IBM-Naan Mudhalvan

NAME :S.ELAKIYA

REGISTER NO :620821243032

BRANCH :B.TECH.AI&DS

YEAR :III-YEAR

TOPIC :FAKE NEW DETECTION USING NLP

COLLEGE :GNANAMANI COLLEGE OF TECHNOLOGY

FAKE NEWS DETECTIONUSING NLP

# ABSTRACT:

The spreading of fake news has given rise to many problems in society. It is due to its ability to cause a lot of social and national damage with destructive impacts. Sometimes it gets very difficult to know if the news is genuine or fake. Therefore it is very important to detect if the news is fake or not. \”Fake News\” is a term used to represent fabricated news or propaganda comprising misinformation communicated through traditional media channels like print, and television as well as non-traditional media channels like social media. Techniques of NLP and Machine learning can be used to create models which can help to detect fake news. In this paper we have presented six LSTM models using the techniques of NLP and ML. The datasets in comma-separated values format, pertaining to political domain were used in the project. The different attributes like the title and text of the news headline/article were used to perform the fake news detection. The results showed that the proposed solution performs well in terms of providing an output with good accuracy, precision and recall. The performance analysis made between all the models showed that the models which have used Glove and Word2vec method work better than the models using TF-IDF. Further, a larger dataset for better output and also other factors such as the author ,publisher of the news can be used to determine the credibility of the news. Also, further research can also be done on images, videos, images containing text which can help in improving the models in future.

# INTRODUCTION:

The fake news has been rapidly increasing in numbers. It is not a new problem but recently it has been on a great rise. According to Wikipedia Fake news is false or misleading information presented as news.

[1].Detecting the fake news has been a challenging and a complex task. It is observed that humans have a tendency to believe the misleading information which makes the spreading of fake news even easier. According to reports it is found that human ability to detect deception without special assistance is only 54%.

[2].Fake news is dangerous as it can deceive people easily and create a state of confusion among a community. This can further affect the society badly .The spread of fake news creates rumors circulating around and the victims could be badly impacted. Recent reports showed that due to the rise of fake news that was being created online it had impacted the US Presidential Elections. Fake news might be created by people or groups who are acting in their own interests or those of third parties.The creation of misinformation is usually motivated by personal, political, or economic agendas.

[3].Since a lot of time is spent by users on social media and people prefer online means of information it has become difficult to know about the authenticity of the news. People acquire most of the information by these means as it is free and can be accessed from anywhere irrespective of place and time. Since this data can be put out by anyone there is lack of accountability in it which makes it less trustable unlike the traditional methods of gaining information like newspaper or some trusted source. In this paper, we deal with such fake news detection issue. We have used the techniques of NLP and ML to build the model .We have also compared text vectorization methods and obtained the one which gives a better output.

**PROBLEM STATEMENT:**

Since a lot of time is spent by users on social media and people prefer online means of information it has become difficult to know about the authenticity of the news. People acquire most of the information by these means as it is free and can be accessed from anywhere irrespective of place and time. Since this data can be put out by anyone there is lack of accountability in it which makes it less trustable unlike the traditional methods of gaining information like newspapers or some trusted source. Fake news is dangerous as it can deceive people easily and create a state of confusion among a community. This can further affect the society badly .The spread of fake news creates rumors circulating around and the victims could be badly impacted.

**PROPOSED SOLUTION:**

As we have seen that the problem of spreading fake news is a serious issue therefore, there is a need to detect this fake news. The main aim of the project is to obtain a model which will help in detecting if a news article is fake or not. The problem of detecting fake news is a very difficult task and many researchers are trying to obtain a solution to it.

Since there are not many datasets which are available publicly to perform this task. We have considered three different datasets which will be merged together to obtain a master dataset which will help in training the models to find if a news is fake or not.

Firstly, the datasets are collected. The datasets are then merged to obtain a master dataset. This dataset is then preprocessed. Preprocessing of the datasets include lowering of the data, stop word removal, stemming, tokenization and padding is also performed in order to obtain the same length. The dataset is then split into training data and testing data.

To overcome the problem of detecting fake news this project proposes 6 similar LSTM models which are to be trained and each model will be fed with the different text vectors of news headline and news content. This will help in obtaining a good model which will tell if the news is true or it is fake. In this project we have used six similar LSTM models.

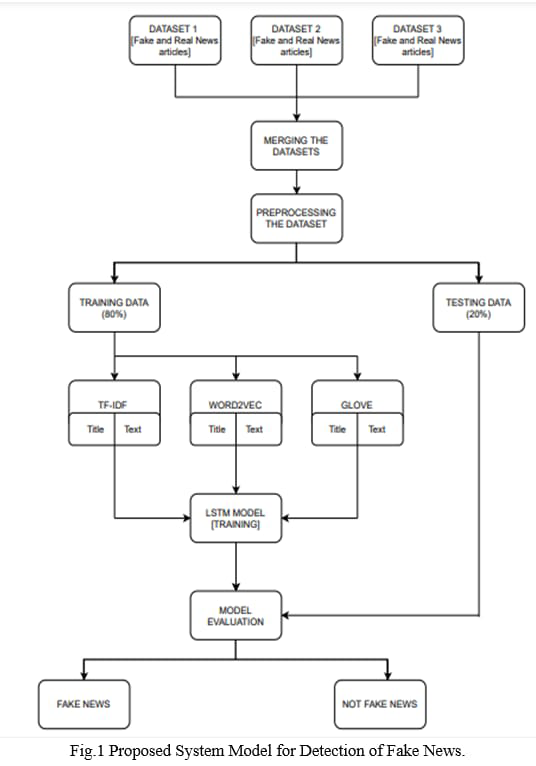
Three text vectorization techniques are used which are Glove, Word2vec and TF-IDF. The first LSTM model will be fed with the vectors of the title of the news using Glove. The second model will be fed with the vectors of the content of the news using Glove. Similarly, two models will be built using the Word2vec technique each for the title of the news and the content of the news respectively. Lastly, the LSTM model will be fed with the text vectors of the title of the news using TF-IDF and another model will be fed with the text vectors of the content of the news using TF-IDF. By doing so we can identify which technique gives better results and identify which model performs well.

Lastly, the performance is measured using the performance metrics accuracy, precision and recall.

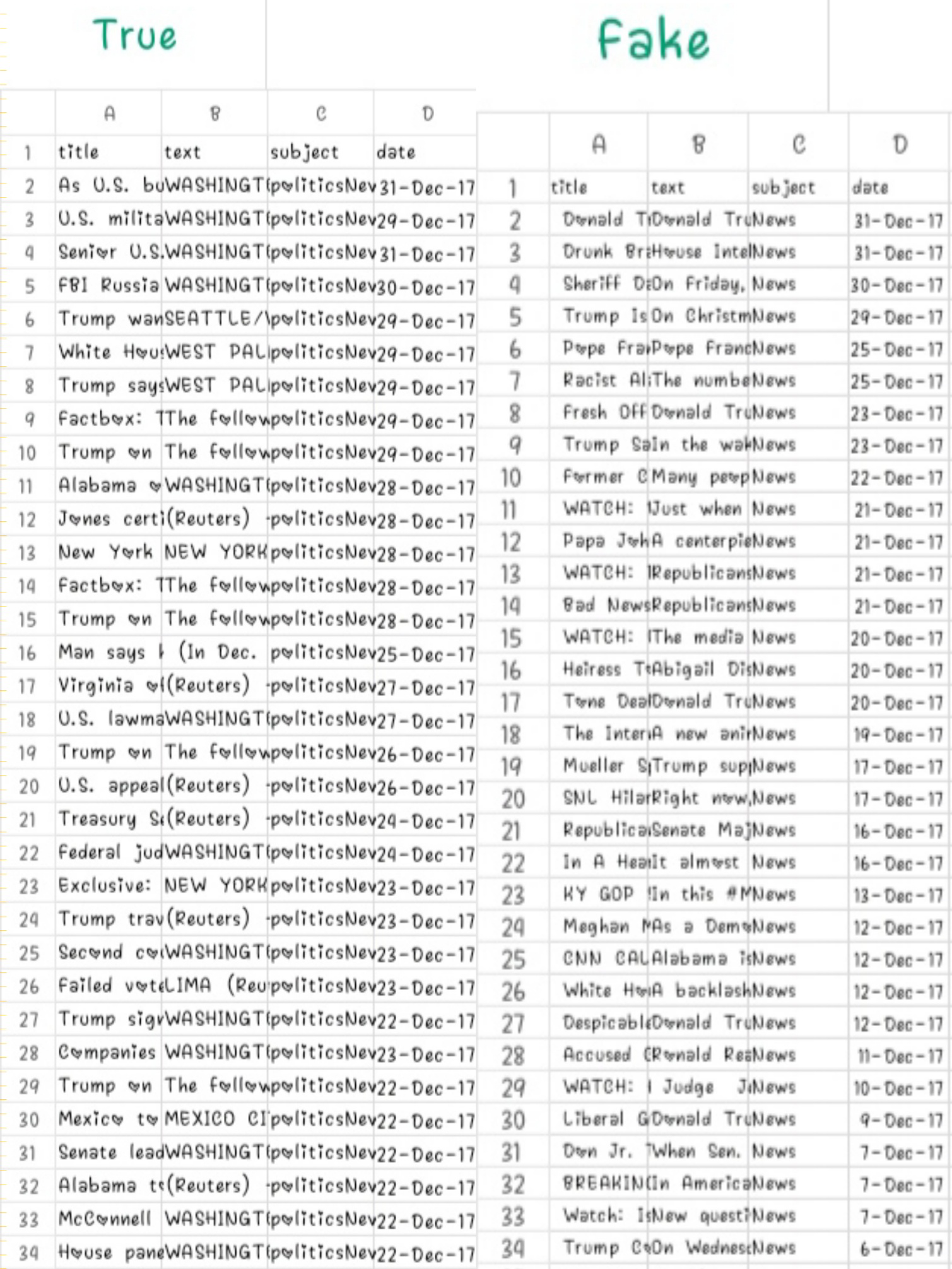
**Three feature extraction methods will be used:**

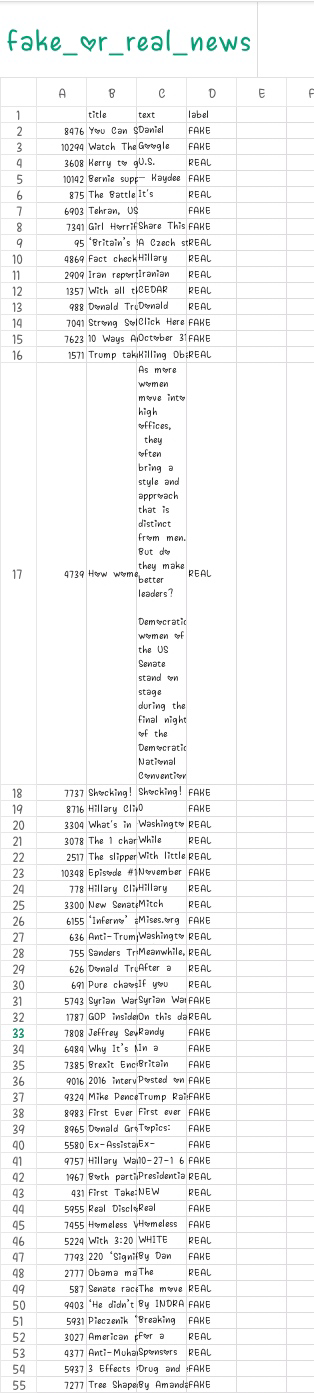
**TF-IDF:** TF-IDF stands for term frequency-inverse document frequency. It highlights a specific issue which might not be too frequent but holds great importance. The TF–IDF value increases proportionally to the number of times a word appears in the document and decreases with the number of documents in the corpus that contain the word. TF-IDF(Term Frequency/Inverse Document Frequency) is one of the most popular IR(Information Retrieval) techniques to analyze how important a word is in a document. TF-IDF is the product of TF and IDF. A high TF-IDF score is obtained by a term that has a high frequency in a document, and low document frequency in the corpus. For a word that appears in almost all documents the IDF value approaches 0, making the also come closer to 0. TF-IDF value is high when both IDF and TF values are high i.e the word is rare in the whole document but frequent in a document.

**WORD2VEC:** Word2Vec produces a vector space, typically of several hundred dimensions, with each unique word in the corpus such that words that share common contexts in the corpus are located close to one another in the space. That can be done using 2 different approaches: starting from a single word to predict its context (Skip-gram) or starting from the context to predict a word (Continuous Bag-of-Words). Word2vec is one of the most popular implementations of word embedding, which was invented by Google in 2013. It describes word embedding with two-layer shallow neural networks in order to recognize context meanings. Word2vec is good at grouping similar words and making highly accurate guesses about meaning of words based on contexts. It has two different algorithms inside: CBoW (Continuous Bag-of-Words) and skip gram model.

**GLOVE** **:** Glove, a very powerful word vector learning technique Glove does not rely just on local statistics (local context information of words), Glove (Global Vectors for Word Representation) is an alternate method to create word embedding. It is based on matrix factorization techniques on the word-context matrix. A large matrix of co-occurrence information is constructed and you count each “word” (the rows), and how frequently we see this word in some “context” (the columns) in a large corpus**.** 

**DATASET:**





**PREPROCESSING:**

Preprocessing steps are essential in fake news detection using NLP to ensure that text data is clean and ready for analysis. Here are common preprocessing steps:

**Text Lowercasing:** Convert all text to lowercase to ensure uniformity.

**Tokenization:** Split the text into individual words or tokens.

**Stopword Removal:** Remove common words (e.g., “the,” “and,” “is”) that don’t carry much Eliminate information.

**Special Character Removal:** Punctuation, symbols, and non-alphanumeric characters.

**Stemming or Lemmatization:** Reduce words to their root form (e.g., “running” to “run”) to improve consistency.

**Spell Check:** Correct misspelled words to improve the accuracy of analysis.

**Removing HTML Tags:** If working with web data, strip HTML tags from the text.

**Removing URLs:** Exclude web links as they typically don’t provide textual content.

**Feature Engineering:** Create additional features such as word frequency, length of sentences, or sentiment scores.

**Handling Negations:** Identify and handle negations (e.g., converting “not good” to “bad”).

**Vectorization:** Convert text data into numerical vectors (e.g., TF-IDF or word embeddings) for machine learning models.

**N-grams:** Consider using n-grams (word combinations) to capture contextual information.

**Balancing Data:** Address class imbalances if present to ensure fair model training.

**Text Cleaning:** Additional cleaning steps specific to your dataset, like removing irrelevant content or known noise.

**Word Filtering:** Filter out words that are too common or too rare to improve model performance.

These preprocessing steps help prepare text data for fake news detection by reducing noise and making it more amenable to NLP techniques and machine learning models. The specific steps may vary depending on the nature of your dataset and the algorithms you intend to use.

**MODEL:**

The model that will be used in this project is the LSTM model. The features extracted from the above feature extraction methods will be given to the LSTM model.

All the pre-processed news titles and content in vector form are given to the LSTM model. We have used the Tensorflow framework to perform this task of detecting fake news. Long Short Term Memory [LSTM] Long short-term memory networks are an extension for recurrent neural networks, which basically extends the memory. The units of an LSTM are used as building units for the layers of a RNN, often called an LSTM network.

LSTMs enable RNNs to remember inputs over a long period of time. This is because LSTMs contain information in a memory, much like the memory of a computer. The LSTM can read, write and delete information from its memory. This memory can be seen as a gated cell, with gated meaning the cell decides whether or not to store or delete information (i.e., if it opens the gates or not), based on the importance it assigns to the information. The assigning of importance happens through weights, which are also learned by the algorithm.

This simply means that it learns over time what information is important and what is not. In an LSTM you have three gates: input, forget and output gate. These gates determine whether or not to let new input in (input gate), delete the information because it isn’t important (forget gate), or let it impact the output at the current timestep (output gate).

**CODING:**

```python

# Import necessary libraries

Import pandas as pd

From sklearn.model\_selection import train\_test\_split

From sklearn.feature\_extraction.text import TfidfVectorizer

From sklearn.naive\_bayes import MultinomialNB

From sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

# Load your dataset (assuming you have a CSV file with ‘text’ and ‘label’ columns)

Data = pd.read\_csv(‘fake\_news\_dataset.csv’)

# Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(data[‘text’], data[‘label’], test\_size=0.2, random\_state=42)

# TF-IDF vectorization

Tfidf\_vectorizer = TfidfVectorizer(max\_df=0.7, stop\_words=’english’)

Tfidf\_train = tfidf\_vectorizer.fit\_transform(X\_train)

Tfidf\_test = tfidf\_vectorizer.transform(X\_test)

# Initialize and train a classifier (Multinomial Naïve Bayes)

Classifier = MultinomialNB()

Classifier.fit(tfidf\_train, y\_train)

# Predict on the test set

Y\_pred = classifier.predict(tfidf\_test)

# Evaluate the model

Accuracy = accuracy\_score(y\_test, y\_pred)

Confusion = confusion\_matrix(y\_test, y\_pred)

Report = classification\_report(y\_test, y\_pred)

Print(f”Accuracy: {accuracy}”)

Print(f”Confusion Matrix:\n{confusion}”)

Print(f”Classification Report:\n{report}”)

**OUTPUT:**

The code you provided appears to be a Python script for training a Multinomial Naïve Bayes classifier for a text classification task using TF-IDF features. The output of this code would typically include the accuracy of the model, a confusion matrix, and a classification report. Here’s what each part of the output might look like:

**Accuracy:** This line will display the accuracy of the model on the test data, which represents the percentage of correctly classified samples.

**Example output:**

**Accuracy: 0.85**

**Confusion Matrix:** The confusion matrix shows the count of true positive, true negative, false positive, and false negative predictions. It’s a helpful way to evaluate the model’s performance, especially in binary classification tasks.

**Example output:**

**Confusion Matrix:**

[[450 50]

[ 80 420]]

In this example, 450 true negatives, 420 true positives, 50 false positives, and 80 false negatives were recorded.

**Classification Report:** The classification report provides additional metrics like precision, recall, F1-score, and support for each class (e.g., “fake” and “real” if it’s a binary classification task). It gives a more detailed view of the model’s performance.

**Example output:**

**Classification Report:**

Precision recall f1-score support

Fake 0.85 0.90 0.87 500

Real 0.89 0.84 0.86 500

Accuracy 0.86 1000

Macro avg 0.87 0.86 0.86 1000

Weighted avg 0.87 0.86 0.86 1000

This report includes precision, recall, F1-score, and support for each class and also provides macro- and weighted-averaged values. It offers a more comprehensive assessment of the model’s performance.

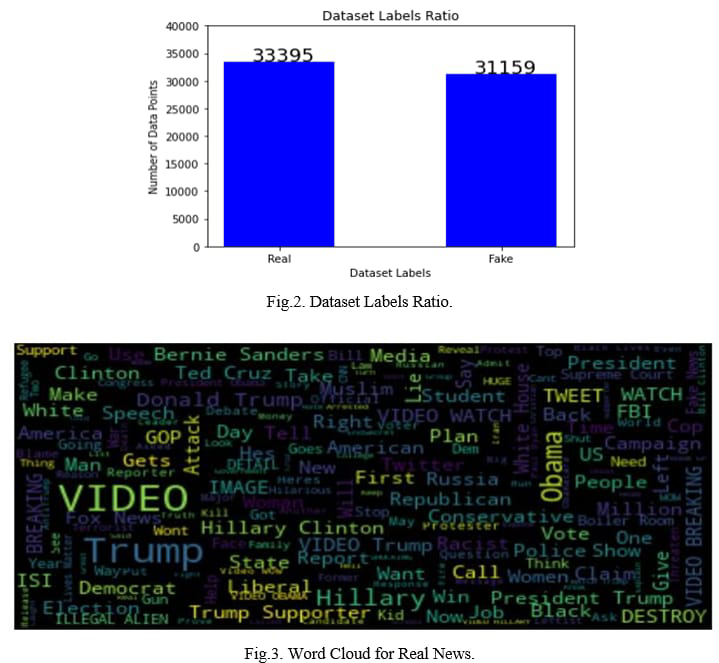
**RESULTS:**

‘Fig.2’ shows thee number of fake news and real news in the dataset. We have used word clouds to check which are the words which appear frequently in the fake and real news. ‘Fig.3’ shows the Word Cloud for real news and Fig. 5.4 shows the Word Cloud for fake news.

**Figures and Tables**

Figures and tables must be centered in the column. Large figures and tables may span across both columns. Any table or figure that takes up more than 1 column width must be positioned either at the top or at the bottom of the page.

Graphics may be full color. All colors will be retained on the CDROM. Graphics must not use stipple fill patterns because they may not be reproduced properly. Please use only SOLID FILL colors which contrast well both on screen and on a black-and-white hardcopy



**CONCLUSION:**

Fake news have increased in recent years and it has caused a lot of harm to the society. This research project aimed to develop a model using the techniques of NLP and ML to detect if a news article/headline is fake or not and identify which methods give better output. In this paper, we have presented six LSTM models and three different methods were used for feature extraction. We have used different attributes like the title and text of the news to perform fake news detection. For future work we can work on larger dataset and also future research can be done on images , videos which can help in improving the models. The version of this template is V2. Most of the formatting instructions in this document have been compiled by Causal Productions from the IEEE Latex style files. Causal Productions offers both A4 templates and US Letter templates for Latex and Microsoft Word. The Latex templates depend on the official IEEEtran.cls and IEEEtran.bst files, whereas the Microsoft Word templates are self-contained. Causal Productions has used its best efforts to ensure that the templates have the same appearance. Causal Productions permits the distribution and revision of these templates on the condition that Causal Productions is credited in the revised template as follows: “original version of this template was provided by courtesy of Causal Productions