Rumor Detection in Social Media with User Information Protection

Md. Rashed Ibn Nawab, Kazi Md. Shahiduzzaman, Titya Eng, and Md Noor Jamal

***Abstract*—Many researchers have already shown that only user-based or content-based features are not enough to detect rumor in social media, and for better prediction, we need to consider both. In our research, we argue that the word embedding feature and sentiment score with subjectivity can also play a vital role in this detection task. Moreover, to detect the rumor at a very early stage and debunk it, we may need to make the detection framework portable to legitimate users. This critical situation demands a secure implementation of rumor detection framework so that the user information used for training the prediction model can be protected from unauthorized access. In our experiment, we have also found that besides SVM, Logistic Regression and Random Forest algorithms, Artificial Neural Network and k-Nearest Neighbor can be used for rumor detection purpose where Artificial Neural Network and Random Forest outperformed (more than 90%) among all these algorithms in terms of accuracy. The other three algorithms also performed well with 80% or more accuracy level. To establish the robustness and efficiency of our proposed rumor detection mechanism, Precision, Recall, F1 Score, 10-fold Cross-Validation, MCC, Confusion Matrix performance measures are used.**

***Index Terms* — RSA Cryptography, Feature Extraction, Machine Learning, Rumor, Word Embedding.**

I. Introduction

We start the introduction part of this study with a proper definition of 'Rumor'. In [1], Rumor is defined as stimulating information or news which is not officially confirmed yet by any mainstream news media or relevant stakeholder but already got attention by people and has spread quickly from person to person. Rumor can also be defined as unverified but contextually relevant online messages which propagate fast and wide when ambiguity rises [2]. From the above two definitions, two characteristics of the rumor are visible to us. First of all, the information or news is doubtful. However, the piece of news or information is very attention-grabbing. Secondly, the rumor circulates very quickly from person to person via any media. On the other hand, for the sake of computing and analysis, we need more insight into rumor. Being inspired by the renowned work by DiFonzo and Bordia in the field of social psychology, the authors proposed a practical definition of rumor, "a rumor is a controversial and fact-checkable statement" [3]. Now the question arises, why rumor detection is so important nowadays. The alarming news is Gartner's research [4] predicts that "By 2022, most people in mature economies will consume more false information than true information". On the other hand, beside the traditional news media platforms, social media are increasingly being used as a tool for gathering and spreading information. Some news articles hosted or shared on the social media platforms have more views compared to direct views from the media outlets' platform. The most horrifying information we have got from the researchers who study the velocity of ambiguous news in social media. They concluded that tweets containing disinformation reach people on Twitter six times faster than the tweets, which are true [5]. In 2015, a rumor spread in social media that the famous multinational food chain Kentucky Fried Chicken (KFC) is selling fried rat in place of fried chicken [6]. Though finally, KFC managed to prove themself not guilty. However, this rumor outbreak was a severe threat to its business reputation worldwide. Not only that, sudden rumor may create public anxiety and make the state unrest. For example, the death hoax of Singapore's first Prime Minister Lee Kuan Yew in March 2015 put the whole nation in deep sadness and increased public anxiety [2]. Even we have witnessed the impact of using social media by a huge community and spreading rumored news in US President Election 2016 [7]. During this political campaign time, supporters of both the candidates, Hillary Clinton, and Donald J. Trump showed huge zeal in social media, especially on Twitter to get more support for their leaders. But the situation became worse when some rumored news came forward to humiliate the opposition leader. Among those rumors, one severe claim by a doctor came in front with some photos and videos that the Democratic Presidential Candidate, Hillary Clinton, has the symptoms of Parkinson's disease questioning her physical stability to be the President of USA [8]. Hence, recently rumor is seen as one of the greatest threats to democracy, free debate, and a peaceful society at large.

From the above discussion, it is quite evident that we need to get rid of the rumors. The most effective way to solve this problem is to detect the rumor at an early stage efficiently and quickly. The faster we can identify a rumor the faster we can minimize its impact on the online and social community. Nevertheless, the detection of the rumor is not easy because the data from social media is unstructured and noisy. Besides, because of the recent increasing user privacy policies, it is getting quite difficult to get social media data for analysis purposes. Hence, we need to find a mechanism that will ensure the privacy and security of social media data. In our

Published on July 01, 2020.

Md Rashed Ibn Nawab, Northwestern Polytechnical University, China.
(e-mail: rashednawab@outlook.com)
Kazi Md Shahiduzzaman, Jatiya Kabi Kazi Nazrul Islam University, Mymensigh, Bangladesh.
(e-mail: shahiduzzaman@jkkniu.edu.bd)
Titya Eng, University of Battambang, Cambodia.
(e-mail: engtitya@outlook.com)
Md. Noor Jamal, Huazhong University of Science and Technology, China.
(e-mail: noorjamal97531@gmail.com)

DOI: http://dx.doi.org/10.24018/ejece.2020.4.4.219

research, we have carefully designed some questions so that finding the answer to these questions will help us to formulate the research goal and move forward to that goal.

1. Which machine learning method is more suitable to detect rumor in social media?
2. Can Neural Networks also be used to detect rumor in social media?
3. To what extent can we detect rumor in social media?
4. Keeping the training data secure, how can we make our framework portable?

We are going to conduct this research with two primary targets. Firstly, we want to develop a framework to classify rumor and non-rumor messages in social media so that it can help to detect and debunk the rumor. For this purpose, we have used different text preprocessing and feature extraction techniques, machine learning algorithms like Logistic Regression, SVM, Random Forest, k-Nearest Neighbor, Artificial Neural Network. We used user-based and content-based features. However, here we argue to put more emphasis on the content-based features. We utilized different performance measures like Accuracy, Precision, Recall, F1-Score etc. to check the robustness of our experiment. Secondly, we want to make our framework portable by developing a simple and user-friendly software considering the fact that the training data for this kind of experiment contains sensitive user information and pattern of thousands of social media users which the user may not want to expose publicly or analyze by unknown third parties. The reason behind considering the portability of this framework is, the more portable the framework is, the more rumor can be debunked at an early stage. Considering the overall scenarios, we have proposed to use RSA cryptography algorithm to secure the social media extracted training data from publicly exposing while making the framework available to the authorized and legitimate end-users like the law enforcement organizations or news media.

The remaining of this research work is organized as follows. In Section-II, we focus on identifying research gaps. We talk about the methodology in the following section in detail. Section-IV discusses the experiment evaluation methods in brief. Next, we compare our results with other core references and analyze them in deep. Finally, we summarize our work in conclusion with a few future research directions.

II. Literature Review

When we start talking about rumor detection mechanisms, we must be aware of the fact that despite being a new research area, there are several ways to detect rumor in social media. Like many researchers think that rumor correction can help to detect a rumor and create adequate public awareness to debunk it. The challenge in this approach is quite tricky to identify rumor correction in the emergency. However, identifying rumor and rumor correction is getting more and more attention from many researchers, and some of them think message characteristics can play a vital role in distinguishing rumor and rumor correction.

Like in [2], authors showed that three message characteristics, the use of emotions, clarity, and credible source attribution, could help to identify whether the message or statement is a rumor or rumor correction. They also examined how the relationship between message characteristics and message veracity is influenced by opinion leadership. For the analysis purpose, they used binary logistic regression and considered only one event, the death hoax of Singapore's first prime minister spread over Twitter. Their dataset comprised of the 5885 tweets associated with the death hoax of Singapore's first prime minister on 18th March 2015. The limitations of this paper are they used a single dataset from a single source; they did not develop any classifier for the online message or statement veracity prediction. This paper could not link any relation between user motivation and message veracity. Moreover, there was further scope to automate manual coding of tweets for dataset preparation.

Based on the fact that the rumor and the counter-rumor, aka anti-rumor, rumor denial portrays the opposite pattern, the researchers in [6] took the privilege to classify rumor and counter-rumor and expressed it as a way to detect and debunk rumor. For this classification purpose, they retrieved total of 1,052 tweets from Twitter and used content-based features and user-based features of the collected tweets. All the tweets collected were in the context of the wrong accusation on KFC's fried rat selling. To check the robustness of their proposed system, they used several classification algorithms, Naïve Bayes, JRip, Random Forest, SVM, and Voting. They used five performance evaluation metrics to evaluate the performance of the classifiers, namely, Precision, Accuracy, Recall, F1-measure, and ROC area. The main contribution of this paper is it mainly focuses on distinguishing between rumor and counter-rumor, and it showed that a meta-classification approach could perform better than individual classifier. However, the performance of SVM was quite similar. The limitation of this research is it considered a dataset of a single rumoring incident, and they could propose a comprehensive feature set to identify rumor and counter-rumor tweets. Besides, they achieved 88% accuracy by Voting, a meta-learning classification algorithm where we achieved 94% and 91% accuracy by Random Forest and ANN respectively. Unlike us, they did not consider word embedding features of the tweets.

Unlike the previous two approaches, we have found a different approach to detect rumor in [4]. Besides the inherent features of the microblog, in this research, they focused on user behavior in a Chinese microblogging system named Sina Weibo for detecting rumor. The authors of this paper considered the behavior of both the author and the followers of any rumor post. One of the key contributions of this research is they introduced five new user behavior features named (1) average number of followees per day, (2) average number of posts per day, (3) number of possible microblog sources, (4) ratio of questioned comments and (5) number of corrections where the first three features are post author behavior features and the others are post follower or reader's behavior features. Combining with four other features, verified user or not, number of followers, number of retweets and comments, total 9 features are retrieved from the microblog post to identify rumor characteristics in a message. They also accepted the claim that the overall rumor detection is a binary classification task and used Logistic Regression, Decision Tree, SVM, Naïve Bayes, k-Nearest Neighbor classification algorithms to check the efficacy and efficiency of the proposed approach. In their experiment Precision, Recall and F-Score reached 0.8645, 0.8535, and 0.8590 respectively. Another key value of this research is that they have also found out the rumormongers and the microblogs posted by their followers and followees. The limitation of this research is the approach they proposed is social media platform-specific, only applicable for Sina Weibo. They also did not consider the sentiment or emotion of the users. They also put no emphasis on the rumor or non-rumor text shared by the users.

In [9] the authors proposed a framework and developed a web interface for general people to check the veracity of tweets. There are two major components in the proposed system, (1) Core and (2) Website. For the analysis purpose, they extracted seven user features and 13 content features from Twitter's metadata. In this paper, the authors not only considered the user-based and text-based features but also took media content into account. Finally, they used the J48 Decision Tree and SMV classification algorithm to determine whether a tweet is fake or not. Being inspired by their research work, we also used 2 features, isURLCredible and sentimentScore, and proposed another content-based feature, hasMedia, in our work.

III. Methodology

A. Dataset

The most authentic way to access, collect, and store data from these platforms is to use their APIs (Application Programming Interfaces). Nevertheless, before using these APIs we must read the documentation very carefully and get the approval for API usage, which is time-consuming and very difficult to get nowadays because of enhanced user privacy issues. It is worth mentioning here that only Twitter has both REST API and Streaming API to get data from its database and in real-time respectively. On the contrary, Sina Weibo only has REST API, and Facebook provides some software development kits to develop apps on its data. Considering all the constraints, we have decided to build our model on the available Twitter dataset for rumor detection and veracity classification. The dataset we have used here is known as "PHEME Dataset of Rumours and Non-rumours" [10]. It contains annotated rumor and non-rumor post on Twitter during the following five breaking news,

- Charlie Hebdo: 458 rumors and 1,621 non-rumors.
- Ferguson: 284 rumors and 859 non-rumors.
- Germanwings Crash: 238 rumors and 231 non-rumors.
- Ottawa Shooting: 470 rumors and 420 non-rumors.
- Sydney Siege: 522 rumors and 699 non-rumors.

From the preliminary description of the dataset, it is visible that the overall dataset is imbalanced. This dataset contains a total of 1972 rumor tweets and 3830 non-rumor tweets. For checking the robustness of our proposed framework, from this dataset, we have randomly chosen 1371 rumor tweets and 1621 non-rumor tweets for further rumor analysis. For dividing the dataset into training and testing data, we set the test_size to 30%.

B. Feature Selection and Extraction

Feature Selection is also known as Attribute or Variable Selection is the selection process of attributes or features in the dataset. The primary purpose of feature selection is to remove unnecessary features. It is also different from dimensionality reduction in a sense that dimensionality reduction methods reduce the number of attributes by creating new combinations of attributes or features. Feature extraction is the process of converting the original dataset into a sub-dataset with a reduced number of variables and these variables are chosen in such a way that it is suitable for modeling and enables the machine learning models to discriminate among the records. In our research, we have categorized all features extracted into two categories, (1) Content-based features, (2) User-based features. Here user-based features are the property attributes of the Twitter user, and content-based features are the content representations of the tweet posted by that respective user. Unlike other researches in this area we have reduced the number of user-based features to 5 only. However, we put more emphasis on content-based features and proposed word embedding features, subjectivity coupled with sentimentScore and hasMedia with other 5 user-based features and 18 content-based features of the tweet for rumor detection purpose. A brief description of the user-based features and content-based features are enlisted in Table I and Table II.

TABLE I: USER-BASED FEATURES

Features	Description
isUserVerified	Represents whether the Twitter user is verified or not. If the user is verified then the value of this feature is set to 1, and 0 otherwise.
friend2followerRatio	This is the ratio of the number of friends to the number of followers of a specific user.
countStatuses	Symbolizes the total number of tweets shared by the user including retweets.
countListed	Counts the total number of subscriptions to the public lists by the user.
countFavourites	Denotes the total number of liked tweets by this user in the account's lifetime.

TABLE II: CONTENT-BASED FEATURES

Features	Description
hasQuestionMark	Represents whether the tweet contains a question mark. In presence on question mark this feature value is set to 1 otherwise, 0.
questionMarkCount	Counts the total number of question mark in the tweet text.
hasExclamationMark	Symbolizes if the tweet text contains an exclamation mark. If it had the value of this feature is set to 1, and 0 otherwise.
exclamationCount	Counts total number of exclamation marks in the tweet text.
hasHashtag	The value of this feature is set to 1 if any hashtag is identified in a tweet, and 0 otherwise.
hashtagCount	Total number of the hashtag in a tweet text.

hasURL	This feature value is set to 1 if the user has shared an URL in the tweet text. In opposite, it is set to 0.
isURLCredible	Checks the credibility of the URL source. If the credibility of the URL is questioned then the value is set to 0, and 1 otherwise. But if no URL is available in a tweet then this feature value is set to 2.
hasMedia	Checks whether the tweet contains an image or video shared with it. If yes, value is set to 1. In exception, it is set to 0.
hasUserMention	Denotes whether the user mentioned any other Twitter user in his/her tweet. If yes, the feature value is set to 1 otherwise 0.
u2lCaseRatio	This is the ratio of uppercase letters to the lowercase letters of a tweet text.
countPunctuation	Counts the total number of punctuations used in a tweet.
countNegWords	Represents the total number of negative words used by the user in the tweet shared.
countSwearWords	Total number of offensive words used by the user in a tweet.
countRetweet	Total how many times this specific tweet has been retweeted. This feature value can be extracted from the "retweet_count" attribute in a tweet object.
countFavorite	Total how many times this specific tweet has been liked by other users.
countWord	Total number of words in a tweet.
countCharacter	Total number of characters in a tweet.
sentimentScore	The value of this feature is achieved by performing sentiment analysis on the tweet text.
subjectivity	Subjectivity is related to emotion or personal feelings, opinion or beliefs in the text. It is like an indicator to point out whether a piece of text contains emotion or not.
word embedding features	We need to convert the text in a tweet into a form which can be fed to the predictive model. By word embedding, words are represented as the vectors of real numbers. Here we used GloVe's pre-trained word vector and took top 20 features for each word and the total of 280 features representation for each tweet.

C. *Text Preprocessing*

In natural language processing (NLP), lemmatization is a text normalization process and the word lemma point to the root or dictionary form of a word. Though it is quite similar to stemming with the common goal to remove the inflectional forms of a word, while stemming is a crude heuristic process which just cuts the last part of a word to perform text normalization and it does not care whether the resultant word is meaningful. On the other hand, lemmatization just discard the inflectional endings of a word and returns it to its base form utilizing the vocabulary and morphological analysis of words [11]. As the context of the text is considered an important factor is case of rumor analysis and because of the working principle of stemming, the context may be lost, we decided to use only lemmatization as the text normalization process in our research work. In our project we will use NLTK (Natural Language Tool Kit) python library to perform lemmatization. Before lemmatization, we also had to remove URL, user mention, hashtag, punctuation and white space from the tweet text as a part of text preprocessing.

D. *Rumor Detection Model*

Next, our task is to develop the rumor detection model. According to the proposed framework, we supposed to get our dataset in a suitable format after performing text preprocessing, feature selection & extraction steps for further rumor detection mechanism. The rumor detection model is depicted in Fig.1. We have checked the applicability of the Artificial Neural Network (ANN) and k-Nearest Neighbor algorithm besides the most popular Support Vector Machine (SVM), Logistic Regression, and Random Forest algorithms. We all know that classification Accuracy is self-explanatory and widely used performance measure in any classification task. As the dataset we are considering is not balanced, so considering only classification accuracy is not enough. Besides, we will also consider Precision, Recall, F-Score, Mean AUC Score with 10-fold Cross-Validation. To further validate these scores, we will calculate the MCC Score (Matthew's Correlation Coefficient), and draw Confusion Matrix and ROC Curve.

However, this small amount of data can be used as testing data in our proposed framework and help them detecting rumor keeping huge training data secured from leaking to them or any other third party in the communication channel between the authorized organization and the developer group. Fig.2 shows our proposed framework to solve this problem. In this architecture, there are two planes, User Plane and Developer Plane. These planes are divided based on the access and authority of data, and scope of work. The developer plane again divided into two modules, Module-1 and Module-2. Here, Module-2 of the developer plane is actually our proposed rumor detection architecture.

The purpose of Module-1 in Developer Plane is to secure the training data from unauthorized access using RSA ((Rivest, Shamir, and Adleman)) cryptography algorithm, still making the model portable and sharing with other end users like the law enforcement organization or any other organization who want to work on detection and debunking of rumor. The core reason behind choosing the RSA algorithms is firstly, it is a public-key or asymmetric cryptosystem generating a pair of keys named Public Key and Private Key, and knowledge about one key does not guarantee to estimate the other key very quickly. Secondly, it is capable of handling almost all known password attacks. That is why RSA is widely used for digital signature, key exchange, and encryption of small data block in hundreds of software nowadays. Thirdly, it is easily implementable in Python. Most importantly, in our experiment, we have used RSA cryptography with 1024 bits considering the available computing capabilities in the market. After that, we will generate the standalone EXE file and share it with the authorized third party interested in rumor detection. We have found that just running this EXE file rumor can be detected very easily. Even the authorized third party can keep their data in the raw format like JSON because necessary preprocessing and structuring of the data is done by the Module-2 of our proposed framework. Dividing the whole framework into two different planes also ensures the secrecy of the authorized third party's work and their data from the developer group.

DOI: http://dx.doi.org/10.24018/ejece.2020.4.4.219

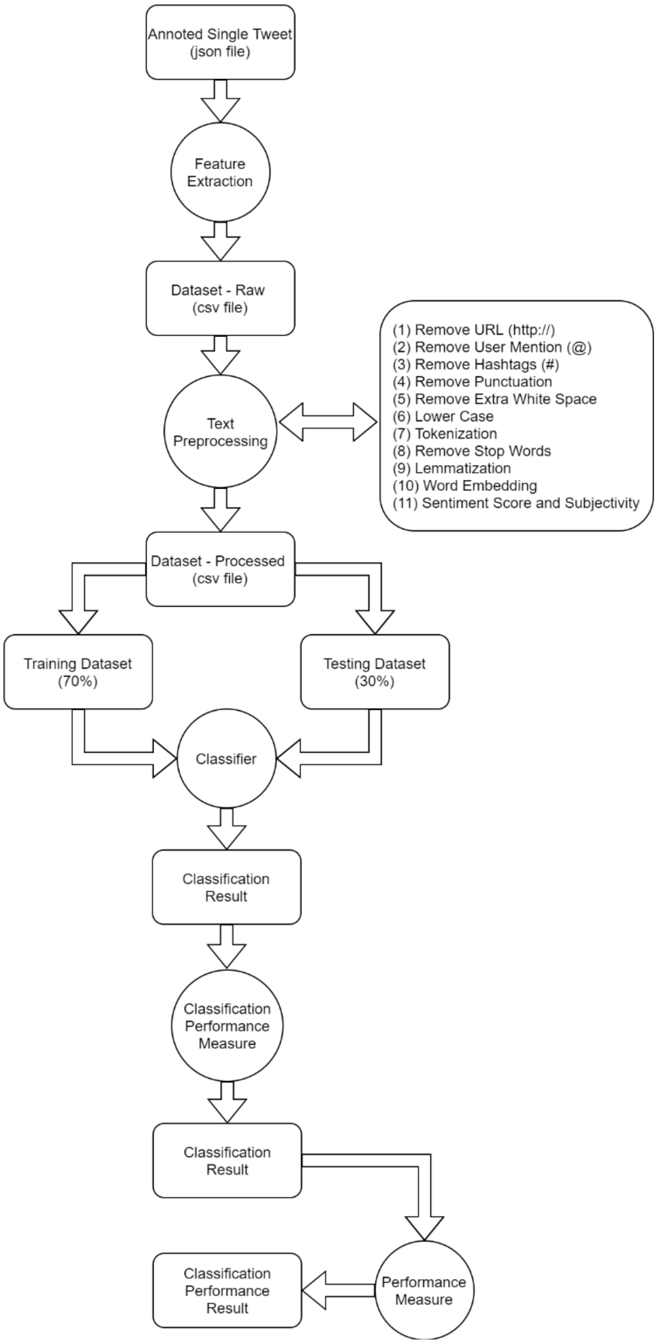


Fig.1. Proposed Rumor Detection Framework.

IV. Evaluation Methods

A. Accuracy

Though the Confusion Matrix representation looks absolutely perfect, it is always easier if we could find the same performance indication of the classifier by a single digit. That's why another performance metric was introduced, Accuracy. The mathematical formula of Accuracy is as follows,

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

A. Precision, Recall, F1-Score

Since Accuracy treats every class equally important so it is not a good measure to check the performance of a classifier trained by an imbalanced dataset. To solve this problem, Precision and Recall are introduced where the correct prediction of one class is given priority over the other one. The formal mathematical expression for computing Precision and Recall are as follows,

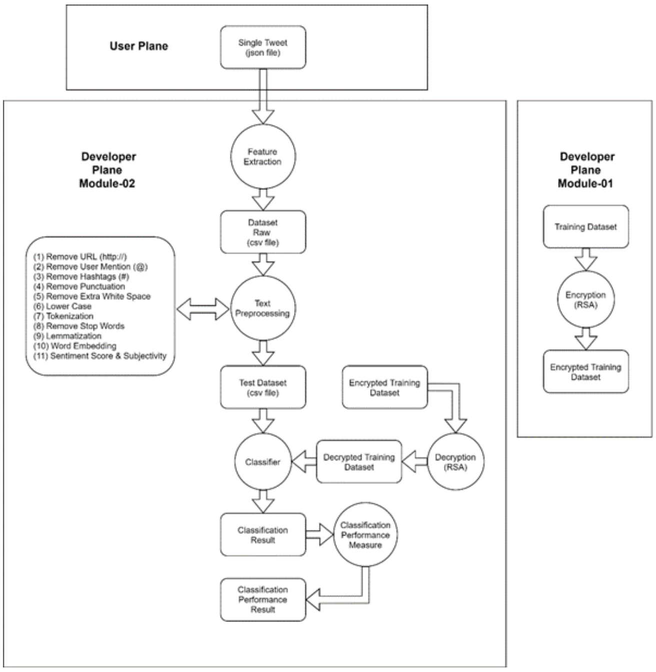


Fig.2. Framework for Securing Training Data.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} = \frac{True\ Positive}{Total\ Predicted\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} = \frac{True\ Positive}{Total\ Actual\ Positive}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

From the expressions above we can say that Precision identifies how accurate the model predicts the positive class If the Precision increases, the number of false positive error decreases. Besides, the Recall measures the fraction of positive examples correctly predicted. If classifier achieves higher Recall, we can assume that the amount of False Negative must be very less. F1 Score is the harmonic mean between the Precision and the Recall. The harmonic mean of two numbers tends to be closer to the smaller of the two numbers. Therefore, a high value of F1 means both the Precision and Recall have good result.

B. Receiver Operating Characteristics (ROC)

The ROC curve is another powerful visualization tool to represent the performance of a binary classifier. The ROC

DOI: http://dx.doi.org/10.24018/ejece.2020.4.4.219

curve shows the tradeoff between the TPR (True Positive Rate) and FPR (False Positive Rate). Generally, the more area under the curve, the better the performance of the classifier. ROC curve is used widely to show the performance difference of the classifiers used for a specific problem.

C. *Cross-Validation*

Also known as k-fold Cross-validation is a resampling process to divide the dataset into a specific number of equal-sized blocks (k). Then every block of data is used for testing in turn, and other blocks are used for training the model. When the dataset is relatively small, then we can use this process to get a better idea about a machine learning model performance. The total error from this process is estimated by summing up the errors contributed by all the runs.

D. *Matthews Correlation Coefficient*

Matthews Correlation Coefficient, MCC, in short, is also a powerful performance measure for binary classifications, and some scientists think this is the most informative single score performance measure in this case. This is also another approach to simplify the confusion matrix into a single value performance indicator. The value of MCC lies between −1 and +1, where +1 indicates the best prediction, and −1 indicates inverse prediction.

V. Experimental Results

Unlike other researches, in our experiment, we have put comparatively less emphasis on user-based features and used only 5 of them with 20 content-based features and 280-word embedding features for every tweet. Analyzing different performance metric graphs in this section, we have found that the Random Forest algorithm performs the best (94%) in terms of accuracy. However, the performance of ANN is also very near to Random Forest with 91% accuracy. Considering the simplicity of our project, the accuracy level we achieved is relatively very high compared with other relevant research works [6] [12].

Now, if we look at Precision, Recall, and F1-Score by different classifiers, it is clearly visible that because of the randomness concept, Random Forest performed the best in all three measures. Other classifiers also performed up to the mark. Here, the F1-Score clearly justifies the other two measures, Precision & Recall.

As we already know that MCC is a reliable indicator of the quality of the prediction by the binary classifier. We also know from the above discussion that the more the MCC score is near to +1 a better prediction is assumed. In our experiment, MCC scores of all the classifiers are at a satisfactory level (>=0.6). Besides, ROC curve, confusion

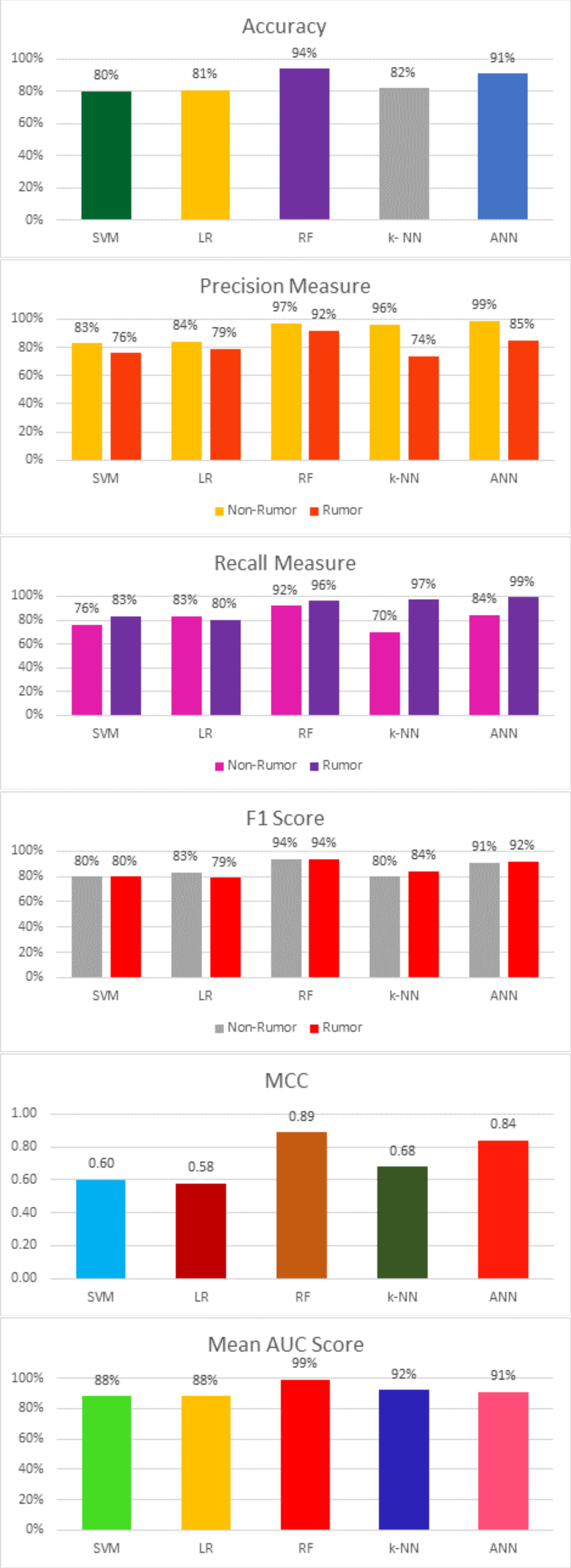


Fig. 3. Comparison of Accuracy, Precision, Recall, F1-Score, MCC, Mean AUC Score

DOI: http://dx.doi.org/10.24018/ejece.2020.4.4.219

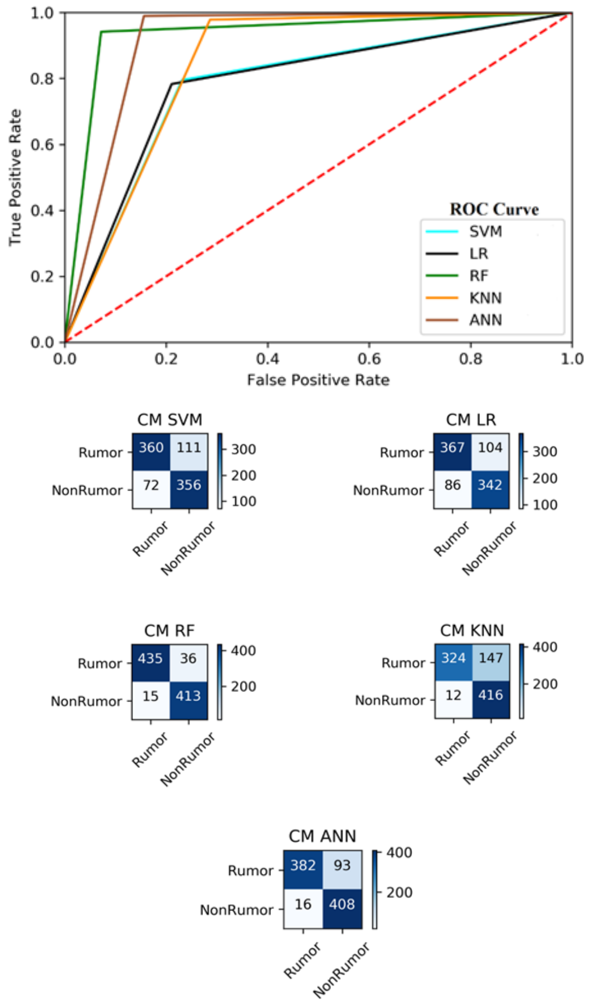
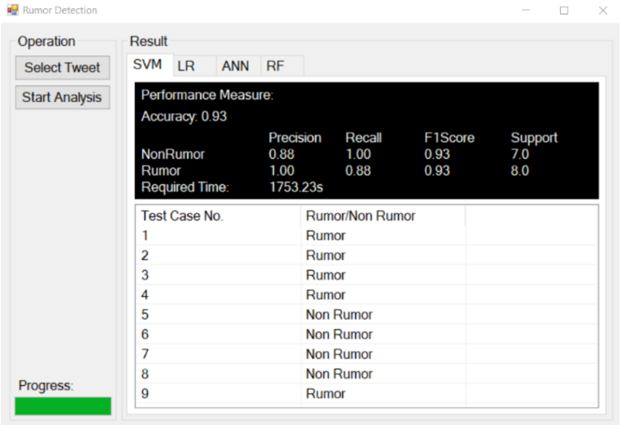


Fig. 4. ROC Curve, Confusion Matrix

matrix, Mean AUC Score (10-fold cross-validation) also supports our claim. So, we can conclude that our proposed framework of rumor detection is working absolutely fine with very good performance exhibited by Random Forest and ANN classifier. Fig.3 depicts the comparison among different classification algorithms, and Fig.4 shows ROC curve and confusion matrix in our experiment.



```
Performance Measure:
Accuracy: 0.93
                 Precision   Recall    F1Score    Support
NonRumor         0.88        1.00      0.93       7.0
Rumor            1.00        0.88      0.93       8.0
Required Time:   1753.23s
```

Test Case No.	Rumor/Non Rumor
1	Rumor
2	Rumor
3	Rumor
4	Rumor
5	Non Rumor
6	Non Rumor
7	Non Rumor
8	Non Rumor
9	Rumor

Fig. 5. Report from Portable Rumor Detection Framework

Beside the rumor detection task, another goal of this experiment is to make this model portable to authorized users maintaining the security of the training data. Fig.5 shows the rumor detection report generated by the proposed framework at the user end. The final software can also generate a CSV file containing rumor detection result. However, it can also show the result in real-time. From the generated report (Fig.5) it is evident that our proposed framework is working as expected. The maximum time recorded to get this report is approximately 1753 seconds or 29.21 minutes, as it involves RSA cryptography with 1024 bits.

VI. Conclusion

Before concluding our study, we would like to recall the introduction section, where we formulated some research questions. We can see that we confidently answered all the questions. To do that, firstly, we proposed and implemented a rumor detection framework to detect rumor is social media based on user-based and content-based features. After implementing our proposed framework, we got some tremendous results where the Random Forest classifier achieved 94% accuracy, and Artificial Neural Network achieved 91% accuracy. Not only that, the other three classifiers, SVM, Logistic Regression, and k-Nearest Neighbors, which are suggested by most of the researchers in this research area, also outperformed with more than 80% accuracy in our experiment. Besides, we also utilized various well-accepted performance measures to validate our results and, we can see that all the performance measures are justifying our model's accuracy level. Moreover, this is the first experiment to date, which considered the portability and availability of rumor detection framework to the authorized end users, keeping the training data secured with RSA cryptography algorithm. However, the execution time of 29.21 minutes, in this case, could be further reduced using sophisticated optimization techniques.

Rumor detection is not a recent technology hype, even a hundred year ago we can find the existence of various kinds of rumors in the society, and it is not overestimating that we will need rumor detection after 100 years from now as it is a continuous process and always open research area to explore application scopes of different technologies available.

References

[1] H. Shaban, "Twitter reveals its daily active user numbers for the first time," Washington Post, 7 February 2019. [Online]. Available: https://www.washingtonpost.com/technology/2019/02/ 07/twitter-reveals-its-daily-active-user-numbers-first-time/ ?noredirect=on&utm_term= .90f3e88c6abc. [Accessed 20 April 2019].

[2] A. Y. K. Chua and S. Banerjee, "Rumors and rumor corrections on Twitter: Studying message characteristics and opinion leadership," in *2018 4th International Conference on Information Management (ICIM)*, Oxford, UK, 2018.

[3] Z. Zhe, R. Paul and M. Qiaozhu, "Enquiring Minds: Early Detection of Rumors in Social Media from Enquiry Posts," in *Proceedings of the 24th International Conference on World Wide Web*, Florence, Italy, 2015.

[4] G. Liang, W. He, C. Xu, L. Chen and J. Zeng, "Rumor Identification in Microblogging Systems Based on Users' Behavior," *IEEE Transactions on Computational Social Systems,* vol. 2, no. 3, pp. 99-108, 2015.

[5] F. Chierichetti, S. Lattanzi and Alessandro Panconesi, "Rumor Spreading in Social Networks," in *Automata, Languages and Programming, ICALP 2009. Lecture Notes in Computer Science*, vol.

5556, A. M. Y. M. S. N. W. T. Susanne Albers, Ed., Berlin, Heidelberg, Springer, 2009, pp. 375-386.

[6] A. Pal and A. Y. K. Chua, "Classification of rumors and counter-rumors," in *2018 4th International Conference on Information Management (ICIM)*, Oxford, UK, 2018.

[7] Y. Wang, J. Luo, R. Niemi, Y. Li and T. Hu, "Catching Fire via "Likes": Inferring Topic Preferences of Trump Followers on Twitter," *CoRR,* vol. abs/1603.03099, 2016.

[8] Z. Jin, J. Cao, H. Guo, Y. Zhang, Y. Wang and J. Luo, "Rumor Detection on Twitter Pertaining to the 2016 U.S. Presidential Election," *CoRR,* vol. abs/1701.06250, 2017.

[9] S. Krishnan and M. Chen, "Identifying Tweets with Fake News," in *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, Salt Lake City, Utah, USA , 2018.

[10] A. Zubiaga, M. Liakata and R. Procter, "Learning Reporting Dynamics during Breaking News for Rumour Detection in Social Media," *CoRR,* vol. abs/1610.07363, 2016.

[11] C. D. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, 1st ed., USA: Cambridge University Press, 2008, pp. 32-34.

[12] A. Vijeev, A. Mahapatra, A. Shyamkrishna and S. Murthy, "A Hybrid Approach to Rumour Detection in Microblogging Platforms," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Bangalore, Karnataka, India, 2018.

DOI: http://dx.doi.org/10.24018/ejece.2020.4.4.219