# Social Media Veracity Detection System Using Calibrate Classifier

P. Suthanthiradevi, S. Karthika

# Social Media Veracity Detection System using Calibrate Classifier

SuthanthiraDevi P[1][https://orcid.org/0000-0002-7386-821X], Karthika S[2]

[1,2] Department of Information Technology, SSN College of Engineering, Kalavakkam – 603110, Tamil Nadu, India

[1]devisaran2004@gmail.com,[2]skarthika@ssn.edu.in

**Abstract**
In the last decade, social media has grown extremely fast and captured tens of millions of users are online at any time. Social media is a powerful tool to share information in the form of articles, images, URLs and, videos online. Concurrently it also spreads the rumors. To fight against the rumors, media users need a verification tool to verify the fake post on Twitter. The main motivation of this research work is to find out which classification model helps to detectingthe rumor messages. The proposed system adopts three feature extraction techniques namely Term Frequency-Inverse Document Frequency, Count-Vectorizer and Hashing-Vectorizer. The authors proposed a Calibrate Classifier model to detect the rumor messages in twitter and this model has been tested on real-time event #gaja tweets. The proposed calibrate model shows better results for rumor detection than the other ensemble models.

**Keywords.** Rumor, Count Vectorizer, TF-IDF, Hashing Vectorizer, Ensemble learning

## 1 Introduction

On the emergence of online social networking services, many researchers have been interested to analyze the veracity of social media data. Nowadays social media sites like Twitter, Facebook has more popularity than other micro blogging services. It requires minimum time and cost to share information and the level of usage increases the volume and velocity of the data. People spend time on social media is increased gradually [1]. At the same time, this social media platform is speeding up the disclosure of data and broadcasting the incorrect information. A huge amount of rumor messages spread over this media during crisis time. The definition of a rumor is "An item of circulating information whose veracity status is yet to be verified at the time of posting" [2]. Rumors can affect the society as well as individuals in the following ways (i) It can disturb the authenticity of the news media. (ii) This rumor information influences the media users to accept biased stories. Some of the people and companies disseminate the rumor news for their political and financial gain [3]. For example, during the US presidential election 2016, most of the posts on social media were fake

[4]. In India, during the national election 2019, various Whatapp users were created to spread the rumor message against the current ruling party [5].During times of crisis like cyclone Gaja, a huge amount of tweets are generated by people and institutions who report various news and information related to the cyclone. A total of around 90,867 tweets are collected using various keywords and hashtags such as #Cyclone-Gaja, #SaveDelta, and @TNSDMA, which includes images, videos, and texts. Sample of fake images and twitter posts which are generated by various users are shown in fig 1.

At present detecting rumor on social media is the biggest challenge for government officials. It is important to deal with the issue of rumor message dissemination on twitter during the crisis time. The main objective of this work is to identify the tweet posts as rumor or not by using the Ensemble Classifiers such as Bagging, Boosting and Calibrate classifiers. These classifiers were trained and tested with the aid of the cyclone event #gaja dataset. The authors propose a Calibrate classifier to identify the rumor tweets with significant accuracy as compared to the state-of-art machine learning algorithms. The models have been evaluated with three feature extraction techniques namely Term Frequency-Inverse Document Frequency (TF-IDF), Count-Vectorizer (CV) and Hashing-Vectorizer (HV). A brief outline of the related work in the field of rumor detection is discussed in section 2. Static analysis for the dataset has been explained in section 3. The proposed methodology for rumor detection is explained in section 4. The performance of the classifier results are discussed in section 5. Section 6 concludes this research work.


**Fig. 1.**Sample rumor tweet and image for #gaja dataset

## 2 Related Work

Jing Ma et.al proposed a kernel learning method for detecting rumors in microblog posts. This method learns discriminate clues for detecting rumors and measure similarity among the propagation trees. It overcomes the drawbacks of the feature-

based method and allows further information discriminations. Two twitter dataset was tested on PTK model and achieves 75% accuracy in one dataset and 73% accuracy in another dataset [6].

Kwon et.al analyze the difference between the rumors and non-rumors based on the network, temporal, linguistic and user features. The time window algorithm examines rumor characteristics over short and long time windows. The authors compare the prediction level over different time windows and it was observed that, during the initial period the user features were effective for predict rumors. Linguistic features were stable and powerful predictors of rumors over a time. The Network features were used to predict information spreading on a network over a longer time period [7].

ZheZhoo et.al designed a rumor detection approach by clustering the tweets and each cluster contains enquiry patterns. The clusters are ranked based on statistical features and then compared the properties of the whole cluster into a signal tweet. By this method, the rumors in an early stage can be detected effectively and in order to improve the detection method first, they improve the filtering mechanism and correction signal [8].

Michal lukasik et.al suggests an approach for classifying judgments of rumors in both supervised and unsupervised domain adaptation. The multi task learning approach was performed effectively when compared to single-task learning [9].

Zilong et.al tracks both fake news and real news from the Twitter message in Japan and Weibo in China. Both media has spread fake news distinctively from multiple broadcasters. The real news has spread using dominant sources [10]. The authors analyze the predictability feature of this difference of the propagation networks to detect fake news in an early stage. They demonstrate filtering out fake news from the beginning of their propagation using collectivestructural signals [12].

**Table 1.**Summary of machine learning techniques and evaluation metrics

| References | Proposed Approach | Features | Evaluation Metrics |
|---|---|---|---|
| Zilong Zhao et al. | Collective Structural Model | Topological features | Heterogeneity measure |
| Kwon et al. | SpikeM Model | Structural, Linguistic, Temporal features | F1-Score |
| Arkaitz Zubiaga et al. | General Methodology | User and Twitter feature | Nil |
| AditiGupta et al. | Semi-supervised SVM Rank | TF-IDF | ROC_AUC curve |
| Sandeep Soniet et al.[16] | Predictive accuracy, cue words and cue groups | Cue word, Cue word group | Precision, Recall |

**Table 2.**Taxonomy of machine learning algorithms for credibility analysis in Twitter

| References | Types of event | | Detection Method | Data Set | Inference |
|---|---|---|---|---|---|
| | Specified | Unspecified | Supervised | | |
| Zilong Zhao et al .[11] | ✓ | | ✓ | Twitter data from Japan and Weibo in China | Identified the fake news at an early stage using collective structural signals. |
| Kwon et al. [13] | ✓ | ✓ | ✓ | KoreanFanDeath, LadyGaga, Montauk Data | Compare the prediction level over different time windows, during the initial period the user features are effective for predict rumors, linguistic features are stable and powerful predictors of rumors over time. |
| ArkaitzZubiaga et al. [14] | ✓ | | ✓ | Ferguson unrest, Ottawa shooting, Sydney siege, CharlieHebdo-shooting, German wings plane crash | True rumors tend to be resolved faster than false rumors. Rumors in their unverified stage produce distinctive bursts in the number of retweets within the first few minutes, substantially more than rumors proven true or false. |
| Aditi Gupta et al. [15] | ✓ | | ✓ | The Boston Marathon blasts in the US, Typhoon Haiyan/Yolanda in the Philippines | A real-time web-based tool to check the information credibility based on the score. |

## 3 Dataset

Due to the cyclonic storm 'Gaja' over the Bay of Bengal rainfall started between Cuddalore and Bamban on 15-November,2018 at 5.30 P.M. There were many rumor news are disseminated in twitter during this event. The authors have collected 90,867 tweets from 24,534 unique users with the aid of hashtags namely #cyclonegaja, #savedelta, and @TNSDMA. The #gaja corpus consists of source tweets, retweets and replies tweets.

The distribution of length of a tweet in terms of word counts are to be analyzed. Figure 2 shows the length of the tweet Vs total length of the tweets related to this event. It has been analyzed that there are 2,500 unique words to be identified in the dataset. The length of the each tweets are varies between 5 to 35 word counts. The collected tweets are used to model the classifier.

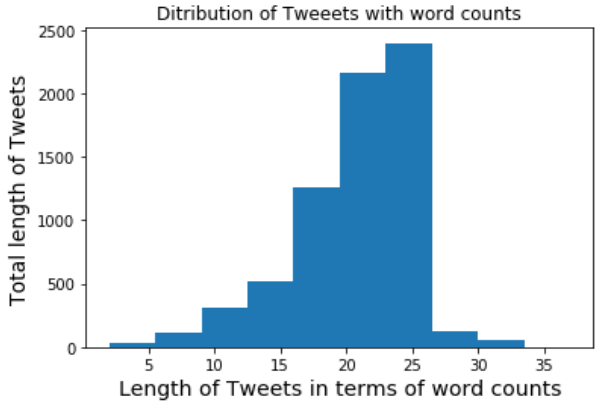



**Fig. 2.**Distribution of Tweets with word counts for the #gaja dataset

The authors validate the data annotation work through Fleiss kappa coefficient. It is used to measure the Inter Rated Reliability with three annotators for classifying the rumor or non-rumor tweets. This is derived by

$$k = \frac{\hat{p} - \widehat{p_e}}{1 - \widehat{p_e}} \qquad (1)$$

Where $1 - \widehat{p_e}$ is the degree of agreement that is achievable than the chance and $\hat{p} - \widehat{p_e}$ is the degree of agreement actually achieved above chance.The following table 3 shows that the sample annotation process of rumor tweets.

**Table 3.**Sample tweets for Data Annotation

| Tweet | A1 | A2 | A3 |
|---|---|---|---|
| Gale wind speed reaching 70-80 kmph gusting to 85 kmph prevails over East central and adjoining West central& South-east Bay of Bengal. | NR | NR | NR |
| Schools and colleges in Tamil Nadu remained shut today and offices and business establishments were asked to relie? | R | NR | R |
| Kind people of twitter #help | NR | NR | NR |
| #GAJA is very likely to further intensify during the next few days and become a severe cyclonic storm over the Bay of Ben | NR | NR | R |
| Cyclone Gaja To Turn Into ? Severe Cyclonic Storm? In Next 24 Hours, Tamil Nadu And Andhra Pradesh To Witness Heavy Rainfall.? | NR | R | R |

The annotator1 (A1) validate nearly 70% of the tweets as non-rumor and annotator2 (A2), annotator3 (A3) validate 60% of the tweets as non-rumor. The excepted probability of the overall annotation is calculated using k. The k value for this annotation process is 0.58.It has been observed that our agreement is moderate.The statics of the data annotation is shown in table 4.

**Table** 4.Statics about the data after Data Annotation

| | |
|---|---|
| Total Number of Tweets | 90,867 |
| Number of Non-rumor Tweets | 63606 |
| Number of Rumor Tweets | 27260 |

# 4 Methodology

The overall architecture of the proposed Calibrate classifier for rumor tweet detection is shown in Figure 3. This module consists of Data store, Pre-processing, Feature generation, and Ensemble Classifier. To train these models, tweets which are collected from the #gaja event are used.

## 4.1 Data Store:

Twitter allows us to mine the data of any user using Twitter API or Tweepy. The Streaming API works by making a request for a specific type of data filtered by keyword, user, geographic area. Tweets were collected using various hash-tags like #CycloneGaja, @TNSDMA, #SaveDelta, etc. The tweets collected in a streaming fashion represent the tweets that were posted in that particular time duration.

### 4.2 Data Pre-processing

To remove stop words such as pronouns, conjunctions, and prepositions there were eight preprocessing rules are applied. This noise reduction in the text helps to improve the performance of the classifier and remove the textual content not related to the event. The preprocessing rules are Convert to lowercase, RT removal, Replacement of User-mentions, URL Replacement, Hash Character Removal, Removal of Punctuations and Symbols, Lemmatization and Stop word Removal. The preprocessed data is then fed into feature generation module.

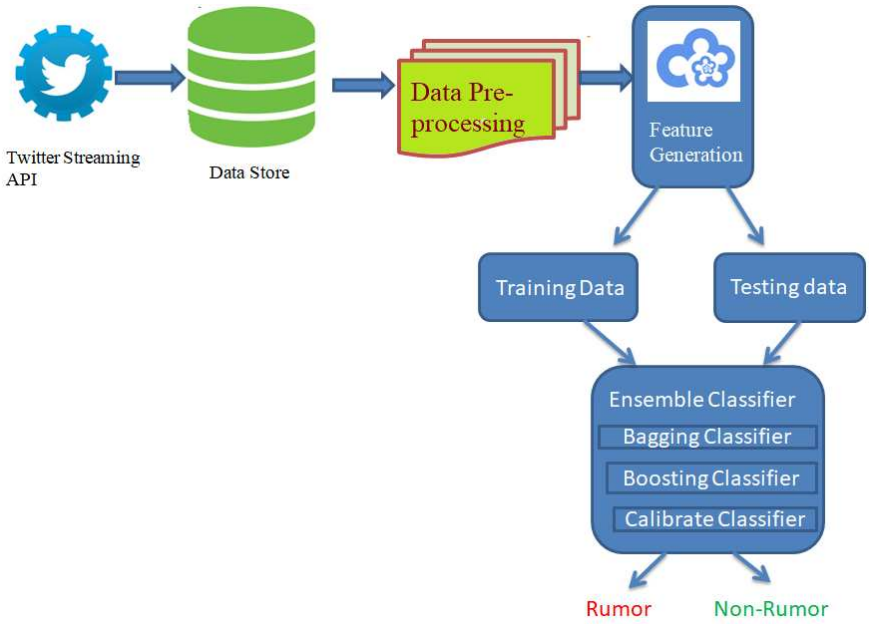



**Fig. 3.**The architecture of the proposed rumor detection system

### 4.3 Feature Generation

In this module, the features are extracted from the pre-processed data. In order to convert the collected text document into an integer or floating-point values are known as feature vectorization.

**Count Vectorizer (CV)**

Count Vectorizer converts the word into the matrix of token counts. A CV is based on count of the word occurrences in the document. CV is selected for feature extraction because it has performed both the tokenization and counting the occurrence of the word in the data. It is observed that CV converts each word in the document as a vector value (integer). Each vector consists of the feature name and its corresponding word occurrences. Each column in the matrix contains the terms like cyclonegaja,

gate, and speed as feature names. The rows (doc1,doc2) represents the frequency of words retrieved from the vector and their corresponding word count. The following fig 4 shows that the sparse matrix for sample #gaja dataset. In this matrix doc0 and doc1 represent the number of words retrieved from the dataset. The limitation of this technique is less frequency terms have more influence than the high frequency words.

| | cyclonegaja | gale | kmph | reaching | speed | user | warning | wind |
|---|---|---|---|---|---|---|---|---|
| **Doc0** | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| **Doc1** | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

**Fig. 4.**Feature Generation using Count Vectorizer

**Hashing Vectorizer (HV)**

Hashing Vectorizer tokens are encoded as a numerical index. It requires only limited amount of memory for feature generation. This HV is chosen for simplifying the implemention of the bag-of-words and improves the scalability. It has generated the hash value for a given dataset. The most popular MURMURHASH3 hashing algorithm is applied to the hashed words to generate a random number. These values are divided by the length of the data and find the corresponding remainder value.

| Documents | MURMURHASH3 | Divide By | Remainder |
|---|---|---|---|
| User | 34875149 | 8 | 5 |
| Cyclone gaja | 31900345 | 8 | 1 |
| Wind | 36750019 | 8 | 3 |

| Key | Value |
|---|---|
| K1 | User |
| K2 | |
| K3 | cyclonegaja |
| K4 | |
| K5 | Wind |
| K6 | |

| Key | K1 | K2 | K3 | K4 | K5 | K6 |
|---|---|---|---|---|---|---|
| Value | user | | cyclonegaja | | wind | |
| | Featue1 | Featue2 | Featue3 | Featue4 | Featue5 | Featue6 |
| | | | | | | |

**Fig. 5.**Feature Generation using Hashing Vector

Based on the remainder values every word is stored into the corresponding key-value pairs. The hash value of the textual data is estimated using MURMURHASH3 hash function is shown in fig 5. In this matrix each column represents the key values and row contains the feature names. The limitation of this technique it doesn't retrieve the feature names.

**Term Frequency-Inverse Document Frequency (TF-IDF)**

TF-IDF is used to generate a weighted matrix for the important words in the dataset. The following fig 6 shows that the TF-IDF weighted matrix for preprocessed data. It is observed that TF-IDF are tokenized the documents, learn the word and assign weights for each word. In this matrix the columns are represented by the token and the row (doc0,doc1) represent weighted value for number of words retrieved from the dataset. Beel et al.[18] showed that 83% of text based categorizations are done by using the tf-idf vectorization technique.

| | cyclonegaja | gale | kmph | reaching | speed | user | warning | wind |
|---|---|---|---|---|---|---|---|---|
| **Doc0** | 0.499221 | 0.3552 | 0.000000 | 0.000000 | 0.000000 | 0.499221 | 0.499221 | 0.3552 |
| **Doc1** | 0.000000 | 0.3552 | 0.499221 | 0.499221 | 0.499221 | 0.000000 | 0.000000 | 0.3552 |

**Fig. 6.** Feature Generation using TF-IDF

### 4.4 Ensemble Classifier

The feature generated dataset is fed into the classifier for the detection of the rumor tweet. Ensemble methods are to build a learning algorithm in a statically and computational way. It is used to deal imbalanced data efficiently. Rumor detection system is implemented by using ensemble methods. In this research work the ensemble methods such as Bagging, Boosting and Calibrate Classifiers are used to detect the rumors. The bagging classifier extracts a subset of the training dataset from multiple models. Boosting classifiers learns to fix the prediction errors of a prior model chain. Calibrate classifiers are used to combine the predictions of multiple models. In this experiment, the authors have used non-linear classifier to detect the rumors. But these classifiers are generally predicting uncalibrated results. Calibrate classifier is used to turn the output of the model into well-calibrated continues probabilities of the models. Rumor and Non-rumor tweets are classified with the aid of the ensemble learning classifiers.

# 5 Results and discussions

## 5.1 Pre-processing

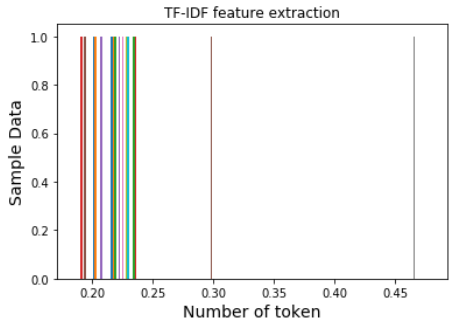
The tweets repositories of 90,689 tweets were annotated by manually as either Rumor or Non-rumor class with respect to the ground truth obtained by official user of @TNSDMA. The tweets are pre-processed by applying rules as discussed in data pre-processing section. The following table 5 shows the original and preprocessing tweets.

**Table 5.**Pre-processing of sample tweet for #gaja event

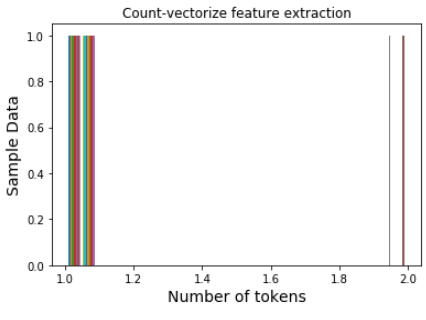
| | |
|---|---|
| Original Tweet | RT @ndmaindia: #CycloneGaja #Wind Warning: Gale wind speed reaching 70-80 kmph gusting to 85 kmph prevails over East central and adjoining W? |
| Preprocessed tweet | user cyclonegaja wind warning gale wind speed reaching kmph gusting to kmph prevails over east central and adjoining |

## 5.2 Feature Generation

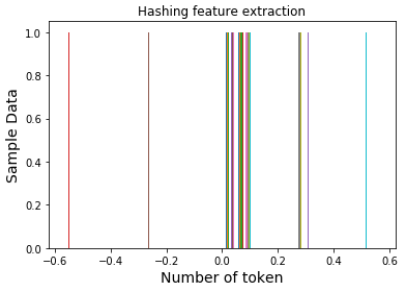
The preprocessed tweet messages are applied for feature generation. Both content and contextual features are computed using three methods such as TF-IDF, Count Vectorizer and Hashing Vectorizer techniques. The following figure 7 shows the vector generation for the above-preprocessed tweet. The conversion of textual content into the numerical vector is implemented using TF-IDF, CV and HV techniques. The vector graph shows that the learned vocabulary and number of document frequencies. In this graph where x-axis represents number of token values, whereas y-axis represents the size of the sample data.Fig 7(a), 7(b) and 7(c) are visualizing the encoded vector values for the preprocessed text. The normalization value for TF-IDF is (0, 1), CV (1,2) and for HV (-1,1). In fig 7(a) show the result of vector using TF-IDF, most frequently used words in the documents are shadowed between 0.20 to 0.25 and less frequent values are showed near (0.30,0.45)




(a)




(b)




(c)

**Fig. 7.**Numerical vector generation using TF-IDF, CV and HV techniques

Fig 7(b) show the results of CV, the frequent occurrence of the words is shadowed near 1 and least is shadowed near 2. In Fig 7(c) show the hash values for most frequent and less frequent values between (-1,1). The generated feature values are fed into classifier to detect the rumor tweet in #gaja dataset.

### 5.3 Calibrate Classifier for Rumor Detection System

In this experiment, the classifiers are trained using #gaja dataset. This dataset is divided into 80% for training and 20% for testing. Three vectorizer techniques are applied on the following classifiers.

**Bagging Classifier**

Bagging or Bootstrap aggregation takes multiple samples from the original dataset and train the classifier. This classifier can be trained on models Xm={$x_1$,$x_2$,....$x_n$} using the original dataset Ym={$y_1$,$y_2$,....$y_m$},then the average model is derived by

$$x = \frac{1}{N}\sum_{N=1}^{N} x_n \qquad (2)$$

where 'n' represents the total number of data and $x_n$ represents the number of models. The Bagging is implemented using the Decision Tree classifier with hundred numbers of trees.

**Boosting Classifier**

This classifier trains weak models using training data. It has computed the error of the model and gives more importance to the mistake models. Retrain the model by using weighed training samples. The probability of the selected classifier is derived by

$$P = \frac{1+m_i^n}{\sum_{j=0}^{N} 1+m_j^n} \qquad (3)$$

where n represents the total number of data and $m_i^n$ represents the number of models. The boosting is implemented using the AdaBoost classifier with seventy number of trees.

**Calibrate Classifier**

Calibrate classifier combines the predictions from multiple machine learning classifiers such as Logistic Regression, Decision Tree and Support Vector Machine.

All predicted values are added and averaged to form the probability vector is derived by

$$\hat{Y} = \arg \frac{1}{N_c} \sum_c (p1, p2 \dots pn) \quad (4)$$

where $p1, p2 \dots pn$ represents the number of predictions and $N$ is the number of classifiers.

The Bagging, Boosting and Calibrate classifiers are implemented with the aid of vectorizer values for detecting the rumors. The detection of rumor instances using TF-IDF technique performs well for #gaja events. It can achieve accuracy of 95.7% for bagging, boosting and 96.4% for calibrate classifier. It can learn the vocabulary from the data and apply inverse frequency weights to encode the data. CV only counts the word occurrence and does not assign any weighted values. HV doesn't return feature names for further analysis. The authors have inferred that TF-IDF vectorizer Vs classifier perform well on a high volume of data. Since, TF_IDF assigns accurate weighted vector for each word in the dataset. The rumor classification results for the #gaja dataset with respect to the accuracy are shown in figure 8.

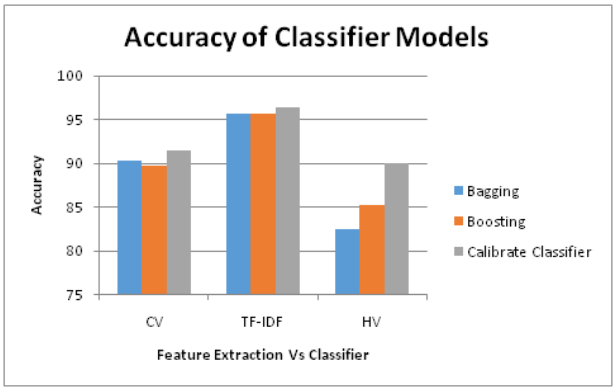



**Fig.** 8.Performance Analysis using Accuracy metric for #gaja data

**Table 6.**Comparison of existing models with a proposed system

| Author | Proposed Approach | Feature Generation Method | Accuracy |
|---|---|---|---|
| Sawinder et al. | Classify news article as fake or real using various machine learning classifiers | TF_IDF | 93.8 |
| | | CV | 89.3 |
| | | HV | 86.3 |
| Proposed Calibrate Classifier | Rumor message detection system is proposed to achieve high accuracy. | TF_IDF | 96.42 |
| | | CV | 91.45 |
| | | HV | 89.91 |

The Calibrate Classifier with TF_IDF performs better due to the fact it combines multiple classifiers for predicting the rumor tweet. The results of the calibrate classifier are compared to the existing classifier on false news detection as shown in Table 6. Based on the comparison of existing model, the authors conclude that the performance of the Calibrate Classifier is better than other ensemble classifiers.

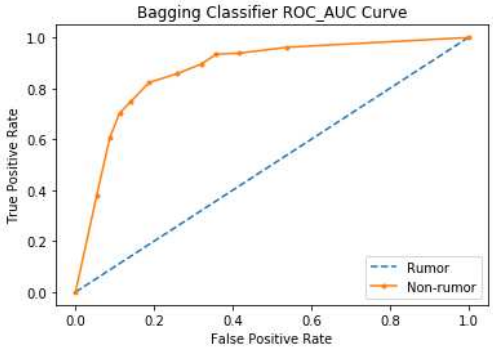
## 6 Evaluation metrics

The classifiers of ensemble learning algorithms are evaluated based on the ROC-AUC curves. ROC curve is used to predict the probability of binary classification. This curve takes the true positive and false positive values. The True Positive and False Positive Rate are derived by

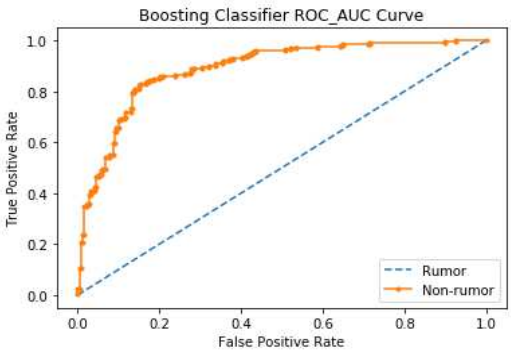
$$\text{TP Rate} = TP/(TP+FN) \quad (5)$$

$$\text{FP Rate} = FP/(FP+TN) \quad (6)$$

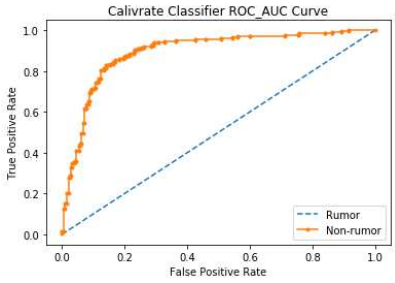
The ROC curve used to compare different models directly and the summary of the classifier is evaluated by using the area under the curve. All the three classifiers are evaluated based on the rank and measures how often a classifier ranks true positives higher than true negative. It can check the classifier output actually matches the probability of the event. The performances of the three classifiers are evaluated using the ROC-AUC curve. ROC-AUC are appropriate when the model with perfect skill value between (0,1). The ROC curve is plotted using the TP Vs FP values. It is observed that the raise in false positive values with an increase in true positive values by varying a threshold of the ensemble classifiers. In fig 9(a),9(b) and 9(c) represents the TP and FP values for all classifiers. The ROC_AUC curves are plotted true positive rates on y-axis and false positive rates on x-axis.




(a)




(b)




(c)

**Fig. 9.** True Positive and False Positive Rate for the rumor detection system

The roc_auc_score() function takes both the true values between (01) from the training and testing probabilities for one class. It has returns the auc score between (0.0,0.1) for non-rumor and rumor values respectively. Calibrate Classifier outperforms the other classifiers with the true-positive rate of 0.894.

## 7 Conclusion

The main goal of this research work is to detect the rumor messages in social media and analyze the best classifier to prevent the rumor message dissemination. The authors are experiments with the #gaja dataset using three vectorizer techniques TF-IDF, CV, and HV. Ensemble classifiers are used to classify the rumor and non-rumor messages. The results show that the Calibrate Classifier outperforms than the bagging and boosting classifiers. The experimental results are evaluated with the aid of the ROC-AUC metrics and it proves the calibrate classifier results are accurate.In future, the rumor detection system to classify the rumor data based on the retweet count.

## 8 References

1. https://www.socialmediatoday.com/marketing/how-muchtime-do-people-spend-social-media- infographic/
2. ArkaitzZubiaga, Ahmet Aker, KalinaBontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. ACM Computing Surveys (CSUR) 51, 2 (2018), 32.
3. Shu, K.; Sliva, A.; Wang, S.; Tang, J.; and Liu, H. 2017. Fake news detection on social media: A data mining perspective. ACM SIGKDD Explorations Newsletter 19(1):22–36.
4. https://www.independent.co.uk/life-style/gadgets-andtech/news/tumblr-russian-hacking-us-presidential-election-fakenews-internet-research-agency-propaganda-bots-a8274321.html
   arXiv:1809.01286v1
5. Fake news on whatsapp. http://bit.ly/2miuv9j. Last accessed: 27-08-2019
6. Ma, Jing, Wei Gao, and Kam-Fai Wong. "Detect rumors in microblog posts using propagation structure via kernel learning." *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1. 2017.

7. Kwon, Sejeong, Meeyoung Cha, and Kyomin Jung. "Rumor detection over varying time windows." *PloS one* 12.1 (2017): e0168344.
8. Zhao, Zhe, Paul Resnick, and Qiaozhu Mei. "Enquiring minds: Early detection of rumors in social media from enquiry posts." *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2015.
9. Lukasik, Michal, Trevor Cohn, and KalinaBontcheva. "Classifying tweet level judgements of rumours in social media." *arXiv preprint arXiv:1506.00468* (2015).
10. Zhao, Zilong, et al. "Fake news propagate differently from real news even at early stages of spreading." *arXiv preprint arXiv:1803.03443* (2018).
11. ArkaitzZubiaga and HengJi. Tweet, but verify: Epistemic study of information verification on twitter. Social Network Analysis and Mining, 2014
12. Kwon, Sejeong, Meeyoung Cha, and Kyomin Jung. "Rumor detection over varying time windows." *PloS one* 12.1 (2017): e0168344.
13. Zubiaga, Arkaitz, et al. "Analysing how people orient to and spread rumours in social media by looking at conversational threads." *PloS one* 11.3 (2016): e0150989.
14. Gupta, Aditi, et al. "Tweetcred: Real-time credibility assessment of content on twitter." *International Conference on Social Informatics*. Springer, Cham, 2014.
15. Soni, Sandeep, et al. "Modeling factuality judgments in social media text." *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2014.
16. Ahmed, Hadeer, Issa Traore, and Sherif Saad. "Detection of online fake news using N-gram analysis and machine learning techniques." *International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*. Springer, Cham, 2017.
17. Kaur, Sawinder, Parteek Kumar, and Ponnurangam Kumaraguru. "Automating fake news detection system using multi-level voting model." Soft Computing (2019): 1-21
18. Beel, Joeran, et al. "paper recommender systems: a literature survey." International Journal on Digital Libraries 17.4 (2016): 305-338.