Suspicious Online Forum Data Detection

Anant Chourasia[1], Mohit Agrawal[2], Sakshi Thakur[3], Ayushi Bais[4], Muskan Singh[5], and Vrushali Bongirwar[6]
[12,3,4,5,6]*Shri Ramdeobaba College of Engineering and Management, Nagpur, India.*

ABSTRACT

The exponential growth seen in the past few years of internet users and its easy access through handy available devices, As a result of which the size of digital content has been growing very rapidly in recent years through messaging applications, posts on social networking sites, vlogs, blogs, online forums and other internet platforms. With the advancement of social media sites over the internet, one's post can be reached all over the world which means it can be used to influence people in both good and bad ways. Although social media has many advantages,people misuse social media to exploit other people. The anonymity offered by internet and flexibility afforded has made it convenient for users to socialize in an aggressive and disrespectful manner over the internet. Text contents are mostly used to perform suspicious activities. Hate speech is currently an interest in the domain of social media. So, one of the most difficult issues for NLP researchers and programmers is to develop an algorithm that can detect text with hate speech automatically and efficiently from its specific contents. So here, a classification based Machine Learning model is proposed to efficiently and automatically classify text into non-suspicious and suspicious categories based on its contents.

KEY WORDS: NLP, NATURAL LANGUAGE PROCESSING, SUSPICIOUS DATA DETECTION, MACHINE LEARNING, TEXT CLASSIFICATION,TEXT MINING, FEATURE EXTRACTION, RANDOM FOREST CLASSIFIER.

INTRODUCTION

People nowadays are compulsively addicted to social networking sites. It has become an important part of our life. These sites are being used as a live platform for expressing our opinions, feelings and for promotional purposes also. Social media has now become one of the mainstream mediums for spreading unlawful and disgraceful criminal activities by fraudsters and scammers. Twitter being at the top among all other social platforms, it delegates it's end users to read, react and send tweets. Tweets are posts which are text-based and a single tweet can be of maximum 140 characters. Communication between multiple parties in a public forum is provided by Twitter, allowing you to get instant response from potential users. Since communication on Twitter is open for everyone. It is observed that often, contents in the tweets are sometimes misleading or even suspicious. In virtue of the popularity of Twitter, it is at risk of users who often try to find a way to spoil it. Of every 21 tweets and every 200 messages on social media is estimated or appraised to be unlawful or spam. A way by which this problem could be tackled is depicted in this paper.

Our proposed approach for the problem which is mentioned in the above paragraph aims on perception and detection of suspicious and hate tweets. To start with this research, we firstly develop our dataset categorising non-suspicious and suspicious texts including a number of public tweets and replies available to developers. Data of any user can be mined from Twitter using Twitter API known as Tweepy. The mined data will be the user extracted tweets. The"TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY" acronymed as TF-IDF are the

ARTICLE INFORMATION

**Corresponding Author: chourasiaa@rknec.edu*
Received 18th Oct 2020 Accepted after revision 25th Dec 2020
Print ISSN: 0974-6455 Online ISSN: 2321-4007 CODEN: BBRCBA

Thomson Reuters ISI Web of Science Clarivate Analytics USA and Crossref Indexed Journal

NAAS Journal Score 2020 (4.31)
A Society of Science and Nature Publication, Bhopal India 2020. All rights reserved.
Online Contents Available at: http//www.bbrc.in/
Doi: http://dx.doi.org/10.21786/bbrc/13.14/6

major components of the scores resulting after assigning it to each word. Once this important phase of extracting the features has been done, we used one of the popular machine learning algorithm for classification i.e. Random Forest to classify given tweets into suspicious and non-suspicious ones. We also aimed to perform a relative and comparative evaluation and analysis of these machine learning algorithmic models by utilizing our collected and proposed datasets.

Illustrated below are the key contributions of our proposed and implemented work :

- Developing a corpus consisting of 30000 texts identified and thus labelled as non-suspicious or suspicious.
- Designing a model for classifying tweets into 2 categories being suspicious and non-suspicious on corpus which is previously developed by exploring different combinations of features.
- Comparing the performance of the model proposed by us; with other various ML techniques i.e. Stemming and Lemmatization.
- Analyzing the model's performance on various different distributions of the created dataset.
- Visualizing a comparison on the performance of the model which uses machine learning algorithms (i.e., Random Forest)

We expect and contemplate that the work presented by us in this paper shall play an evolving role in the categorization of suspicious tweets and development of detection systems for suspicious texts on social media.

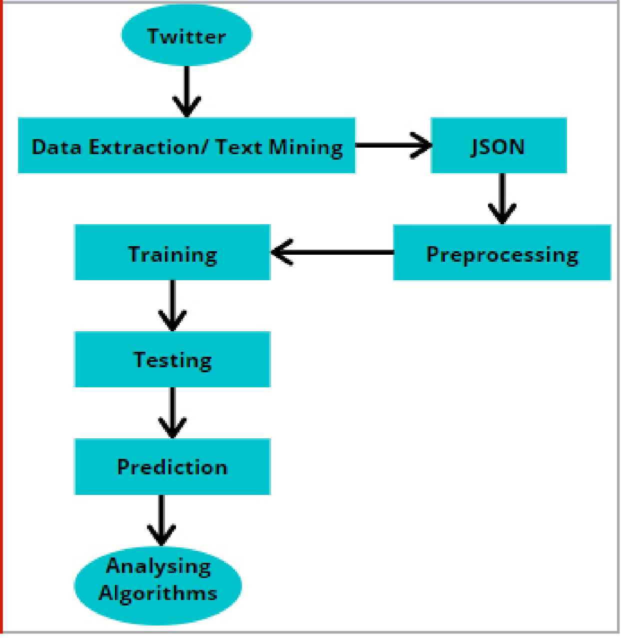
Literature Review: There exist papers which suggest detection of hate tweets using Natural Language Processing techniques; others suggest detection of malicious tweets using statistical analysis. The algorithms generally used by these authors were Naive Bayes, Logistic Regression or a deep learning method named Convolutional Neural Network (CNN). Previous work related to the same is mentioned below in brief:

i. Detecting malicious tweets in trending topics using a statistical analysis of language: In this paper, a great way to detect spam tweets in Twitter's top trending topics is discussed. This paper is an addition of the fundamental language modelling approach used to examine the branching between the language model of the topic and the suspicious messages with respect to that topic. There are five processes: Trending topic collection, spam labelling, Feature extraction, classifier training and spam detection. The result is used along with the machine learning system. The algorithms used in this paper are: Decision Trees, Naive Bayes, Logistic regression, Support Vector Machine. A value of 93.7% and 89.3% is obtained in spam and non-spam accurately classified and 6.3% of non-spam was wrongly classified as spam.

ii. Hate speech detection using language natural processing techniques: In paper, hate speech was detected using NLP. Here, hate speech is considered as abusive and hostile messages. On twitter dataset, a deep learning method named CNN has been applied. CNN consists of various layers like input layer, convolutional layer, embedding layer, pooling layer and fully connected layer. The model's input is a pre-processed tweet. The publicly available Word2Vec word embedding was used to set the weight of the embedding layer having 300 dimensions already trained on 3000 million words. It resulted in good precision and F-score. However, few non-hate speech as hate-speech and the most of hate-speech class is also wrongly identified by the model. In this paper, Keras in python was used for the pre-processing.

iii. Active chat monitoring and suspicious chat detection: This paper consists of a model based on Machine Learning to classify text in Bengali language having suspicious meaning. Various feature extraction techniques with n-gram features are used. Here, pre-processing is performed in three steps: redundant characters removal, tokenization and removal of stop words. The vectorization is done by Bag of Word and term frequency-inverse document frequency technique. The dataset consists of 7000 suspicious and non-suspicious text documents. In this paper five models: Logistic regression, random forest, decision tree, stochastic gradient descent and multinomial naïve bayes are compared with each other considering two feature extraction methods: Bag Of Words and tf-idf. An accuracy of 87.57% was achieved after applying the learning algorithms: Stochastic Gradient Descent(SGD) classifier and term frequency-inverse document frequency feature extraction along with unigram and bigram.

Figure 1: System Architecture



III. Proposed Methodology: In the proposed methodology, the fundamental step is extracting the data from social networks. This data is then pre-processed using regular expressions, stemming , unidecode and stop words are removed using Natural Language Processing. This pre-processed data is further used for text normalization and data cleaning using stemming. This processed data is further used for data processing which is done by calculating word frequencies using TfidfVectorizer. The vectors are then fed to the classifier to train the model. The trained model is tested on various machine learning algorithms and analysed the accuracy based on predictions of the algorithms. (Fig 1).

A. Data Extraction/Text Mining: For extracting tweets from Twitter, we first need to get access to the Twitter developer account. The developer account provides us the consumer key, access key, consumer secret, and access secret. These keys are of foremost importance as these keys will help in the API authentication. People with Developer accounts are allowed to access tweets by Filtering on specific keywords or requesting a sample of tweets from specific accounts. Later we can use the Tweepy API which allows us to mine the data of any user. It is a user-friendly Python library used for accessing the Twitter data. The data that is downloaded is the tweets extracted from Twitter in JSON format.

B. Pre-processing and Data cleaning: Data preprocessing is an important step in the data mining process. Data preprocessing is a data mining technique that is used to alter the raw data into a useful and efficient format. In our case, the downloaded data contains too much noise per twee. Therefore, pre-processing is done extensively for every tweet. Steps in preprocessing are mentioned below in brief:

Regular Expressions: Data Collected from the interet has some issues i.e. the downloaded data has some dirt and noise. Therefore, there occurs a need to use Regex. Regular expressions are text patterns that describe the search pattern. This is very much familiar or easy to use with text-based data sets.

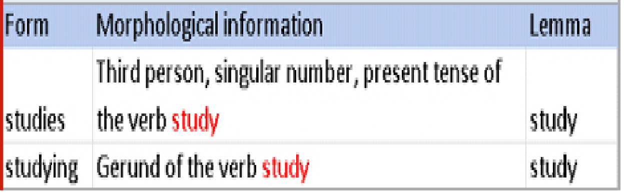
Removal of stopwords: The removal of Stopwords is not necessary for every NLP task. Removal of Stopwords can be used for text classification. Where the text has to be classified into various categories. Text classification (text categorization or text tagging) is the method of assigning a set of default categories to free-text. Text classifiers can be used to arrange, structure, categorize, and so on. For example, chat conversations can be sorted out by language, and so on. Here, we are organizing based on hate or not. The Stopwords are removed from the text so that we can work mainly on the text that defines the meaning of the text. However, removing stopwords is not advisable to use in tasks like text summarization.

Benefits of removing stopwords: If we remove the stopwords the size of the dataset decreases and it helps in training the model easily.also helps in saving space as the vectors created will consume less space.Removing stop words helps to upgrade the accuracy as there are less meaningful words left. For example, Google removes stopwords for fast and to the point retrieval of data from the database. Data cleaning is the method of detecting incorrect, incomplete, irrelevant, or inaccurate parts of the data and then modifying, replacing, or deleting the garbage data. For data cleaning, we used:

Stemming: It maps a group of words to the same stem. In this process the morphological variants of a root/ base word are produced. The output is called a base or root word even if it is not a valid word. Stemming reduces the suffix from a word like it reduces "retrieval", "retrieves" to "retrieve". Here, we are using Porter's Stemming Algorithm to pure data. It is one of the popular methods proposed in 1980 which is used for the process of removing commoner morphological and inflexional ending from words. Inflexional is basically a process of removing extra letters from nouns, verbs, adjectives.

Lemmatization: It is the process of grouping together the inflected form of words so they can be analyzed as a single item, identified as a lemma. An example of lemmatization is "Studies", "Studying" became "Study". (Fig 2) Stemming and Lemmatization are nearly similar but the difference is stemming only looks at the form of the word whereas lemmatization looks at the meaning of the word. That is after applying lemmatization we get a valid word. WordNet Lemmatizer can be used for this process. WordNet Lemmatizer uses the WordNet Database to look up lemmas of words.

Figure 2: Lemmatization example

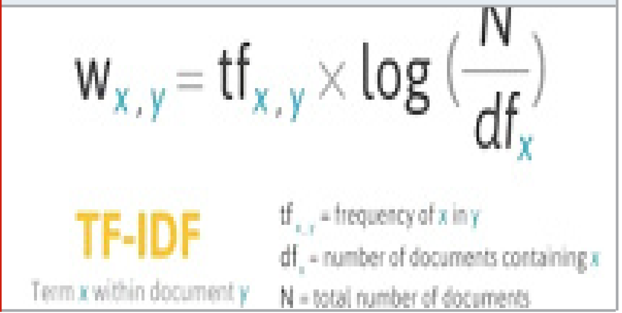


Form	Morphological information	Lemma
studies	Third person, singular number, present tense of the verb study	study
studying	Gerund of the verb study	study

C. Data Processing: The next step is Data Processing. We firstly calculated word Frequencies with TfidfVectorizer. The term TF-IDF stands for "Term Frequency – Inverse Document Frequency" which is the vector of the resulting scores assigned to each tweet. TF-IDF (for formula refer fig 2)Term Frequency: Summarizes the number of times or how often any given word appears within a specified document. Inverse Document Frequency: Downscales words that appear a lot of times across the document. Further without getting into the detailed mathematics and statistics, basically TF-IDF vectorizer are scores based on word frequency across the document that tries to give more importance to the words that are more frequently used in a document. TfidfVectorizer will tokenize documents, learn the corpus and therefore the inverse document frequency weightings and thus outputs an encoded vector. These encoded vectors are used directly by a machine learning algorithm for the classification.

D. Classifier Algorithm: The classification algorithm used in the model is Random forest. It is a supervised learning algorithm. Random forest randomly selects data samples and generates decision trees on it. These decision trees collectively are known as forest. In random forest classification, every tree votes and the final result is the most popular one. More no of trees results in better accuracy. Decision trees are created on randomly selected data samples, after which each tree predicts and the solution is selected by means of voting. It is based on the divide and conquer approach of decision trees generated on a randomly split dataset.

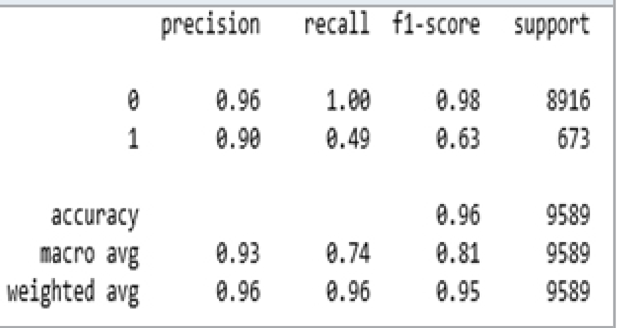
Figure 2: Formula for TF-IDF



$$W_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF
Term x within document y
$tf_{x,y}$ = frequency of x in y
df_x = number of documents containing x
N = total number of documents

Figure a: Classification report of model



```
              precision    recall  f1-score   support

           0       0.96      1.00      0.98      8916
           1       0.90      0.49      0.63       673

    accuracy                           0.96      9589
   macro avg       0.93      0.74      0.81      9589
weighted avg       0.96      0.96      0.95      9589
```

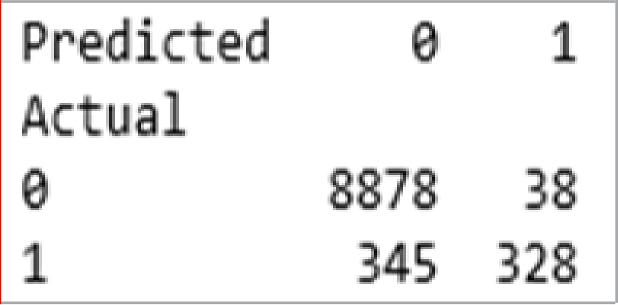
RESULTS

CONCLUSION

In this paper, we presented a way for detecting suspicious discussions on the online forums, through which we can uncover suspicious activities. Our analysis is based on particular topics, depending on the keywords to filter the streaming data from Twitter. After pre-processing the data, we used TFIDFVECTORIZER for converting text to feature vector which is used as input for our estimator. The classification Algorithm we used here is Random Forest Classifier which gives 96% accuracy. The purpose of this system is to monitor suspicious discussions on online forums and identifying them.

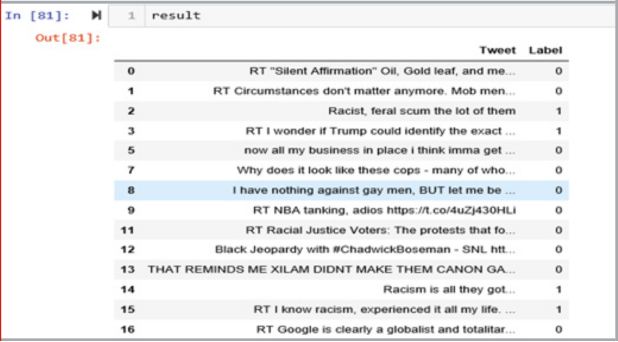
The future goal is to balance the dataset to make it less biased, apply Web Scrapping to work on the Real time streaming data and to test the system with other various models for comparison of accuracy.

Figure b: Confusion matrix



```
Predicted      0     1
Actual
0           8878    38
1            345   328
```

Figure c: Final predictions on downloaded test data.



```
In [81]:     1  result
   Out[81]:
```

	Tweet	Label
0	RT "Silent Affirmation" Oil, Gold leaf, and me...	0
1	RT Circumstances don't matter anymore. Mob men...	0
2	Racist, feral scum the lot of them	1
3	RT I wonder if Trump could identify the exact ...	1
5	now all my business in place i think imma get ...	0
7	Why does it look like these cops - many of who...	0
8	I have nothing against gay men, BUT let me be ...	0
9	RT NBA tanking, adios https://t.co/4uZj430HLi	0
11	RT Racial Justice Voters: The protests that fo...	0
12	Black Jeopardy with #ChadwickBoseman - SNL htt...	0
13	THAT REMINDS ME XILAM DIDNT MAKE THEM CANON GA...	0
14	Racism is all they got...	1
15	RT I know racism, experienced it all my life. ...	1
16	RT Google is clearly a globalist and totalitar...	0

REFERENCES

Juan Martinez-romo* and Lourdes Aroujo, "Detecting malacious tweets in trending topics using a statistical analysis of language"

Shanita Biere, "Hate speech detection using language natural processing techniques"

Omar Sharif , Mohammed Moshiul Hoque , A. S. M. Kayes , Raza Nowrozy and Iqbal H. Sarker, "Active chat monitoring and suspicious chat detection"

Speech and Language Processing. Daniel Jurafsky & James H. Martin. Copyright c 2019. All rights reserved. Draft of October 2, 2019, for all the process of text preprocessing. https://web.stanford.edu/~jurafsky/slp3/slides/2_TextProc.pdf

How and when to remove stop words in natural language processing. Using python.https://www.geeksforgeeks.org/removing-stop-words-nltk-python/

Introduction to stemming. https://www.geeksforgeeks.org/introduction-to-stemming/

Documentation of scikit learn about tf-idf vectorizer. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

How is TFIDF vectorizer in scikit-learn supposed to work. https://stackoverflow.com/questions/36800654/how-is-the-tfidfvectorizer-in-scikit-learn-supposed-to-work Random forest Classifier

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.