



# Fake Information Analysis and Detection on Pandemic in Twitter

J. Jeyasudha<sup>1</sup> · Prashnim Seth<sup>2</sup> · G. Usha<sup>3</sup> · Pranesh Tanna<sup>2</sup>

Received: 4 April 2022 / Accepted: 4 August 2022

© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2022

## Abstract

Twitter has become a popular platform to receive daily updates. The more the people rely on it, the more critical it becomes to get genuine information out. False information can easily be shared on Twitter, which influences people's feelings, especially if fake information is linked to COVID-19. Therefore, it is of utmost importance to detect fake information before it becomes uncontrollable. Real-time tweets were used as part of this study. A few features like tweet's text, sentiment etc., were extracted and analyzed. The project returns a set of statistics determining the tweet's veracity. In this study, various classifiers have been used to see which of them works best with the proposed model in classifying the used dataset. The proposed model achieved the best accuracy of 84.54% and the highest F1-score of 0.842 with Random Forest. With careful analysis while feature selection and using few features, the model developed is equivalent in performance to the other models that use a lot of features. This confirms that the model developed is less complex and highly dependable.

**Keywords** Twitter · Fake information · Machine learning · Random forest · Classification · Pandemic

## Introduction

The information flows have shifted to various media over time. In an era of digitization, information and events from around the world are transmitted primarily through online social media (OSN) such as Instagram, Twitter, Facebook and WhatsApp. While information travels across the globe in seconds, this transition poses a huge threat of false Information and misinformation being shared by some people. This false information causes chaos and panic.

Curbing the spread of such fake news has become the utmost priority for people with resources and knowledge. If there is no monitoring of fake news and spread of tweets containing false information, they may go viral and the

consequence would lead to devastating mental health impact on millions of people, who rely on sources like twitter for their news feed. It may not only stop at that but can also lead to a global rise of revolts and breakdown. This is why projects and research like these are important for overall well-being.

An enhanced model has to be developed that will be able to minimize the spread of fake information from Twitter and help all users' fears caused by the fake information about the pandemic. The fake Information about the infectious disease is identified and uses a model similar to those used for controlling a pandemic.

Real-world data make it clear that the spread of fake information can be detrimental for people, businesses, industries and many other facets of the society. The results of the proposed method can, thus, help solve some of the current global problems associated with the spread of false information.

This research will compare various machine learning methods for detecting fake information, analyze the difference between them by comparing the accuracies and determining which one gives the best results.

The focus here is on tweets related to COVID-19. The tweets considered here are real-time tweets scraped from twitter to increase the practicality of this project. The spread of fake news related to pandemic is a major concern as it

---

This article is part of the topical collection "Intelligent Systems" guest edited by Geetha Ganesan, Lalit Garg, Renu Dhir, Vijay Kumar and Manik Sharma.

---

✉ Prashnim Seth  
prashnimseth@gmail.com

<sup>1</sup> Department of Computational Intelligence, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

<sup>2</sup> Department of Software Engineering, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

<sup>3</sup> Department of Computational Technologies, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

has ramifications in factors like vaccination or following the norms of lockdown. Therefore, having COVID-19 as the focus is beneficial.

The outcome of this paper would benefit the public at large. Fake Information, as we know, can harm people in many ways and that is why we must detect and control the dissemination of false information on social networking sites. Thus, to stop the spread of false information, and to control this infodemic, we are developing a model to detect the false information.

## Literature Survey

In [1] the paper “Characterizing the Propagation of Situational Information in Social Media During COVID-19 Epidemic: A Case Study on Weibo”, the dataset contained 36,000 posts. They used content features—URL/Hashtag, publishing timing, length, reposted amount. Machine Learning Algorithms that were applied, were SVM, Naive Bayes, Random Forest. The algorithms used achieved an average accuracy of 85%. The paper was published in April 2020.

In [2] “Rumour Detection in Social Media with User Information Protection” by Md. Rashed Ibn Nawab, Kazi Md. Shahiduzzaman, Titya Eng, and Md Noor Jamal, published in July 2020, the dataset was classified into Rumors and Non-rumors. A total of 1972 rumor tweets and 3830 non-rumor tweets were collected. The models used were ANN, Random Forest, KNN, SVM, and Logistic Regression. The accuracies were Random Forest—94%, ANN—91%; SVM, Logistic Regression and KNN > 80%.

In [3] the paper “Identifying Tweets with Fake News”, the dataset size was of 10,698 tweets. There were a total of 13 content features and 5 user features. The models applied were the J48 decision tree classifier and SVM. The J48 decision tree classifier returned the highest accuracy of 80%.

In [4] “Identifying misinformation on Twitter with a support vector machine”, the analysis was done both, with and without the news content. There were a total of 22 attributes of the dataset. The models and their accuracy were Naïve Bayes (95.55%), neural network (97.09%), and support vector machine (98.15%). The authors were Supanya Aphiwongsophon and Prabhas Chongstitvatana and the paper was published on 27 March 2020.

In [5] “Detecting Fake News with Tweets’ Properties”: written by Ning Xin Nyow, Hui Na Chua in November 2019, the dataset size was 23,206. There were two types of attributes in the dataset, tweet specified and news specified. The accuracies were: RF algorithm (98.6%), DT algorithm (98.3%). The Recalls were: RF algorithm (95.4%) and DT algorithm (94.51%). The F1-scores were: RF algorithm (97.2%) and DT algorithm (93.82%).

In [6] “Detecting fake news from twitter”, the dataset size was 250, small dataset being one of the drawbacks of this paper. The features of this paper were User follower count, User friends count, User status count, Favorite count, Users profile URL, Retweet count and Length of tweet. The models and accuracy are Random Forest (74%), Logistic Regression (68%), and Decision Tree (67.6%).

In [7] “Comparative Analysis of Fake News Detection using Machine Learning and Deep Learning Techniques”, there were a total of 25,000 dataset, with 5000 test data. The features were divided as tweet and user features. Tweet features: Text, createdAt, retweetCount, favouriteCount, Source, Length. User features: userId, username, userCreatedAt, userDescription, userFollowers, userFriends, User status count, User verified, User profile contains a URL or not. The model and the accuracies are LSTM (93%), Keras Neural Network (90.3%), SVM (85%), and Naive Bayes (68%). The author is Simran Dabreo and the paper was published in April 2020.

In [8] “Fake News Detection/ Fake Buster”, the dataset size was 26k. Data were analyzed and presented in confusion matrix for all the models for clarity. Large dataset was used for more efficient accuracy derivation. The models used and their accuracies are Naïve Bayes: 72.94%, SVM: 88.42%, Neural Network with TF: 81.42%, Neural Network with Keras: 92.62%, and LSTM: 94.53%. The author is Vaishnavi R and the paper was published in May 2020.

In [9] “A Deep Neural Network for Fake News Detection”, the author of this paper is Nigel Fernandez. The Bert model was implemented with elaborated architecture and the Credit score was used to calculate the scale of authenticity of the text. The models applied here and their accuracies are Bag-of-words: 90.21%, Bag-of-words + TFIDF: 91.92%, Bag-of-ngrams: 91.41%, Bag-of-ngrams + TFIDF: 92.47%, SVM + Bigrams: 83.12% and it had an Accuracy Majority of 94.2%.

In [10] “Fake News Detection by Learning Convolution Filters through Contextualized Attention”, 12,836 short statements were taken from POLITIFACT. The implementation reveals that contextual attention helps in finding features. The model and their accuracies are Baseline: Binary classification (57.8%), 6-way Classification (23.1%). For Fake-Net: Binary classification (63.3%), 6-way Classification (24.9%). The author of this paper is Ekagra Ranjan and the paper was published in August 2019.

In [11] “FakeNewsNet: A Data Repository with News Content, Social Context and Spatiotemporal Information for Studying Fake News on Social Media” by Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee and Huan Liu, the authors of the paper have come up with a fake news repository called the FakeNewsNet using a system called the FakeNewsTracker, for applications on fake news research in social media. They also came up with a data analysis which

compared various user profiles and proved that the news posted by the bot users is more likely to be fake. The paper was published in June 2020.

In [12] the paper “Covid-19 fake news sentiment analysis” by Celestine Iwendi, Senthilkumar Mohan, Suleman Khan, Ebuka Ibeke, Ali Ahmadian, Tiziana Ciano, the authors use an information fusion process to obtain real news from various media sources. The dataset used had 586 true news gathered from sources like WHO, CDC, etc., and 578 fake news gathered from social media websites like Facebook. They used AdaBoost, Decision Tree, KNN, GRU, LSTM, and RNN and obtained an F1-Score of 84.04%, 80.20%, 76.85%, 88%, 86% and 88%, respectively. The paper was published in July 2022.

## Proposed Methodology

### Dataset

The tweets relating to the pandemic were collected using a tool called Twint. Keywords like coronavirus, covid19, coronavirusPandemic, etc. were used to collect the tweets about the pandemic. A total of 2500 real-time tweets were extracted using the tool. The data were then scrubbed and various tweets were removed as part of the process. The data were cleansed based on reasons like the language of the tweet, whether it was an advertisement or if the tweets were missing significant information.

The tweets were then manually labeled as fake or real based on the URLs, their source, other extra information provided in the tweets etc. The final corrected dataset contained 768 fake tweets and 749 real tweets.

### Feature Selection and Extraction

The following features were extracted to be used in the study:

1. User follower count
2. Tweet like count
3. Text of the tweet
4. Hashtags
5. URL
6. Polarity of the tweet (Obtained by sentiment analysis)
7. User favorites

### System Architecture

The system architecture in the Fig. 1 above describes the steps of the proposed methodology for this project. The figure illustrates the data collection, pre-processing and cleaning, feature extraction, machine learning and deep learning methods applied in a step-by-step format.

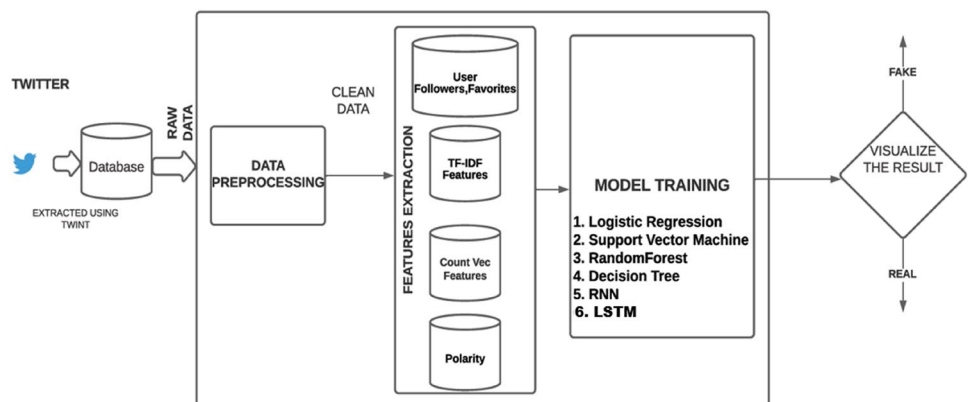
The architecture shows that the data are extracted from twitter. The raw data are then pre-processed and cleaned. The features mentioned are extracted and the model is trained accordingly. The machine learning methods stated in the diagram are applied on the train data. The test data are then used to test the model and visualize the results as real or fake.

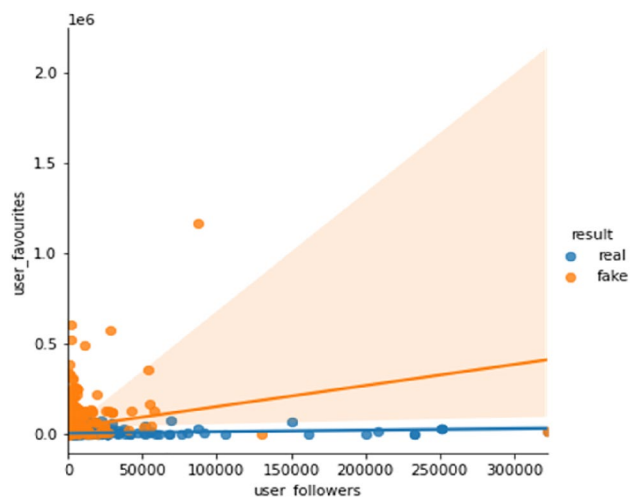
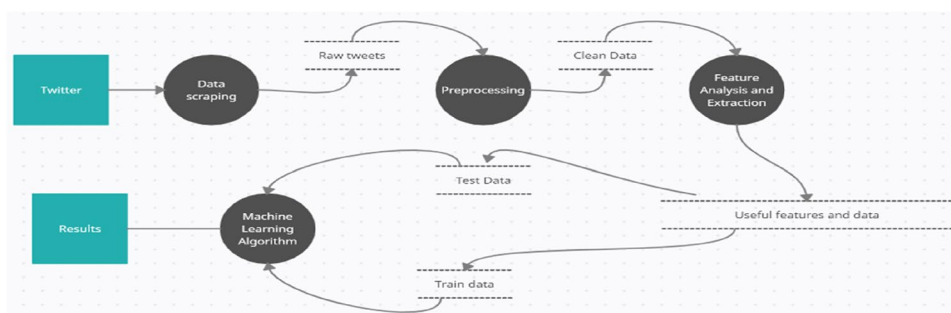
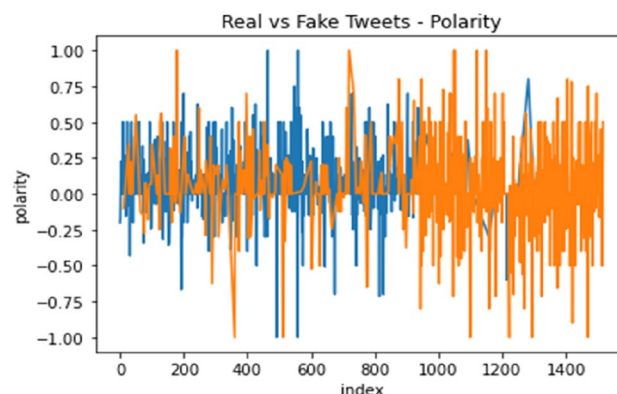
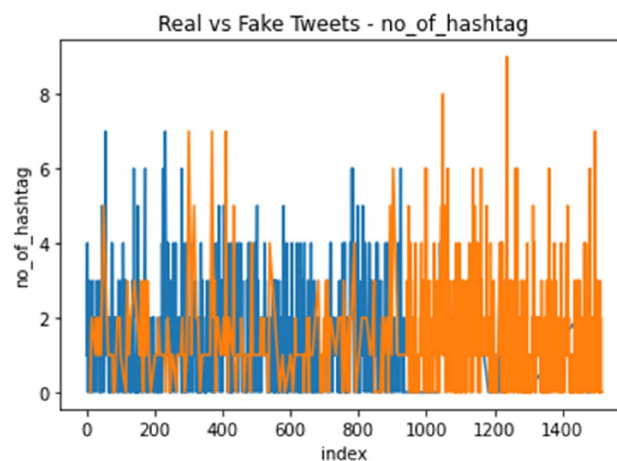
### Data Flow Design

The data flow diagram in Fig. 2 shows how the data flow in the system. It describes how the data are collected, pre-processed and classified as real or fake.

The diagram below shows that the raw real-time data are scrapped from twitter and cleaned using various pre-processing methods. The useful features are extracted to be used in the model. The data are then split into test and train. The model is built using the split data and gives the results.

**Fig. 1** System architecture of the proposed methodology



**Fig. 2** Data flow representation of the proposed methodology**Fig. 3** User followers /user favorites—real vs fake tweets**Fig. 4** Polarity of real tweets and fake tweets**Fig. 5** Number of hashtags in real vs fake tweets

## Implementation

### Data Analysis

On analyzing the features of the dataset, it was found that there was a relation between the user followers and the likes of the tweet also known as user favorites. Figure 3 below shows the variation.

Another difference was found when analyzing the polarity of the tweets. There was a variation with the fake and real tweets. Figure 4 shows the variation in the real and fake tweets on analyzing the polarity.

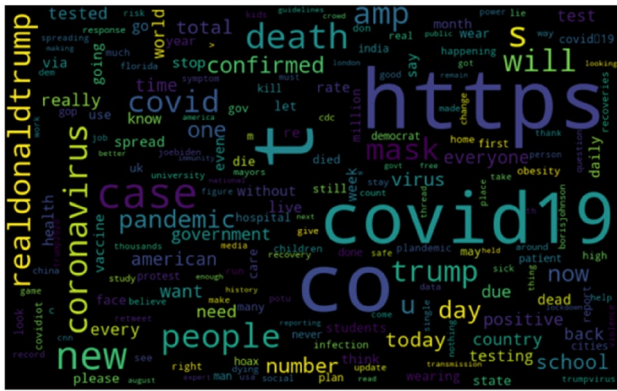
Figure 5 shows that the Number of Hashtags in real vs fake tweets has a significant variation between the true and fake tweets. The feature, hence, was used in the model for classification.

A few other features were also analyzed which did not show any significant difference between the real and fake tweets and therefore were not used while performing the classification using the algorithms.

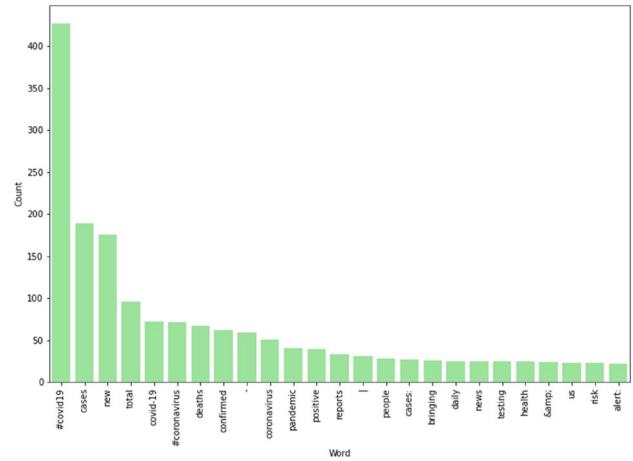
Figures 6 and 7 are WordCloud and word analysis of the two types of tweets which proves that this feature was unusable in the study.

Figures 8 and 9 show the most frequent words in real and fake tweets. It can be seen that there is no significant difference between the most frequent words used in real and fake tweets.

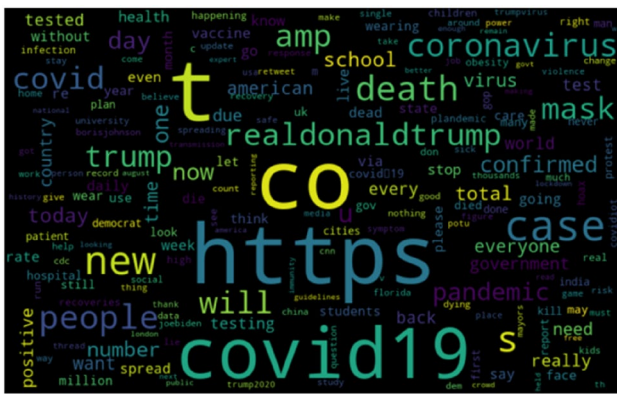




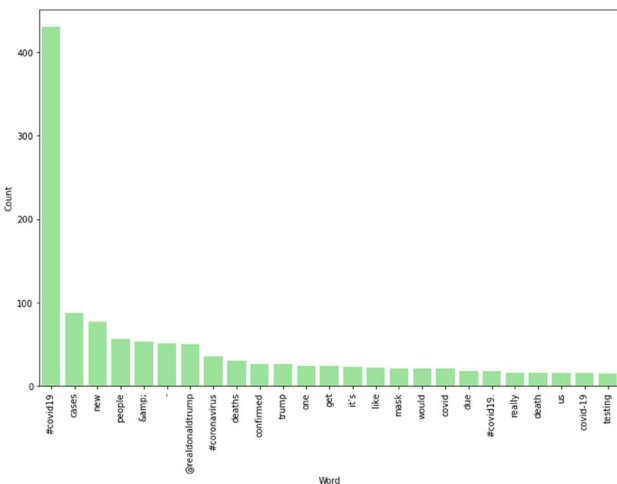
**Fig. 6** WordCloud for fake tweets



**Fig. 9** Most frequent words used in real tweets

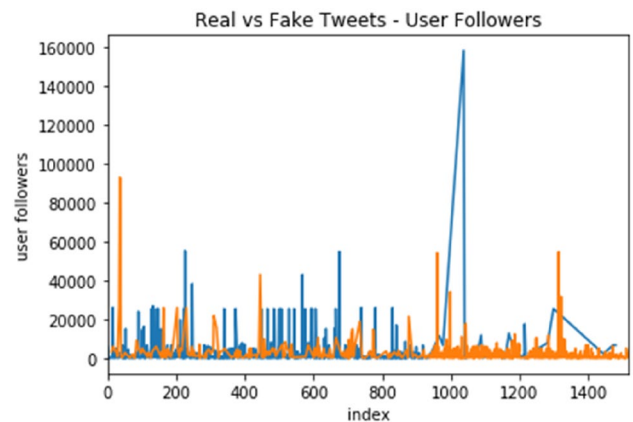


**Fig. 7** WordCloud for real tweets



**Fig. 8** Most frequent words used in fake tweets

Figure 10 is a comparison between the real tweets and the fake tweets for the User Followers feature. On analysis, it was observed, that when this feature was used separately it



**Fig. 10** User followers for fake tweets vs true tweets

is not very useful. But, when the User Followers feature is viewed combinedly with the User Favorites feature, a significant difference between the real and fake tweets can be seen.

## Model Creation

The selected features are user\_followers upon user\_favorites, the polarity of the tweets based on the sentiment analysis done and the hashtags in the tweets.

After the features were selected, the proposed model is created for classification. Using PCA (Principal Component Analysis) for dimensionality-reduction and Min-Max Scaler for normalizing the input features, the dimensions of the data and features were adjusted.

The text-based features are first tokenized by applying the TF-IDF vectorizer to the data. WordCloud is used to for defining a vocabulary of words for the model. Countvectorizer is applied to obtain the frequency of each word in the

labeled text. The categorical features are then encoded so that they can be used along with the numerical features.

The test and train data are then classified into 30% and 70%, respectively, for implementation. Now the training and the test data are passed through the various machine learning and deep learning algorithms as explained in the next section.

## Machine Learning and Deep Learning Algorithms used for Classification

Machine learning is used in a wide variety of applications. Classifying information as real or fake is one of the applications of machine learning.

Detecting misinformation is said to be a binary classification task. Many machine learning methods are suitable for classification.

The following classification methods were used and compared against each other:

1. **Logistic regression:** Logistic Regression is a supervised machine learning algorithm which is used to calculate and predict the probability of a target value. It is used for binary classification.

It predicts a dependent data variable by analyzing its relationship with one or more existing independent variables. The sigmoid function is used to calculate the probability in logistic regression. The logistic function is a simple S-shaped curve used to convert data into a value between 0 and 1.

2. **Random Forest:** Random Forest is also a supervised machine learning algorithm which is used for classification.

It uses an ensemble of Decision Trees, using bagging and feature randomness to build each tree.

3. **Decision Tree:** The Decision Tree is a supervised machine learning algorithm where the training data is continuously split into smaller subsets based on a certain parameter. A decision tree covering the training set is returned by the algorithm.
4. **SVM:** SVM or Support Vector Machine performs classification by finding the hyperplane that maximizes the margin between the two classes with the help of support vectors.
5. **RNN:** RNN or Recurrent Neural Networks is a type of Neural Network Algorithm where the output of the previous step is fed into the next step.

The Keras sequential model is used to keep the model simple. Three dense layers were added with Relu activation and dropout layers in between them with a dropout rate of 0.7 to avoid overfitting. The last dense layer uses the sigmoid activation parameter.

The adam optimizer is used in the model. Early stopping is defined where the validation loss is monitored during the 100 epochs which were included in the model to avoid underfitting.

6. **LSTM:** LSTM or Long Short-Term Memory is also a type of Neural Network Algorithm capable of learning order dependence in sequence prediction problems.

The Keras sequential model is used to keep the model simple. The dropout here is set as 0.3 to avoid overfitting and 100 neurons are added in the LSTM layer. The epochs are set to 10.

## Verification and Validation

There are three main modules in this project.

- Data extraction
- Pre-processing
- Model implementation

*Data Extraction:* As mentioned earlier, the data were extracted using Twint. The main thing to be checked is that the label and do the following:

- Extract the tweets based on the labels entered.
- Extract the number of columns and user information based on the features included.
- Extract the rows of the dataset based on the limit if set.

By this, it can be confirmed that the dataset extracted is what was intended to extract. It also checks if the data have all the features that are necessary for the implementation.

*Pre-processing:* Here, the data are pre-processed and cleaned to prepare it to be applied in the proposed model for implementation and so it does not give out unnecessary errors while implementing the model. The following points were checked:

- Check that there are no null values or unclear text in the dataset.
- Derive the required features from the existing features.
- Remove the unnecessary features/columns from the dataset.

This ensure that there are no errors in the dataset and it can be used for implementation.

**Model Implementation:** Finally, the model is implemented using the pre-processed dataset. The Machine Learning methods used in this project are Logistic Regression, Random Forest, Decision Tree, SVM, LSTM and RNN.

The methods are properly applied with the model and give results. We get the accuracy, F1-score, recall and precision of the applied methods.

## Experimentation Result and Analysis

To determine the efficiency of the model the following evaluation methods are considered:

### Accuracy

Accuracy is the percentage of correct predictions made in the test data.

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

### Precision

Precision defines the fraction of true positives among all the predicted positives.

$$\begin{aligned} \text{Precision} &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \\ &= \frac{\text{True Positive}}{\text{Total Predicted Positive}}. \end{aligned}$$

### Recall

Recall defines the fraction of true positives predicted to all of the examples that are truly in the positive class

$$\begin{aligned} \text{Recall} &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \\ &= \frac{\text{True Positive}}{\text{Total Actual Positive}}. \end{aligned}$$

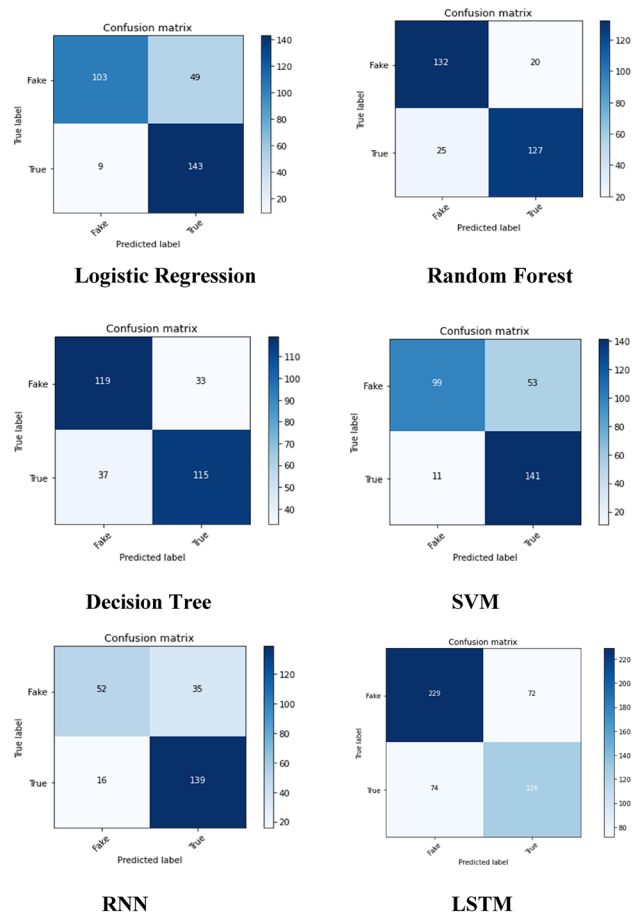
### F1-Score

The F1-score is the harmonic mean of precision and recall of a classifier. It is used to compare the performance of classifiers.

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

These metrics were used to determine and compare the efficiency of each method used. Of all the methods, Random Forest gave the best performance with an accuracy of 85.2% and an F1-score 0.849. Then confusion matrices were plotted for each method as shown below.

## Confusion Matrices



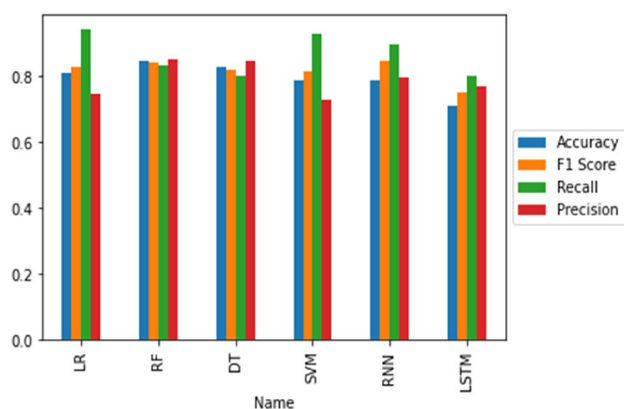
## Results

The Table 1 above presents the results of the developed model. It compares the accuracies, precisions, recalls and F1-Scores of the model when applied with different machine learning and deep learning methods.

From the Table 1, it can be seen that Logistic Regression has an accuracy of 80.92%, precision of 0.744, recall of 0.941 and F1-score of 0.831. Random Forest has an accuracy 85.2%, precision 0.864, recall 0.836 and F1-score 0.849. Decision Tree when applied gave an accuracy 76.97%, precision 0.777, recall 0.757 and F1-score 0.767. On applying SVM the accuracy is 78.95%, precision is 0.727, recall is 0.928 and F1-score is 0.815. RNN has an accuracy of 79%, precision of 0.80, recall of 0.90 and F1-score of 0.84. Using LSTM, we get an accuracy of 71%, recall 0.75, precision 0.80 and F1-Score of 0.77.

**Table 1** Result analysis table

	Accuracy	Precision	Recall	F1-Score
Logistic regression	80.92%	0.744	0.941	0.831
Random forest	85.2%	0.864	0.836	0.849
Decision tree	76.97%	0.777	0.757	0.767
SVM	78.95%	0.727	0.928	0.815
RNN	79%	0.80	0.90	0.84
LSTM	71%	0.75	0.80	0.77

**Fig. 11** Comparison of all methods when applied to the model

It can be observed that Random Forest gives the highest accuracy of 85.2% and a good F1-score of 0.849. RNN, also performs well with its F1-Score being 0.84.

Figure 11 above gives a graphical comparison of the results in Table 1 and from this, it can be observed that Random Forest and RNN work best for the proposed model.

Figures 12, 13, 14 and 15 are the comparisons between the different methods applied to the model in terms of accuracy, F1-score, recall and precision, respectively.

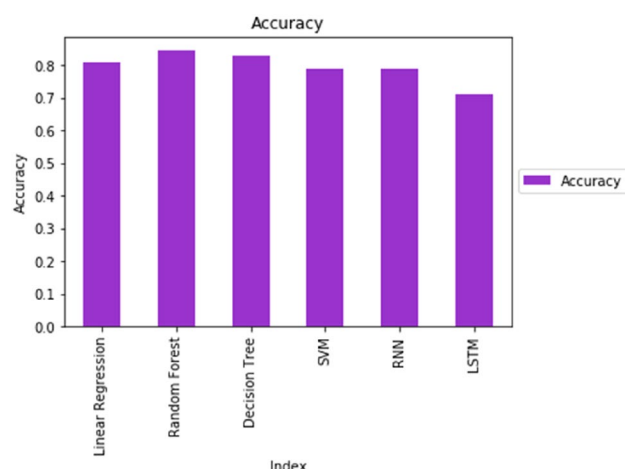
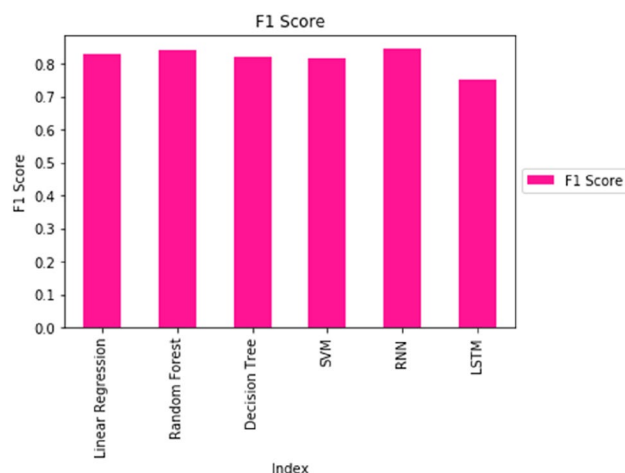
Figure 12 shows that Random Forest has the highest accuracy of 85.2%.

From Fig. 13, it can be seen that Random Forest has the highest F1-Score of 0.849 and RNN comes very close with 0.84.

Figure 14 shows the recall of Logistic Regression is the highest with the value being 0.941.

From Fig. 15, it can be observed that the precision of Random Forest 0.864 is the highest.

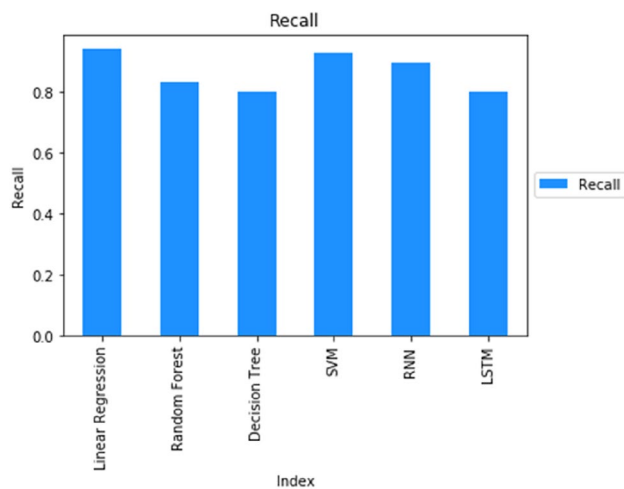
The ROC curves of the two best performing methods were compared to determine the method that works best with the model. From Figs. 16 and 17—the two ROC curves of the two best methods Random Forest and RNN, respectively, it is clear that the proposed model works well and Random Forest works best for the proposed model here in detecting fake information on twitter.

**Fig. 12** Comparison of accuracies of all methods when applied to the model**Fig. 13** Comparison of F1-Scores of all methods when applied to the model

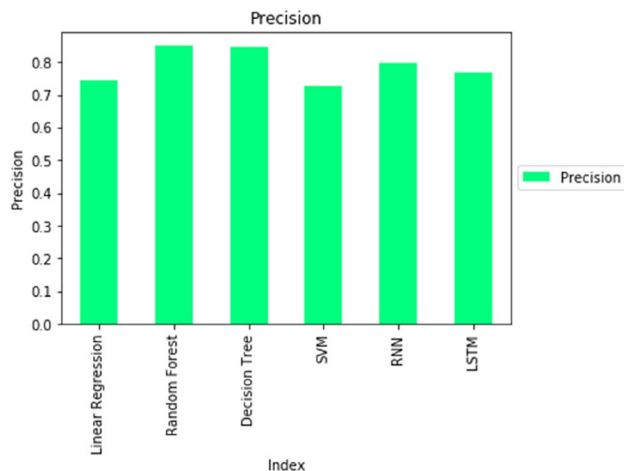
## Conclusion and Future Work

The research presented here proposes a method to classify a tweet as real or fake based on basic features like the tweet hashtags, URLs included, sentiment, popularity of the tweet and other features mentioned in the paper. Multiple machine learning and deep learning algorithms are used for comparison to determine the best one for the model. The classification model performance results show the effectiveness of the model applied by just using the few said features. The model developed here uses only a few features and still is on par with the other developed models which use a lot of features.





**Fig. 14** Comparison of recall of all methods when applied to the model



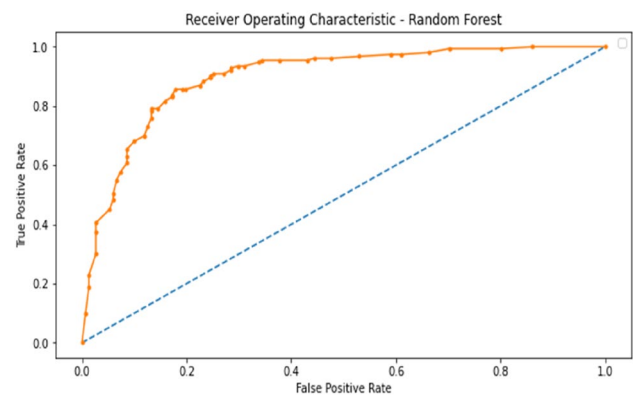
**Fig. 15** Comparison of precision of all methods when applied to the model

According to the results, it can be concluded that the Random Forest classifier best classifies the tweets in comparison to the other machine learning algorithms used in this project.

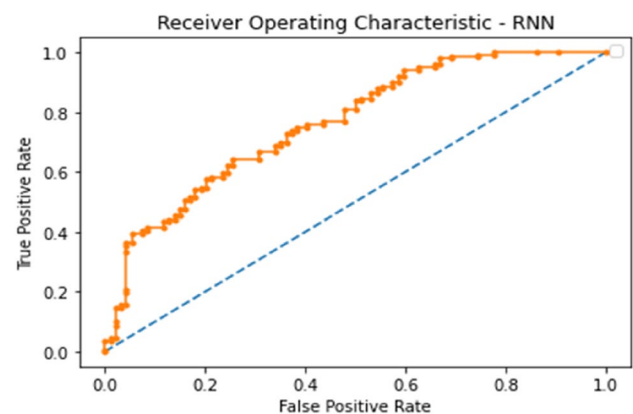
On analyzing the results and in comparison, to other researches, it can be concluded that the model is much less complex and reliable taking into account the real-time data.

This paper is still under development and we are trying to improve the effectiveness of the system. In future research, we will improve the method by including sources from known journals or Google search results, to process early news and views, which have no similarities in the Twitter network.

**Funding** Not applicable.



**Fig. 16** ROC curve for random forest



**Fig. 17** ROC Curve for RNN

## Declarations

**Conflict of Interest** The authors declare that they have no conflict of interest.

## References

1. Li L, et al. Characterizing the propagation of situational information in social media during COVID-19 epidemic: a case study on weibo. *IEEE Trans Comput Social Syst.* 2020;7(2):556–62.
2. Rashed Ibn Nawab M et al. Rumour detection in social media with user information protection. *EJECE* 2020;4(4).
3. Krishnan S, Chen M. Identifying tweets with fake information. In: *Proceedings of international conference on information reuse and integration*; 2018 Jul 6–9. Salt Lake City, USA: IEEE; 2018. p. 460–4.
4. Aphiwongsophon S, Chongstitvatana P. Identifying misinformation on Twitter with a support vector machine. *EASR*, Mar. 2020 Sheryl Mathias, detecting fake news from twitter, Feb. 2018.
5. Nyow NX, Chua HN. Detecting fake news with tweets' properties. *IEEE AINS.*, Nov. 2019.
6. Mathias S. Detecting fake news from twitter, Feb. 2018.
7. Dabreo S. Comparative analysis of fake news detection using machine learning and deep learning techniques, Apr. 2020.

8. Vaishnavi R. Fake news detection/fake buster, May 2020.
9. Fernandez N. A deep neural network for fake news detection, Nov. 2017.
10. Ranjan E. Fake news detection by learning convolution filters through contextualized attention, Aug. 2019.
11. Shu K, Mahudeswaran D, Wang S, Lee D, Liu H. Fakenewsnet: a data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data*. 2020;8(3):1–5.
12. Iwendi C, Mohan S, Khan S, Ibeke E, Ahmadian A, Ciano T. Covid-19 fake news sentiment analysis. *Comput Electr Eng*. 2022;101:107967. <https://doi.org/10.1016/j.compeleceng.2022.107967>.

**107967. Epub 2022 Apr 22. PMID: 35474674; PMCID: PMC9023343.**

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.