Spam detection framework in twitter using Machine Learning

Swathi Darla1, Gargi N2, Archana N3
[1,2,3]*Assistant Professor, Department of Computer Science and Engineering*
K. S School of Engineering and Management, Bengaluru, Karnataka, India

Abstract- The popularity of online social networks is increasing, spammers find these platforms easily accessible to trap users in malicious activities by posting spam messages. To stop spammers in Twitter, Google Safe Browsing and Twitter's BotMaker tools detect and block spam tweets. These tools can block malicious links, however they cannot protect the user in real-time as early as possible. Some of them are only based on user-based features while others are based on tweet based features only. There is no comprehensive solution that can consolidate tweet's text information along with the user based features. To solve this issue, a framework is proposed using Machine Learning techniques which takes the user and tweet based features along with the tweet text feature to classify the tweets. The tweet text feature can identify the spam tweets with more accuracy which was a little less with only user and tweet based features.
Keywords: Data Cleaning, Data pre-processing, Random forest algorithm,Social networks.

I. INTRODUCTION

In the past few years, online social networks like Facebook and Twitter have become increasingly prevailing platforms which are integral part of people's daily life. People spend lot of time in micro blogging websites to post their messages, share their ideas and make friends around the world. Due to this growing trend, these platforms attract a large number of users as well as spammers to broadcast their messages to the world. Twitter is rated as the most popular social network among teenagers. The exponential growth of Twitter also invites more unsolicited activities on this platform. Nowadays, 200 million users generate 400 million new tweets per day.

This rapid expansion of Twitter platform influences more number of spammers to generate spam tweets which contain malicious links that direct a user to external sites containing malware downloads, phishing, drug sales, or scams. These types of attacks not only interfere with the user experience but also damage the whole internet which may also possibly cause temporary shutdown of internet services all over the world. Various tweets from Bot Repository have been collected. Then further the spam tweets and non-spam tweets are characterized. Lightweight features along with the Top-30 words that are providing highest information gain from Bag-of-Words model are derived. On this processed dataset, this model is trained using machine learning algorithms for tweets classification.

The objective of proposed framework is to analyse the raw data sets and convert it into clean data sets, to collect a set of most weighted unique strong words from clean data sets(tweets), to create feature vector , creating training datasets and to create a web application that detect spam tweets.

II. PROPOSED SYSTEM

2.1 Random Forest Algorithm

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees. Random Forest Classifier being ensemble algorithm tends to give more accurate result. This is because it works on principle, Number of weak estimators when combined forms strong estimator.Even if one or few decision trees are prone to noise, overall result would tend to be correct. Even with small number of estimators = 30, it gives high accuracy around 97%.

2.2 Proposed System

The primary data collected from the online sources remains in the raw form of statements, digits and qualitative terms. The raw data contains error, omissions and inconsistencies. It requires corrections after careful scrutinizing the completed questionnaires. The following steps are involved in the processing of primary data. A huge volume of raw data collected through field survey needs to be grouped for similar details of individual responses.Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.Therefore, certain steps are executed to convert the data into a small clean data set. This technique is performed before the execution of Iterative Analysis. The set of steps is known as Data Pre-processing.It includes Data Cleaning,

DataIntegration, DataTransformation,and Data Reduction.Data Pre-processing is necessary because of the presence of unformatted real-world data. Mostly real-world data is composed of:
Inaccurate data (missing data): There are many reasons for missing data such as data is not continuously collected a mistake in data entry, technical problems with biometrics and much more.
The presence of noisy data (erroneous data and outliers): The reasons for the existence of noisy data could be a technological problem of gadget that gathers data, a human mistake during data entry and much more.
Inconsistent data: The presence of inconsistencies are due to the reasons such that existence of duplication within data, human data entry, containing mistakes in codes or names, i.e., violation of data constraints and much more.
From the collected tweet around thousands of unique words are selected out of which 30 words are identified that are possibly strong indicators for marking a tweet as spam or non-spam. In this framework the feature-set with Bag-of-Words model is sampled. Multiple unique words have been identified from the tweets' text. Using these many words along with extracted features set, a new feature set is built.After characterizing the spam and non-spam tweets' text into two separate documents, the following sets are constructed:
US = Collection of unique words in the spam tweets' text.
UNS = Collection of unique words in the non-spam tweets' text.

For each word T in US and UNS probability values has been calculated

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

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

The information gain γTis calculated for each word T as follows:

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

Words are sorted in decreasing order based on their γT. Top 15 words are taken from each of the US and UNS using above calculation and combine these words to form top-30 words that are used in feature set. The benefit of using these words based on their entropy score in the feature-set is that it helps to reduce uncertainty in the prediction outcome as these words have a different impact of frequency count in spam and non-spam tweets. Hence, considering these top 30 words will help the system classify the tweets accurately for each class.Large number of public tweets is collected. Based on tweet's text top- 30 words are extracted which are able to give the highest information gain in order to classify the tweets. As Twitter is available to all users, spammers may change their behaviour over the time. In the real world, spam tweet's feature keeps on changing in an unanticipated way. In the future, Bag-of-Words model will be kept on updated based on new spam tweets by implementing self-learning algorithm.

2.3 System Design
System Architecture design identifies the overall hypermedia structure for the WebApp. Architecture design is tied to the goals establish for a WebApp , the content to be presented, Content architecture, focuses on the manner in which content objects and structured for presentation and navigation. WebApp architecture, addresses the manner in which the application is structured to manage user interaction, handle internal processing tasks, effect navigation, and present content. WebApp architecture is defined within the context of the development environment in which the application is to be implemented.
The Fig1 represents the entire system architecture of the model. Historical data is collected and data pre-processing is performed on the same to prepare the data for extracting the feature. In the feature engineering phase, the features that are actually required for the model is extracted conceptually. Then the data is split into train and test set based upon a ratio given by the admin. The training set is used to train the model and the test set is used to validate the model.

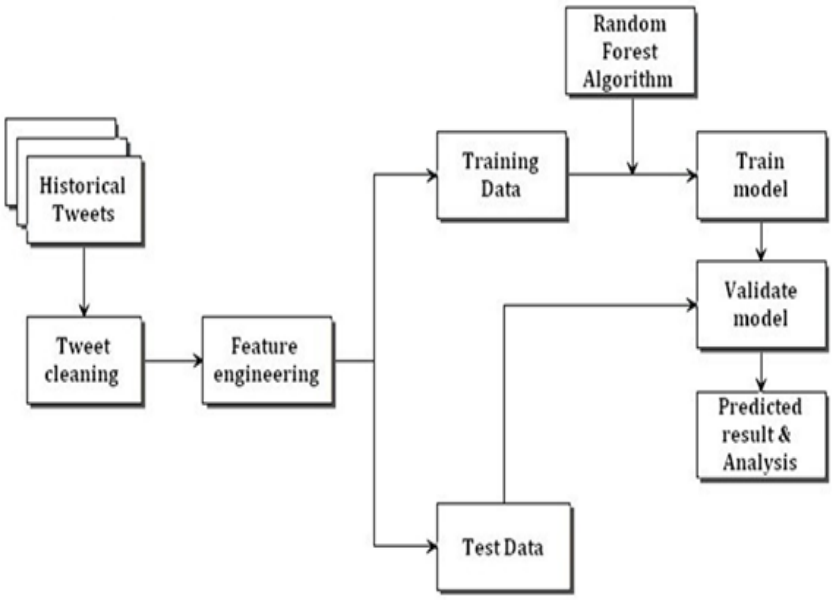

Fig1: Proposed System Architecture

2.4 FLOW CHART

The Fig 2 shows the flow of overall working of the framework. It describes when the pre-processing is to be done and when the data is split into train and test data sets and more.

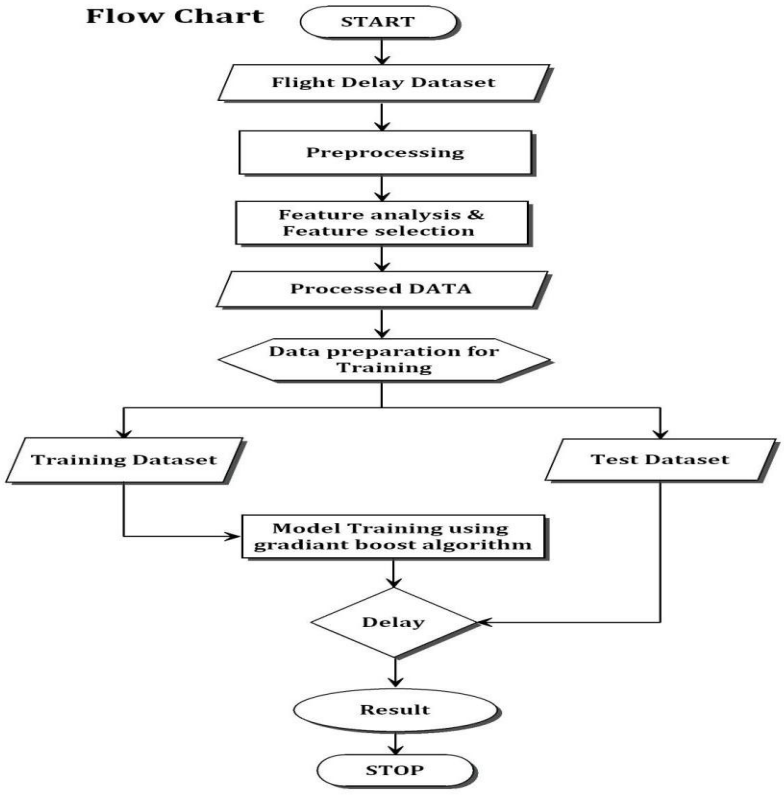

Fig 2: Flowchart for data pre processing

III. EXPERIMENT AND RESULT

The design of the project is implemented by the following modules:
1.Web application using Java servlets and SQL.
2.Pre-Processing the data.
a. Data cleaning.
b. Data integration.
c. Data transformation.
d. Data Reduction.
3. Train and Test data Creation.
4. Model Creation - Random Forest Algorithm- Classification.

The Fig 3 shows the data used is usually split into training data and test data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. The test dataset (or subset) is used in order to test our model's prediction on this subset.

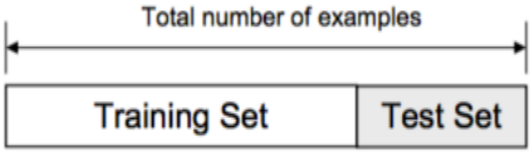

Fig 3: Train/Test Split Diagram

The process of training an ML (Machine Learning) model involves providing an ML algorithm with training data to learn from. The term ML model refers to the model artifact that is created by the training process. The training data must contain the correct answer, which is known as a target or target attribute. The learning algorithm finds patterns in the training data that map the input data attributes to the target and it outputs an ML model that captures these patterns. The ML model can be used to get predictions on new data for which the target is unknown. For example, let's say that an ML model is trained to predict if an email is spam or not spam. The training data would be provided that contains emails for which the target is known (that is, a label that tells whether an email is spam or not spam). Machine would train an MLmodel by using this data, resulting in a model that attempts to predict whether new email will be spam or not spam.

3.1 Random forest pseudo code:

Random forest is a supervised learning algorithm. From the name itself it is clear that it creates a forest of trees and somehow makes in random. The 'Forest' it builds, is an ensemble of Decision trees, most of the time trained with the "bagging" method. The general idea of bagging method is that a combination of learning models increases the overall results.
Pseudo code is an informal high-level description of the operating principal of a computer program or other algorithm. Pseudo code uses structural conventions of a normal programming language, but is intended for human reading rather than machine reading. In implementing the project process one major algorithm uses and pseudo code is presented below.For b=1 to B:
Draw a bootstrap sample Z* of size N from the training data.
Grow a random-forest tree to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size Nmin is reached.
i. Select m variables at random from the p variables.
ii. Pick the best variable/split-point among the m.
iii. Split the node into two daughter nodes.
Output the ensemble of the trees.
To make a prediction to a new point x we do:
For regression: Average the results.
For classification: Majority votes.

3.2 Results:

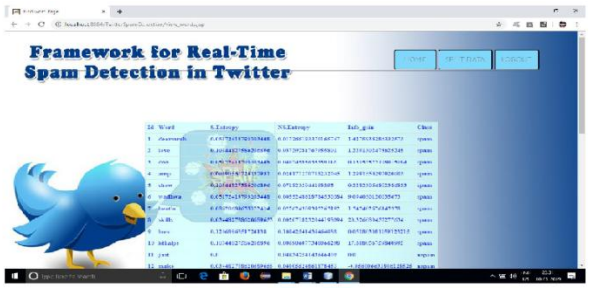

Fig 4:Bag of words page

The Fig 4 showsthe bag of model which includes the spam entropy, non-spam entropy, information gain and class of each word. If the entropy value is greater than one, then the tweet is classified as spam, non-spam otherwise.

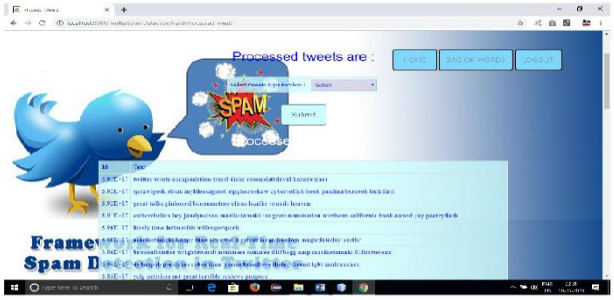

Fig 5: Spam tweets classification

The Fig 5 shows the classified spam tweets along with their corresponding tweet IDs. These tweets are classified based on the entropy values; that is if the entropy is greater than one. The admin can redirect to bag of words page, home page or can logout.

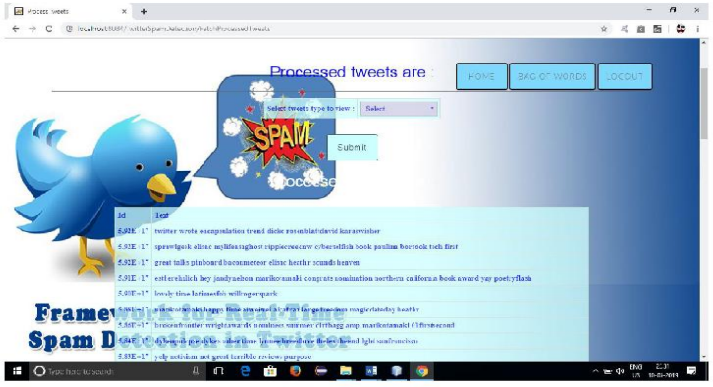

Fig 6: Non spam tweets classification

The Fig 6 shows the classified non spam tweets along with their corresponding tweet IDs. The tweets are classified based on their entropy values; that is if the entropy is lesser than one. The admin can redirect to bag of words page, home page or can logout.

IV. CONCLUSION

An attempt is made to apply supervised machine learning algorithms for the detection of spam and non-spam tweets using a dataset. Random forest algorithm is used to train the train and test data sets. Random Forest Algorithm is a classification algorithm based on the votes of all base classifiers. Furthermore, this algorithm is used to implement a classifier using machine learning methods. Large numbers of public tweets are collected. Based on tweet's text top-

30 words are extracted which are able to give the highest information gain in order to classify the tweets. As Twitter is available to all users, spammers may change their behaviour over the time. Spam tweet's feature keeps on changing in an unanticipated way over the time. In the future, Bag-of-Words model can be updated based on new spam tweets by implementing self-learning algorithm.

V. REFERENCES

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

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

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

[4] T. Wu, S. Liu, J. Zhang, and Y. Xiang, "Twitter spam detection based on deep learning," in Proceedings of the Australasian Computer Science Week Multiconference, ser. ACSW '17. NewYork, NY, USA: ACM, 2017, pp. 3:1–3:8.

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

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

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

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

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

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