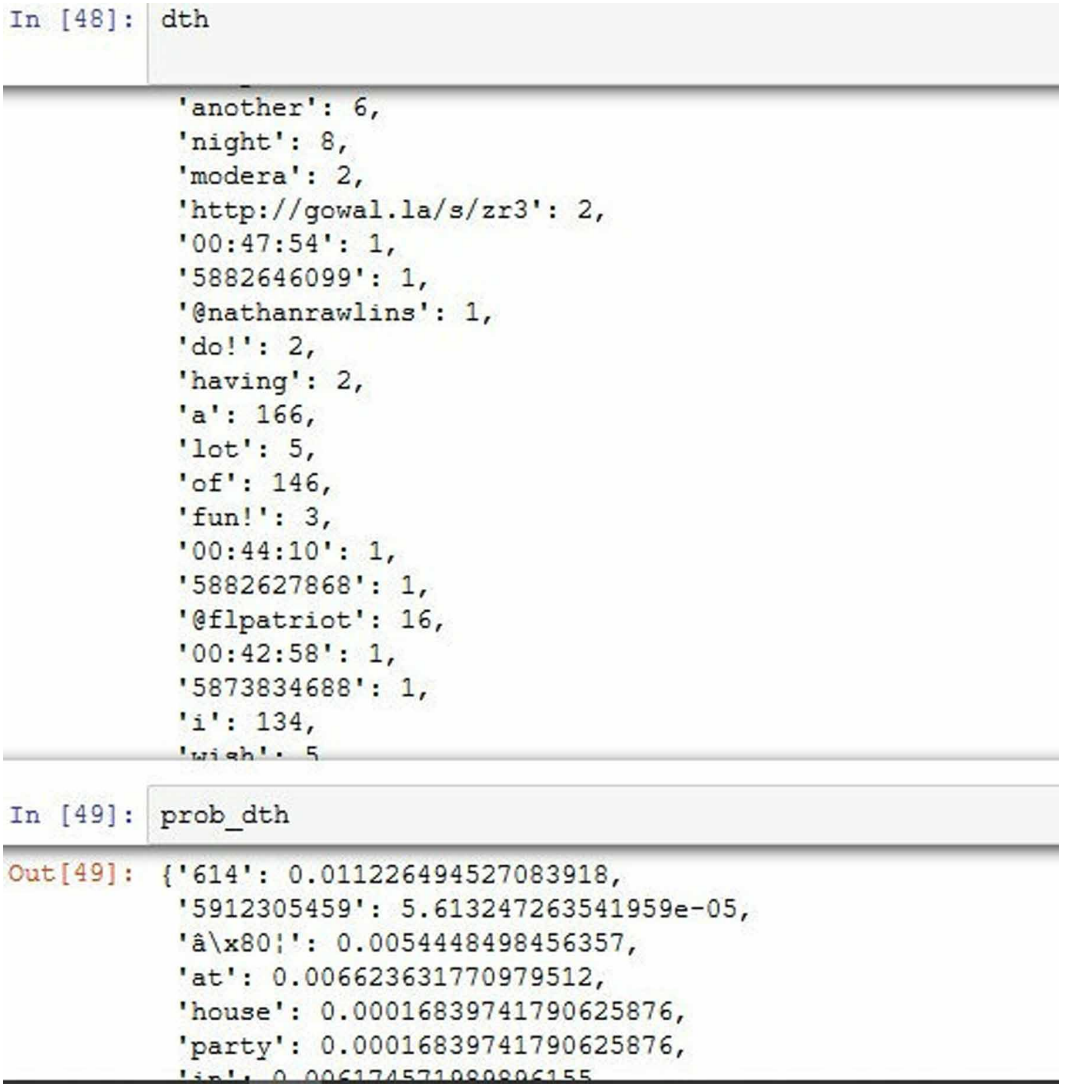


Figure 10.Reliable Ham Tweets

```
spam_pro,ham_pro=probability_compute(prob_dth,prob_dt,"hi This please follow me for more news")
if spam_pro>ham_pro:
    print("spam")
else:
    print("ham")
```

```
ham
```

```
with open('C:/Users/rthyn/Desktop/test6.csv', 'r',encoding="ISO-8859-1") as f:
    #eader = csv.reader(f)
    lines = f.readlines()
    #print(lines)
    for i in lines[0:100]:
        #print(i)
        spam_pro,ham_pro=probability_compute(prob_dth,prob_dt,i)
        if spam_pro>ham_pro:
            print("spam : \t",i)
        else:
            print("ham : ",i)
```

```
ham :  proudly brought to you by green fairy absinth looking for a great xmas present we have created a special xmas

spam :   enjoying the nd last day of my s

ham :  meilinmiranda right with that advice i shall go forth

ham :  hi all thanks for your enquiries regarding our regular events we have finished for the year but always able to

ham :  having the best birthday

ham :  please feel free to connect with us on linkedin

spam :   derterrorist thanks for sharing
```

Figure 11 Tweets Classified by Naïve Bayes

```
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['ham']
['spam']
['spam']
['spam']
```

Figure 12 Tweets classified by Random Forest

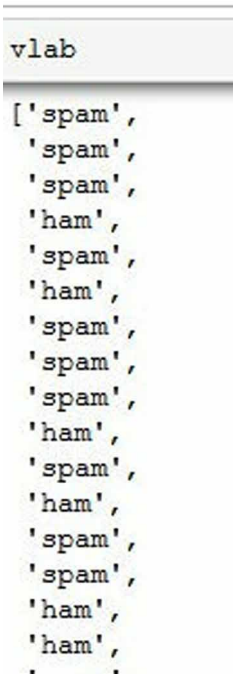
```
In [13]: twt = pd.read_csv("C:/Users/rthyn/Desktop/lost1.csv", engine="python")
         #print(twt)
         for i,j in twt.iterrows():
             #print(j[0])
             lg=logistic.predict(count_vect.transform([j[0]]))
         #print(i)

             print(lg)
         n=len(twt)
         print(n)
```

```
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['spam']
['ham']
```

Figure 13.Tweets classified by Logistic regression

```
['ham']
['ham']
['ham']
['ham']
['spam']
['ham']
['spam']
['spam']
['ham']
['spam']
['ham']
['spam']
['ham']
['ham']
['ham']
['spam']
['ham']
['ham']
['ham']
```

Figure 14. Voting method



REFERENCES

Awad, W. A. (2011). Machine learning methods for spam e-mail classification. *International Journal of Computer Science and Information Technologies*, 3(1), 173–184. doi:10.5121/ijcsit.2011.3112

Bhowmick, A., & Hazarika, S. M. (2016). *Machine learning for e-mail spam filtering: review, techniques and trends.* arXiv preprint arXiv:1606.01042.

Chu, Z., Widjaja, I., & Wang, H. (2012, June). Detecting social spam campaigns on twitter. In *International Conference on Applied Cryptography and Network Security* (pp. 455-472). Springer. doi:10.1007/978-3-642-31284-7_27

Eshraqi, N., Jalali, M., & Moattar, M. H. (2015, November). Detecting spam tweets in Twitter using a data stream clustering algorithm. In *2015 International Congress on Technology, Communication and Knowledge (ICTCK)* (pp. 347-351). IEEE. doi:10.1109/ICTCK.2015.7582694

Gao, H., Chen, Y., Lee, K., Palsetia, D., & Choudhary, A. N. (2012, February). Towards Online Spam Filtering in Social Networks. In NDSS (Vol. 12, No. 2012, pp. 1-16). Academic Press.

Giyanani, R., & Desai, M. (2013). Spam detection using natural language processing. *Int. J. Comput. Sci. Res. Technol*, *1*, 55–58.

Goyal, S., Chauhan, R. K., & Parveen, S. (2016, December). Spam detection using KNN and decision tree mechanism in social network. In *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)* (pp. 522-526). IEEE. doi:10.1109/PDGC.2016.7913250

Hirve, S., & Kamble, S. (2016). Twitter spam detection. *International Journal of Engineering Science*, 2807.

Iqbal, M., Abid, M. M., Ahmad, M., & Khurshid, F. (2016). Study on the effectiveness of spam detection technologies. *IJ Information Technology and Computer Science*, *1*(1), 11–21. doi:10.5815/ijitcs.2016.01.02

Lin, G., Sun, N., Nepal, S., Zhang, J., Xiang, Y., & Hassan, H. (2017). Statistical twitter spam detection demystified: Performance, stability and scalability. *IEEE Access : Practical Innovations, Open Solutions*, *5*, 11142–11154. doi:10.1109/ACCESS.2017.2710540

Meda, C., Bisio, F., Gastaldo, P., & Zunino, R. (2014). Machine learning techniques applied to Twitter spammers detection. *International Carnahan Conference on Security Technology*.

Mccord, M., & Chuah, M. (2011, September). Spam detection on twitter using traditional classifiers. In International conference on Autonomic and trusted computing (pp. 175-186). Springer. doi:10.1007/978-3-642-23496-5_13

Sangeetha, M., Nithyanantham, S., & Jayanthi, M. (2017). Comparison of twitter spam detection using various machine learning algorithms. *International Journal of Engineering & Technology, 7*(1.3), 61-65.

Sedhai, S., & Sun, A. (2018). Semi-supervised spam detection in Twitter stream. *IEEE Transactions on Computational Social Systems*, *5*(1), 169–175. doi:10.1109/TCSS.2017.2773581

Subramaniam, T., Jalab, H. A., & Taqa, A. Y. (2012). Naïve bayesian anti-spam filtering technique for malay language. *International Conference on Computer Engineering & Mathematical Sciences*.

Velammal, B. L. (2019). Development of knowledge based sentiment analysis system using lexicon approach on twitter data. *International Journal of Knowledge Management Studies*, *10*(1), 58–68. doi:10.1504/IJKMS.2019.097125

B. L. Velammal works as an Assistant Professor in Anna University. Her major field of study includes content adaptation, deep learning, multimedia databases, image processing and content analysis. She has published several research articles in various international journals.

Aarthy N. is studying in Anna University. Area of interest includes text mining, machine learning.