Abstract

Fake news is common in today's digital age, raising significant challenges for individuals and society. In this research we developed a news detecting system that classify fake news articles using machine learning techniques. We explore a dataset of true and fake news articles by performing data preprocessing, exploratory analysis, and feature extraction for text classification. Various machine learning models are employed for text classification including traditional models like Logistic Regression, Naive Bayes, as well as ensemble methods such as Random Forest. Hyperparameters of these models are optimized using random search and evaluate their performance using metrics such as accuracy, precision, recall, and F1-score. Additionally, we investigate the efficacy of deep learning models, specially Long-short term memory (LSTM)and Convolutional Neural Networks (CNN) by observing impact of different batch sizes, Our findings showed that larger batch size lead to improved performance. Overall results showed effectiveness of machine learning models in distinguishing between fake and true news, with Random Forest demonstrated high accuracy in classification. This study contributes to the ongoing efforts to combat fake news by introducing an effective and accurate classification system. The findings of this analysis could be useful in developing automated systems for detecting and filtering fake news, resulting in a more informed and reliable news environment.

1 INTRODUCTION

News refers to information or current events that are covered by media outlets and journalists to educate the public about global happenings. It is a tool for educating people about regional, national, and international events in the fields of politics, economics, culture, science, and other fields. People started interacting news through online platforms rather than the traditional news organizations. Social media has emerged as the main channel for internet communication, approximately two-thirds of the population nowadays obtain their news through online sources, and percentage is still rising. Despite the advantages provided by social media, the quality of news is less compared to traditional news (Shu, 2017) and this intended to spread of fake news.

Fake news, often spread through social media platforms have significant impact on society which goes beyond private opinions; there are bigger consequences for society. Fake news can spread false information that may worsen social and political division, affect public opinion, and damage democratic systems. Furthermore, the quick spread of misleading information on social media platforms has the potential to deepen societal divisions.

Fake news not only provide source of spam, but it also has the potential to manipulate public perception and awareness (Allcott & Mathew Gentzkow). It damages the reputation of reliable journalism and decreases public confidence in traditional media sources. However, detecting misinformation is extremely important but also a technically challenging problem. It is difficult to distinguish between false and true information with a human eye. In one study, participants were asked to detect whether the article is fake, and results shows that somewhat 75% to 80% can find that the article is fake after having a hard time.

To control the spread of fake information several fact checking websites has been deployed to expose truth (e.g. snopes.com) and these websites need an expert analysis. Additionally with the help of Artificial intelligence and machine learning (ML) models researchers have developed a news detecting system capable of identifying fake news based on distinctive characteristics. These detective systems not only compacted the spread of fake news but also increased the trust in people who relay news on online platforms.

1.1 Problem Definition

Spread of false information has most impact on today's world, as its quite challenging to detect with human eye. Researchers came up with an automated detective system by using advance technology of Artificial intelligence and machine learning. These systems analyse linguistic and syntactic features extracted from data collected from various articles. Despite the advancement these system face challenges in analysing complex patterns within the data.

The data collected from articles often varies in context, style and some context may contain irony, satire, irrelevant and duplicate which effects the performance of the system. Failure in identifying these patterns results in poor performance of the system and fails to combat the spread of false information.

1.2 Proposed Solution

To address problem stated in problem definition, I propose a comprehensive approach that combines machine learning techniques and natural language processing (NLP) by analysing the trends. My approach is to use computational techniques to interpret patterns and signs of misleading data in text and to develop a system to combat the spread of misinformation, safeguarding public trust and democratic processes.

1.3 Aims and Objectives

Aim:
To develop a news detection system that uses machine learning and Natural language processing to improve social confidence and information quality by successfully detecting and reducing the spread of misinformation across digital platforms.

Objectives:

- Develop advanced preprocessing methods to remove stop words, manage HTML content, and accurately execute lemmatization and punctuation management, ensuring the cleanliness and relevance of the data for model training.
- Develop effective methods for detecting and managing duplicate and irrelevant data in the dataset to improve overall performance.
- Explore machine learning algorithms, such as deep learning or ensemble methods, to create robust system that can accurately identify fake news articles based on textual content.
- Employ advanced NLP method including word embedding And padding to improve the performance of the model.
- Select the best performing model based on accuracy, precision, recall, F1+score and by comparing ROC curves.

2 LITERATURE REVIEW

The influence of social media platforms on news consumption habits has been a focal point of research. In 2017 (k Shu, A Sliva, & J Tang)highlighted how people consume news, with a growing number depending on social media as their primary information source. Surveys conducted between 2012 and 2016 identified a notable increase in news consumption on social media platforms, rising from 49% to 62%. This result has led to social media exceeding television as the primary source of news for many individuals. However, this transition comes with significant drawbacks, notably the proliferation of false information disseminated through social media channels. Scholars like (Allcott & Mathew Gentzkow) have pointed out the prevalence of intentionally misleading articles on these platforms, posing a threat to the authenticity of information consumed by the public.

To combat the spread of fake news, researchers have explored various strategies. Four decades of deception detection research has helped people to detect lies. Even though it is not successful, but it showed 4% better chance on a meta data analysis (Bond & DePaulo). With a particular emphasis on linguistic and network techniques. (Chen & Niall J Conroy) have studied into the use of manual fact-checking websites, by using linguistic approach by monitoring frequency and pattern of pronoun, conjunctions, and negative emotion word usage (Feng & Hirst G). This analysis has shown high accuracy in classification with limited domain. Scholars come with a conclusion that linguistic approach should be built on multiple layers from lexical to highest discourse level analysis for maximum performance.

However, the scalability of these manual methods presents challenges, particularly as the volume of news content increases. To address these scalability issues, automatic fast-checking techniques have been developed, leveraging natural language processing (NLP) and machine learning (ML) algorithms. (k Shu, A Sliva, & J Tang)(classified the fake news into data-oriented, feature-oriented, and model- oriented. Data oriented usually contains different kind of characteristics. Feature oriented aims to determine effective feature. By using these classifications, they developed a model using support vector machines (SVM) and logistic regression classifiers to classify news articles based on linguistic features extracted through lexicon-based analysis.

Using analysis one can find patterns, style and capture the important elements in false information, by using N-gram and “Bag of words” models (Kitti Nagy, 2021). But there are some limitations that N-gram and “Bag of words “model had high chance of losing important information by ignoring context and semantics of words. Scholars came up with an idea that morphological and syntactic analysis seem to improve method of analysing. This Syntactic analysis determine the importance of words in sentences. This article mainly focused on improving fake news classification using dependency grammar by using dependency gram weight(Dgw) and MultipleDgw. But these techniques have showed insufficient accuracy results.

Despite the advancements in classification-based approaches (Victoria L. Rubin, Niall J. Conroy)8) argued that classification alone may not be sufficient, as news contains humour, irony and satire. They collected the data from 360 news articles from US and Canadian national newspapers. Data set has been selected in two sets and merged into legitimate news. By analysing these data, they found that headlines are more relevant for finding satires. By using support vector classification (SVC) algorithm they shows that they had 85% accuracy, which showed high improvement compared to Mihalcea (2016) who achieved 65% baseline by using tf-idf method for the same data. Scholars says that they found individual textual features like parts of speech and punctuation mark are highly indicative for presence of satire. This has raised the accuracy by 5%. Even though the accuracy is high this pattern failed to translate longer sentences. To update the system, they modified semantic distance approach between the lead sentence and last sentence. After updating the system the results of Mihalcea and Pulman showed 84% by using word wise semantic distance, but (Victoria Rubin, 2016)model showed 83% demonstrating negative effect. The results are low when they had a large data.

In addition to classification techniques and satire, the integration of vast data sources presents a challenge for organizations. Selection of features makes difficult when we have a large data, taking this to consideration (Ruchansky N. S., 2017) proposed a CSI model that built on deep neural networks by utilizing three characteristics at once (text, response and source).The main cause of CSI model is that it explicitly outputs information on both article and the user. Scholars proposed CSI model as it does not require social graph, domain knowledge and distribution of behaviour that occurs in the data. They made a model setup by using temporal partitions and specific features as an important component. To check the performance of model, data collected from twitter and Weibo they spread 80% of data for training sasmples,5% for parameter tuning and remaining 15% for testing. Comparison has been made between the CSI model and five-state-of—the-art models (SVM, DT-Rank ,DTC,LSTM,GRU), that have been used for classification task. CSI model showed high accuracy and gives high performance by integrating user features boost up to 4.3% compared to five-state-of-the-art models. However, there are some limitations, the performance is less when it comes to text based.

.

The accuracy and performance of the model is negatively impacted by duplicate and irrelevant characteristics in a dataset (Ahmed, 2022). Scholars collected data from open source which includes articles from 2016. Researcher comes up with an idea that word cloud visualization will helps us to identify important entities in the data. They visualzed data with an example that Political, Americas,2016 and Obama are the high frequent words by applying word cloud to his data. Furthermore, they categorised data into seven categories i.e. false connection, False content, manipulated content, Satire, misleading content, imposter content and Fabricated content. All among the seven categories Fabricated content holds 100% of false and it is deceived and harm. To check the results, they performed experiment on different machine learning models all among them Logistic regression failed to give good results. Later on, they performed using support vector machine, Navie bayes and Passive aggressive (PA) which gave good results. To check an accuracy, researchers had compared results with data set and noticed that Passive aggressive obtains performance above the base line 0.5 compared to other.

Conclusion:

From above observation, researchers had developed machine learning model by using variety of classification and deep learning techniques that differentiate between the fake and true data. However, these models only address linguistic, syntactic and satire content in the data that leads to poor performance of model. There are other issues like removal of stop word, lemmatization, removal of html content and punctuation of characteristics. By filling these research gaps, we can improve the model performance.

3 METHODOLOGY

To address problem stated in problem definition, I propose a comprehensive approach that combines machine learning techniques and natural language processing (NLP) by analysing the patterns. My approach is to use computational techniques to find patterns and signs of misleading data in text and to develop a system to combat the spread of misinformation, safeguarding public trust and democratic processes.
The proposed methodology consists of following steps:

3.1 Data Collection

The dataset used in from the open-access platform Kaggle, specifically the ISOT Fake News Dataset (Ahmed et al., 2018). This dataset includes a wide range of articles covering Middle Eastern governance news, US news, and political content. It includes contributions from reputable organizations such as The New York Times, The Guardian, and Bloomberg, especially during the 2016 US elections. With its large data, the dataset is abundant enough for both training and testing algorithms.
Redundancy, availability, and amount of data were critical factors in the selection of this dataset. Furthermore, the extensive use of existing algorithms on this dataset creates a useful structure for comparative analysis, allowing for the evaluation of outcomes across multiple methodologies.
The dataset comprises of two csv files, named Fake.csv and True.csv. Each of these file is structured with four columns namely :title, text, subject and publishing date of the articles.

3.1.1 Feature Description

The fake dataset contains 22800 articles and true dataset contains 21417 articles collected from various news, including politics, world news, government news, US_ news and middle east news in the year between 2016 and 2017. Both the true and fake datasets consists of 4 features as shown in the table below.

Feature Name	Feature type	Description
Title	Predictor	Name of the article
Text	Predictor	Content in the article
Subject	Predictor	Type of news
Date	Predictor	Published date

Table 1:Feature Description

3.2 Data preprocessing

In The processing we perform following steps:

3.2.1 Tokenization

Tokenization is a method of breaking up text into smaller units. Text is tokenized at the word level, resulting in discrete words or tokens.

3.2.2 Lemmitization

Lemmatization is commonly used in natural language processing for text analysis. Lemmatization is a linguistic process that reduces words to their base or root form in order to handle various word forms.

3.2.3 TF-IFD vectorization

The TF-IDF vectorization technique converts textual input into numerical forms that machine learning models can understand. This method weights each word in the text according to its frequency in the document (Term Frequency, TF) and inverse frequency over the entire corpus (Inverse Document Frequency, IDF). This method is essential for converting unstructured text into structured numerical data, which allows machine learning algorithms to analyse and extract insights from textual data.

3.2.4 Padding

Padding is a technique used in natural language processing (NLP) and deep learning to ensure that variable-length sequences have uniform lengths. Padding in text data typically consists of adding special tokens (usually zeros) to the beginning or end of sequences to make them all the same length.

3.3 Selection of Machine learning models

3.3.1 logistic regression

It is a statistical method used for binary classification tasks with a categorical output variable and only two classes. It calculates the probability of a binary outcome based on one or more predictive variables by assuming a linear relation between the features and target. The probability of outcome, dependent variable is noted between 0 and 1((Hosmer et al., 2013).

Mathematical representation of logistic regression is:

$$p(y) = \frac{1}{1 + e^{-z}}$$

Where, e is base of algorithm approximately equal to 2.718 and Z is linear combination of input features and model coefficient, calculated as:

$$z = \quad \beta 0 + \beta 1X1 + \beta 2X2 + \ldots + \beta n * Xn$$

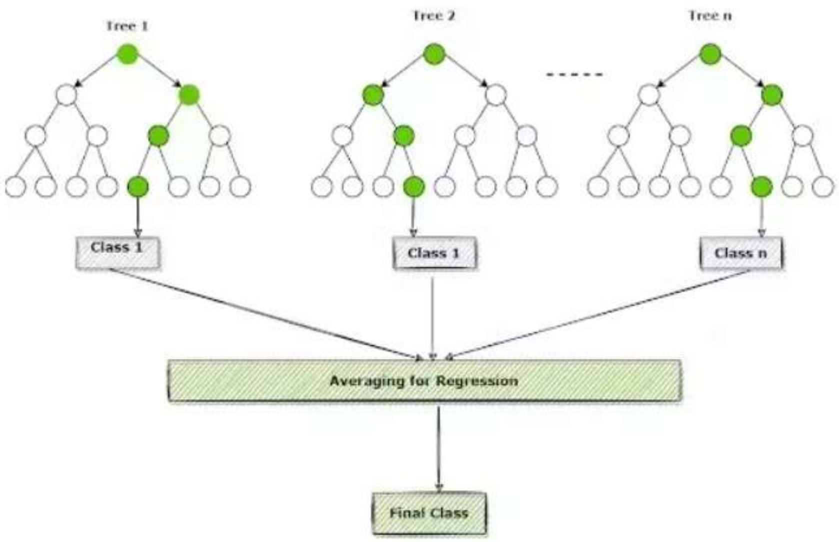
3.3.2 Multinomial Naïve Bayes

It is based on 'Bayes theorem' with the assumption of independence among features and it is widely used in classifications. In text classification Multinomial Naïve Bayes is widely used, by assuming a feature vectors by following a multinomial distribution. It is most efficient and effective inductive learning algorithm which is commonly used in NLP for classification tasks.((Kibriya et al., 2004).

3.3.3 Random Forest

: Random Forest is ensemble learning technique in which multiple decision trees work together to increase prediction accuracy ((Breiman, 2001). Each tree is trained using a distinct subset of the training data and features. Random forest reduces overfitting and variance in individual trees by accurately capturing complex relationships between dependent and independent variables. Working process of Random Forest includes following steps:

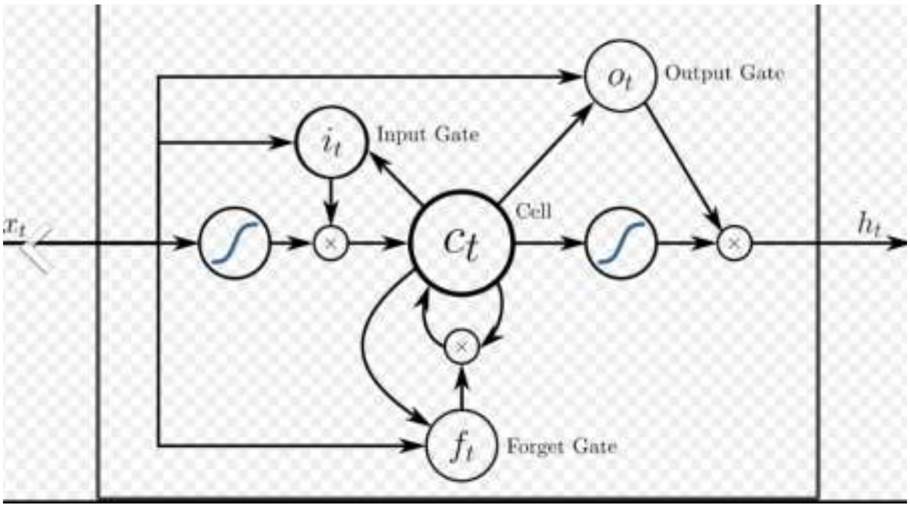
- ➔ Pick random data from the training set.
- ➔ Create decision tree for the subsets and each tree analyse the result.
- ➔ Pick random data from the training set.
- ➔ Create decision tree for the subsets and each tree analyse the result.
- ➔ Final output depends upon the majority of average voting.



.

3.3.4 Long Short-Term Memory(LSTM)

LSTM is a kind of recurrent neural network (RNN) structure used to analyse sequences like text. Unlike standard RNNs, LSTMs allow us to better capture long-term dependencies and contextual text data by regulating information input and output within the cell (Greff et al., 2017). LSTMs are widely used in natural language processing for applications like text categorization, language translation, and sentiment analysis. LSTM is able to utilize both unidirectionally and bidirectionally, whereas Uni-directional LSTM processes data in a single direction, either forward or backward, while bi-directional LSTM processes information in both ways.



In above structure:

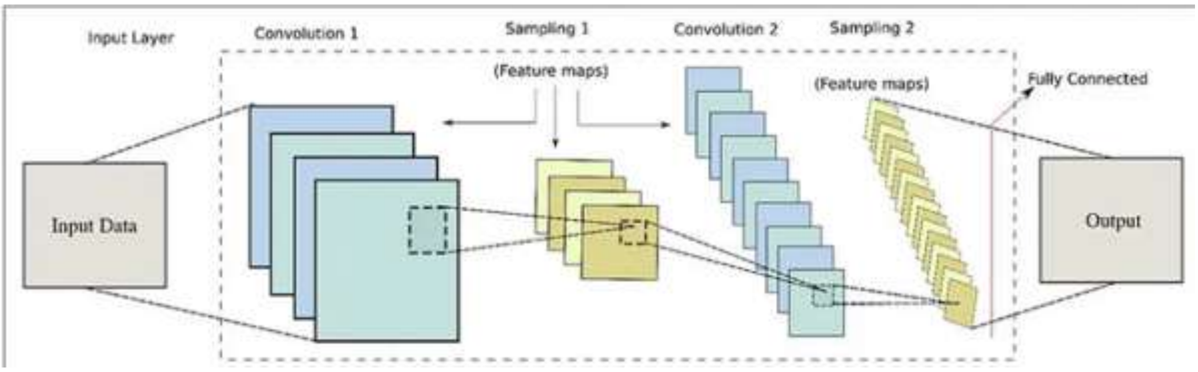
Forget gate: It recognize the information through the network and forgets once it is processed.

Input gate: Input gate helps in deciding the information by updating the cell and transfers information which is important.

Output gate: it is last gate that helps in deciding the hidden state of network and helps the hidden state to carry the information.

3.3.5 Convolutional Neural Networks (CNN):

CNN is highly effective in computer vision and has demonstrated higher efficiency in classification tasks within natural language processing. For text classification, CNN used 1D convolution and embeddings to process sequential data. CNN is made up of layers such as convolutional and pooling, where convolutional applied to text data and the pooling layer retaining the importance of patterns and features. It has a significant advantage in text classification because of its ability to learn both local and global features efficiently ((Zhang & Wallace, 2015). It demonstrates an effective approach to text classification, which is significant for various NLP applications.



3.4 Selection of evaluation methods

To evaluate the performance of machine learning models following metrics will be used to evaluate the machine learning model.

3.4.1 Accuracy

Accuracy is a simple statistic that evaluates the number of true predictions made by the model and provides the model's overall performance. It is represented as the ratio of predicted instances to total instances.

$$Accuracy = \frac{True\ Posituives + True\ Negatives}{True\ Positives + True\ \ Negatives + False\ Positives + False\ Negatives}$$

3.4.2 Precision

Precision is a metrics that calculates the It proportion of true positive predictions among all positive predictions generated by the model, indicating the model's ability to avoid false positives. Statistically expressed as:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

3.4.3 Recall

Recall calculated the ratio of true positive predictions to real positive cases in the dataset. It is also known as sensitivity or the true positive rate. Recall evaluates the model's ability to identify all relevant instances in the data.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

3.4.4 F1-score

F1-score is the average of precision and recall, which results in balanced metric by considering both precision and recall. It provides overall performance of model's performance by using these two metrics.
Mathematically given as:

$$F1\ score\ = 2 * \frac{Precision * recall}{Precision + recall}$$

3.4.5 ROC curve and AUC

The Receiver Operating Characteristic (ROC) curve shows the relationship between the true positive rate (TPR) and the false positive rate (FPR) at various classification thresholds. It helps in visualizing the model's performance over several thresholds through Area under the curve. Area under the curve (AUC) ranges from 0 to 1, AUC of "1" represents perfect classifier, whereas "0" represents random classifier.

True Positive Rate (TPR):
TPR calculates the proportion of actual positive cases identified by the classifier. It measures the model's capacity to distinguish positive cases out of all actual positives.

$$True\ positive\ rate(TPR) = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

False Positive Rate (FPR):
It calculates the proportion of actual negative onstances incorrectly classified as positive by classifier.

$$False\ Positive\ rates(FPR) = \frac{False\ Positives}{False\ Positives + True\ Negatives}$$

4 ARTIFACT

4.1 Data reading

To read the data open-source python library have been used, as it provides strong foundation to read and manipulate the data. By using ‘pd.read_csv()’ function data have been read by creating two data frames named df_fake to read fake dataset and df_true for true data set as shoen in figure 1

```
df_fake = pd.read_csv('Fake.csv')
df_true = pd.read_csv('True.csv')
```

fig1: Reading dataset

4.1.1 Data displaying

Understanding structure and contents plays a crucial role in data preprocessing, in order to reduce the complexity we can use ‘head()’ function from pandas libraries that helps us to read missing values, by displaying the first five rows by default , which provides the overview of the data. Head of two data frames(df_fake ,df_true)have been shown in fig 2 and fig 3

	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year’...	Donald Trump just couldn t wish all Americans ...	News	31-Dec-17
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	31-Dec-17
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	30-Dec-17
3	Trump Is So Obsessed He Even Has Obama’s Name...	On Christmas day, Donald Trump announced that ...	News	29-Dec-17
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	25-Dec-17

Fig 2: fake news data set sample

	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017

Fig3: True news dataset sample

4.2 Data preprocessing

Data preprocessing is a crucial step in data mining and analysis, which optimizes raw data to make it suitable for analysis and modelling. This process involves converting the data into a structured format that is easily understandable.

4.2.1 Labelling

To prepare the data for analysis and differentiate between the two data frames, we introduced a new column named 'target' and assigned labels of 0 and 1 accordingly. Specifically, we labelled 0 for the fake data frame and 1 for the true data frame, as shown in Figures 4 and 5. Data labelling is a crucial step as it supports model training and enhances accuracy. With this labelling we can effectively distinguish between true and fake news articles, thereby enabling more accurate analysis.

	title	text	subject	date	target
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	31-Dec-17	0
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	31-Dec-17	0
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	30-Dec-17	0
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	29-Dec-17	0
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	25-Dec-17	0
...	...	...	...	...	...
22795	McPain: John McCain Furious That Iran Treated ...	21st Century Wire says As 21WIRE reported earl...	Middle-east	16-Jan-16	0
22796	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	21st Century Wire says It s a familiar theme. ...	Middle-east	16-Jan-16	0
22797	Sunnistan: US and Allied 'Safe Zone' Plan to T...	Patrick Henningsen 21st Century WireRemember ...	Middle-east	15-Jan-16	0
22798	How to Blow $700 Million: Al Jazeera America F...	21st Century Wire says Al Jazeera America will...	Middle-east	14-Jan-16	0
22799	10 U.S. Navy Sailors Held by Iranian Military ...	21st Century Wire says As 21WIRE predicted in ...	Middle-east	12-Jan-16	0

Fig4:lebelled fake dataframe

	title	text	subject	date	target
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017	1
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017	1
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017	1
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017	1
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017	1
...	...	...	...	...	...
21412	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017	1
21413	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews	August 22, 2017	1
21414	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017	1
21415	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017	1
21416	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017	1

Fig 5: labelled true data frame.

4.2.2 Data Concatenation

To compile our dataset for analysis, we combined two distinct data frames, df_fake and df_true. This merging process was executed along the row axis using the concat() function from the pandas library. By merging these data frames, we create a single dataset containing both fake and true news articles, simplifying analysis and model training. Head of data after concatenating has shown in figure6

4.2.3 Handling Duplicates

Duplicates in a dataset can lead to inaccurate results. After examining our dataset, we identified 209 duplicates. To ensure data integrity and accuracy in our analysis, we removed these duplicates using the drop_duplicates() function. This step helps in maintaining the quality of our dataset and prevents redundancy in our analysis.

4.2.4 Null values

Null values can affect the reliability and accuracy of data analysis. It's crucial to identify them to prevent data misinterpretation. We employed the isnull().sum() function from the pandas library to count missing values in each column of the data frame..Performing this step we confirmed that there is no null values present in data.

4.3 Visualization with pre-processed data

4.3.1 Distribution of subject according to real and fake

To visualize the distribution of subjects in both true and fake news data, we utilized the value_counts() function to count the occurrences of each subject category .Additionally, we created a bar plot using Seaborn's countplot() function, Illustrating the distribution of subjects categorized by real or fake. The plot is titled "Distribution of The Subject According to Real and Fake Data" as shown in figure 8.

```
subject
politicsNews       11272
worldnews          10145
News                9048
politics            6411
left-news           4302
Government News     1495
US_News              774
Middle-east          770
Name: count, dtype: int64
Text(0.5, 1.0, 'Distribution of The Subject According to Real and Fake Data')
```

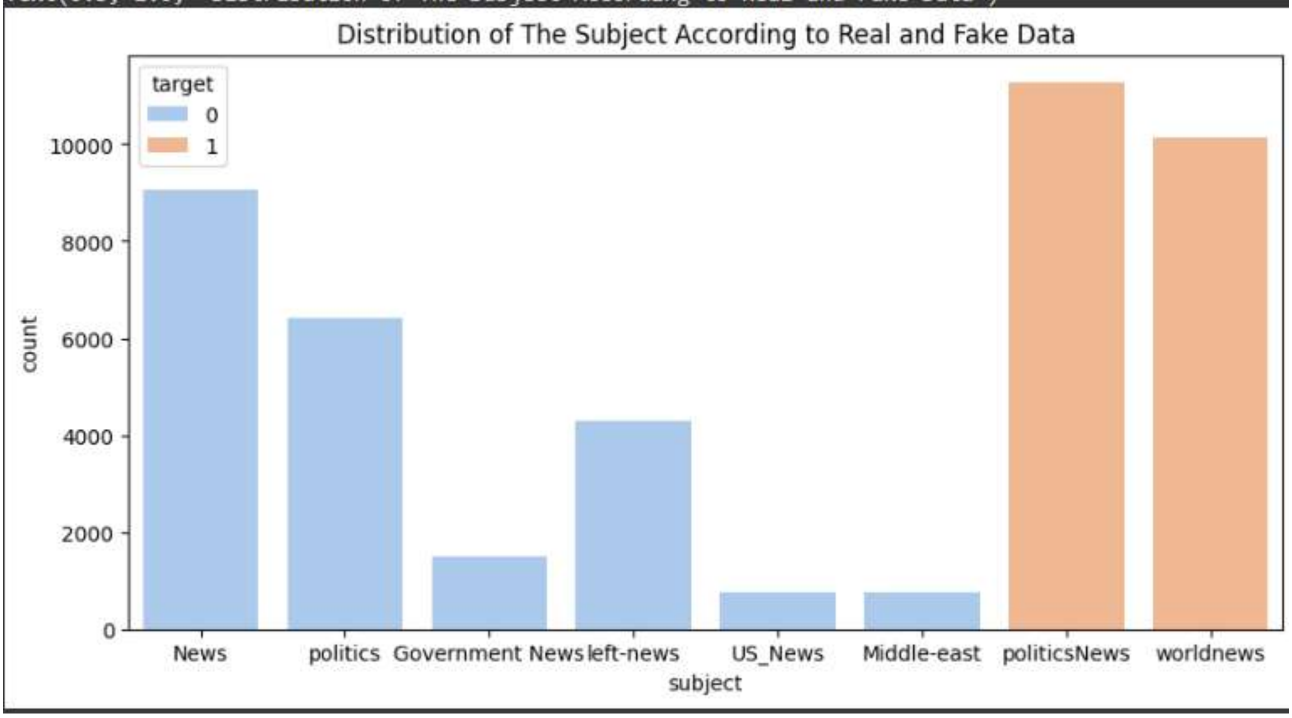


fig7: Distribution of Subject according to Real and Fake Data

4.3.2 Distribution target

We employed two different kinds of plots to show the distribution of the target variable in our dataset. First, a pie chart was utilised to provide an overview of the distribution by showing the percentage of each target category. Furthermore, a bar plot provided a more detailed perspective on the frequency of each category, enhancing our understanding of the dataset as shown in fig8.

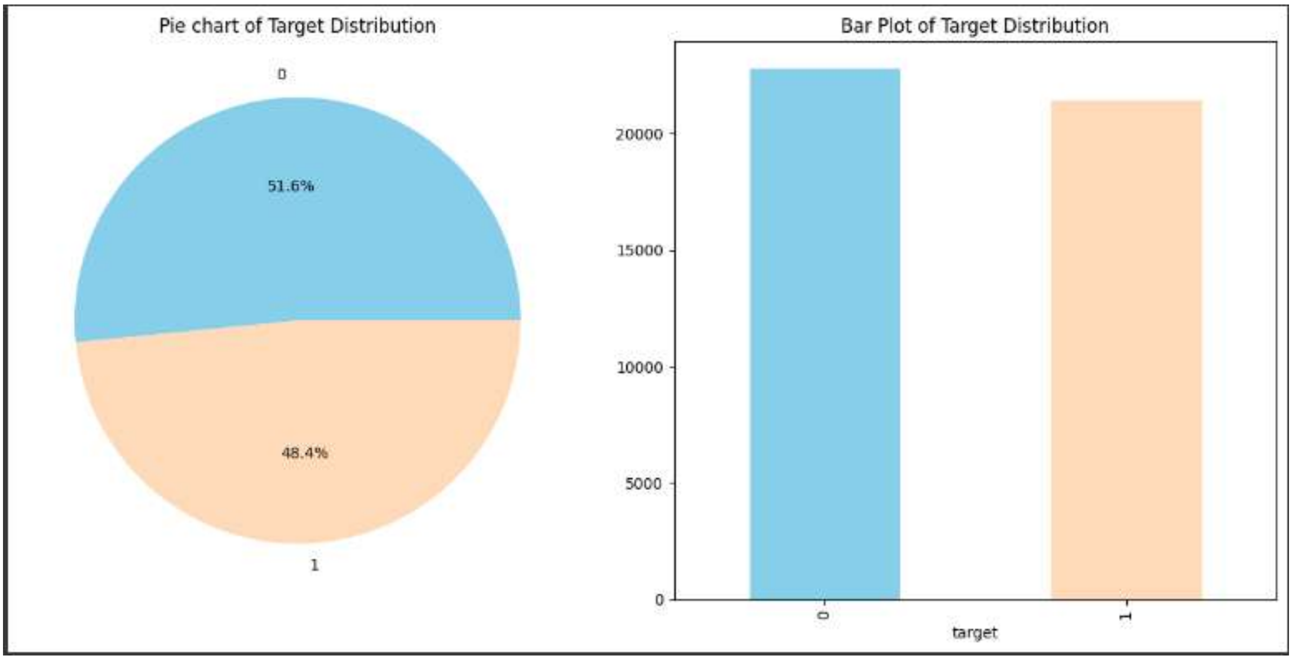


Fig8: Target distribution

From above pie graph, we can identify that 51.6% of data contains fake amd remaining 48.9% contains true data.

4.4 Data cleaning

Data cleaning is the process of identifying and cleaning errors in a dataset to improve its quality and reliability. This step is crucial as it ensures the accuracy of the data, which in turn improves the validity of any subsequent analysis or modeling efforts.

4.4.1 Replacing unwanted data.

Our dataset contains various types of unwanted text elements like URLs, emojis, usernames, non-alphabetic characters, and consecutive letters, which can adversely affect the accuracy of our model during training. To address this, we initiated a series of preprocessing steps.
Firstly, all text was converted to lowercase using the 'lower()' function to ensure uniformity in text analysis. BY, using regular expressions with the 're.sub()' function, URLs were replaced with a standard string, effectively removing irrelevant text from our dataset.

However, text includes emojis, In order to replace a dictionary of emojis and corresponding meaning have been defined and These emojis were then replaced with their respective meanings,, which adds semantic clarity to the text.
Furthermore, to maintain text clarity, non-alphabetic characters (such as symbols and numbers) were eliminated, along with consecutive occurrences of the same letter. These steps collectively ensure that our text data is appropriately ready for analysis.

4.4.2Tokenization

To break down the sentences into individual words, we used tokenization. By importing the 'word_tokenize' function from the 'tokenize' module of the Natural Language Toolkit (NLTK) library.

Initially, the text undergoes tokenization, which splits sentence it into individual words or tokens. Subsequently, short words, those with a length of one character are removed to enhance the quality of the text data.

4.4.3 Stopword removal

Stop words are commonly occurring words in a language that lack significant meaning in the context of a reading articles. To remove these stop words, we imported the stopwords module from the NLTK library. By leveraging the stopwords module, we accessed a predefined set of common words that can be filtered out from text data. Common words such as 'the,' 'is,' and 'and' are then eliminated from the text. This process helps in focusing on meaningful words and provides clearer insights into the data.

4.4.4 Lemmitization

lemmatization is used to transform words into their base or dictionary form. This process helps in standardizing words so that different variations of the same word can be treated as identical during analysis. For example run, ran and running are lemmatized to run, which increases accuracy of text analysis.
After removing stop words, lemmatization is applied to the text by import WordNetLemmitizer class from ‘Stem module of Natural language tool kit library. When lemmatization is applied words are bring backed to their form and irrelevant words are removed from the data after lemmatizing.

After performing all these cleaning techniques, tokens are rejoined back to their text form and head of the pre-processed has been displayed in fig 9.

	text	target	preprocessed_text
0	WASHINGTON (Reuters) - The U.S. Senate will ta...	1	washington reuters senate take first step towa...
1	WASHINGTON (Reuters) - The Interior Department...	1	washington reuters interior department watchdo...
2	It was 1991 when Trump testified to Congress a...	0	trump testified congress irrational exuberance...
3	Has there ever been a US Olympic athlete who, ...	0	ever u olympic athlete representing united sta...
4	The Republican senator who gained infamy last ...	0	republican senator gained infamy last year att...

Fig9: Difference between original text and pre-processed text

4.5 Exploratory data analysis

4.5.1 Violin plot

The violin plot is a powerful visualization tool used to understand the distribution of numeric data across different categories or groups. In this context, the violin plot illustrates the distribution of text length in fake and True news articles.

In the plot, each violin represents a target, with the width indicating the frequency of data points at different text lengths. The shape of the violin conveys information about the distribution—wider sections indicate higher frequency, while narrower sections suggest lower frequency. Additionally, the central line within each violin represents the median, and the surrounding area represents the interquartile range.

By comparing the violins for fake and real news articles, from figure 10 we can observe the differences in the distribution of text length. For fake news violin plot is wider which indicates high frequency of words and for true news plot is narrow compare to fake news which shows low frequency.

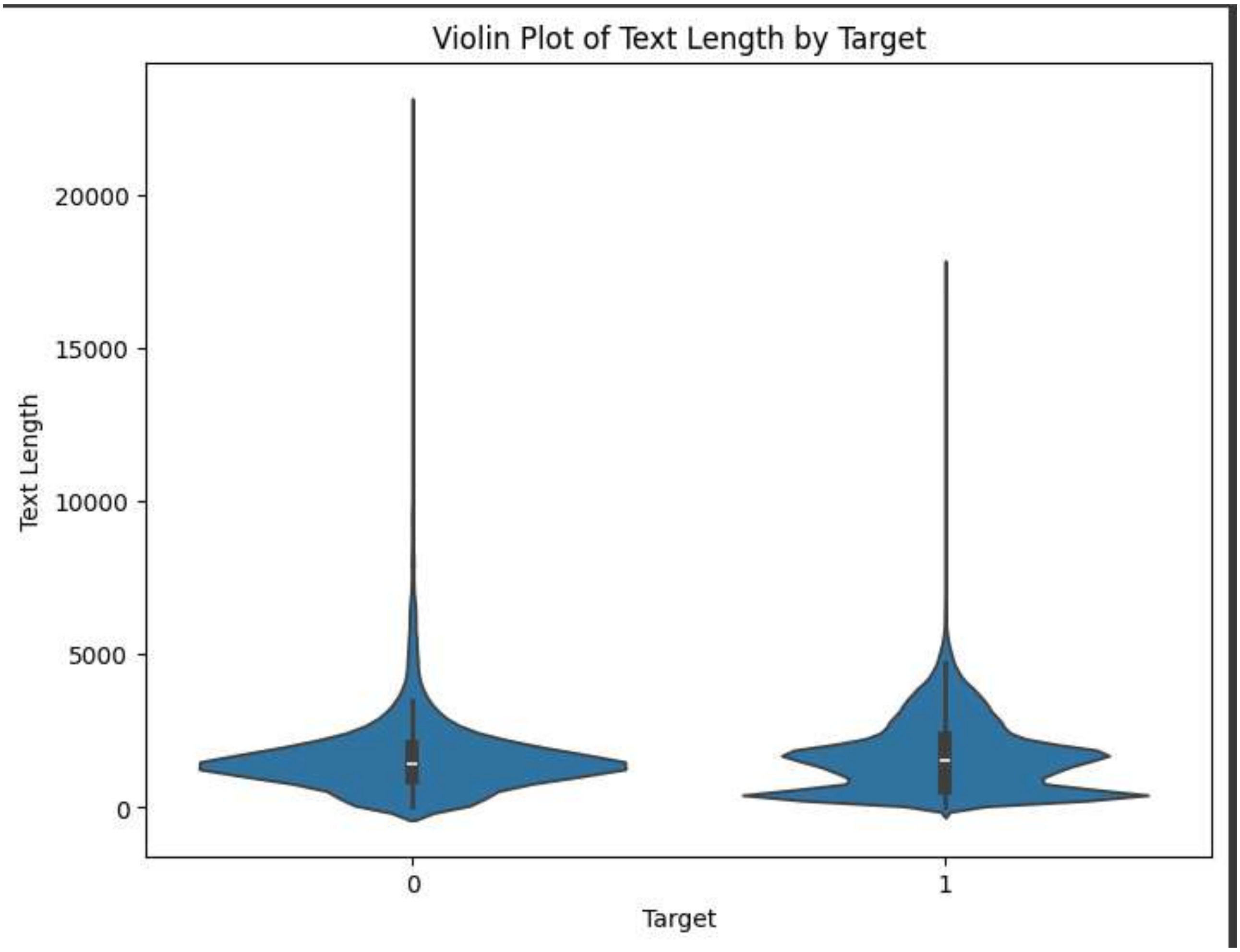


Fig10:Violin plot for text length by target

4.5.2 Word cloud visualization

Word clouds are used to visually represent the frequency of words within a given text . They provide a quick way to identify the most common words in a dataset and visualize their importance. To visualize the most common words in pre-processed data we used 'WordCloud' class from Wordcloud library. By filtering the data based on 0 and 1, corresponding to fake and true, we created a separate wordcloud for each category.it provides a clear view of most frequentlu occurring words in both fake and true news. Generated wordclouds are shown in figure 11and 12, which shows the distribution of words across two categories. For true news words like united, Donald, said, white house, prime minister are the most occurring words as shown in fig 11. For fake news even, one, say, featured, image, according, made, make , going are the most occurring words as shown in fig 12.

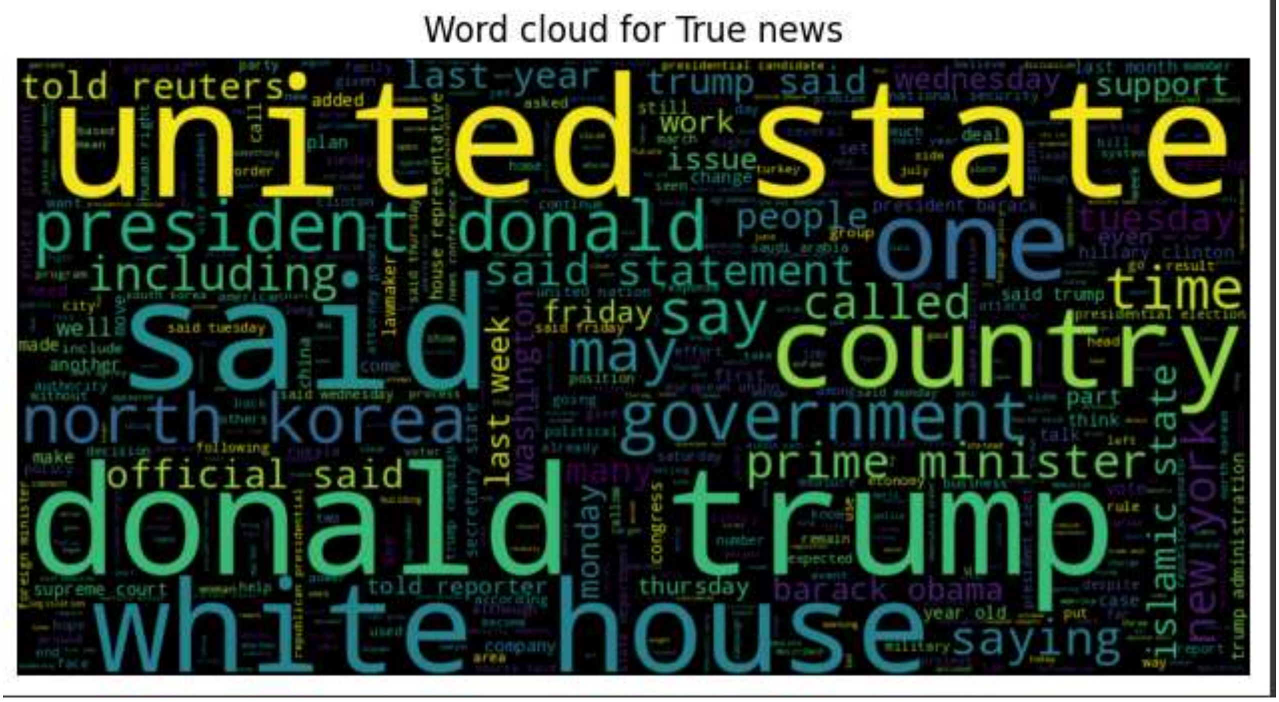


Fig 11: word cloud for true news

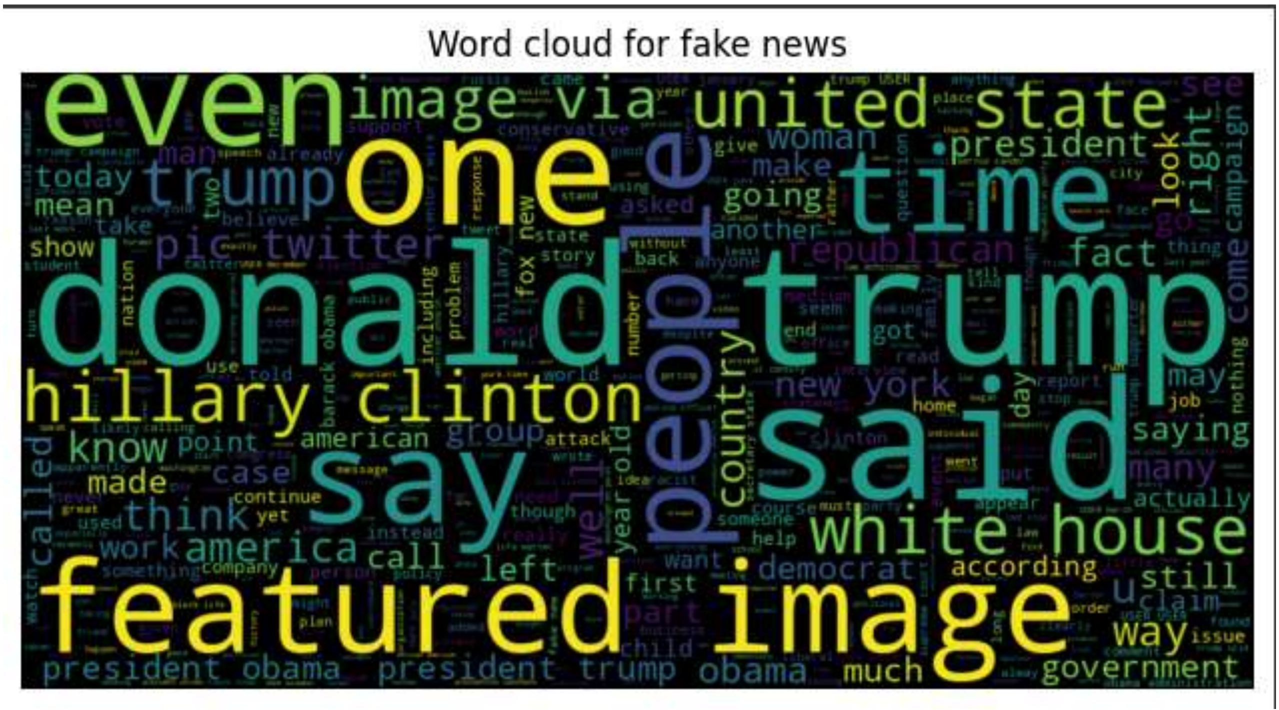


Fig 12: word cloud for Fake news

4.5.3 Histogram analysis

Histograms representing the distribution of word counts in fake and real news texts as shown in figure 13. Two subplots are displayed side by side, showing the frequency distribution of word counts for a specific category of news. The left subplot illustrates the word count distribution for fake news texts, while the right subplot illustrates the distribution for real news texts.
From histogram we can see number of words to be different 500 words are most common in real news category while and 250 words are most common in fake news category.

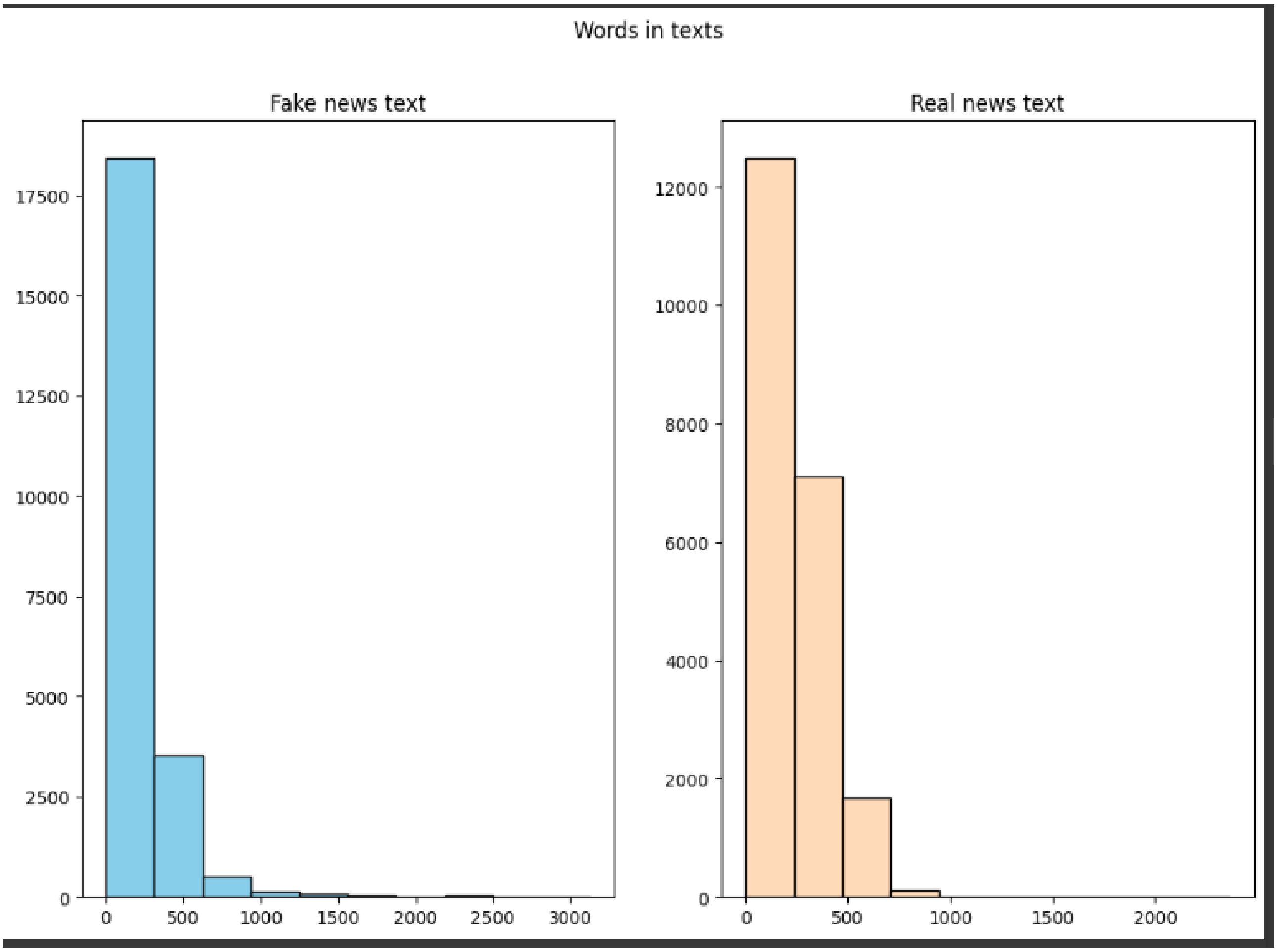


Figure 13 : Distribution of word count in fake and real news

4.6 Modeling

4.6.1 Train-test split

Following data cleaning and preprocessing data undergoes train-test split to develop machine learning model. It divides the dataset into two subsets: the training set and the testing set. The training set is used to train the model on the data, while the testing set is used to evaluate the model's performance. In order to tarin-test the data we imported ‘train_test_split function from 'sklearn.model selction' model to split a dataset into training and testing sets.80% of data is used to tarin the model and 20% is used to test the data.

4.6.2 TF-IDF vectorization

“Term frequency – Inverse Document Frequency” vectorization converts textual data into numerical format, allowing algorithms to process and understand. It represents the importance of each term in a document in comparison to a collection of documents. It converts words to vectores by combining the concepts of TF and IDF. It is widely used in text classification and clustering.

Implementation:

As our data consists of a large amount of text we use TF-IDF vectorization by importing ‘TfidfVectorizer’ class from ‘sklearn.feature_extraction.text’ to convert text to numerical vector by specifying parameter ‘max_feature’ to 5000 which controls the dimensionality of feature score without losing the data.

4.7 Implementing machine learning algorithms

4.7.1 Logistic regression:

Implementation:

To implement logistic regression, we import the 'LogisticRegression' class from the'sklearn.linear_model' module and 'uniform' class from scipy.stats. Hyperparameter tuning is performed using random search, iterating through different combinations of penalty, inverse regulation strength (C) and by optimizing algorithm (solver) values are assigned as shown in table(). Tuned model is then applied to train and test data by giving different iterations and evaluated based on best parameters.

penality	1(Lasso) or l2(Ridge)
C	Inverse of regulation, using distribution between 0 and 4
solver	Optimization algorithm to use, either liblinear or saga

4.7.2 Multinomial Navie bayes

Implementation:

To implement MultinomialNB, we imported the MultinomialNB class from the sklearn,navie_bayes module. Hyperparameter tuning is performed by random search approach by using parameters shown in table(). The model is tuned by iterating through various combinations of hyperparameters. After tuning, the model is implemented on train and test data, by giving different iterations and evaluated based on best parameters.

Alpha	It is smoothing parameter, we used the values of 0.1, 1.0 and 2.0
fit_prior	(True or False) , to learn class perior probabilities or not
Class_prior	Probabilities of the class

4.7.3 Random forest

Implementation:

To implement random forest, we imported the RandomForestClassifier class from the sklearn.ensemble module and randit function from scipy.stats . Hyperparameter tuning is performed by creating a dictionary(param_dist) which holds different parameters and values shown in table() , By iterating through different combinations of hyperparameters using random search. . The tuned model is implemented on train and test data and evaluated using the best possible parameters.

n_estimators	Randint(50,200)
Max_features	[‘auto’, ‘sqrt’]
Max_depth	[10,20,30,40,50, none]
Min_sample_split	Rsndint(2,10)
Min_samples_leaf	Randint(1,10)
Bootstrap	[True, False]

4.8 Deep learning models

4.8.1 Long-short term memory

To implement LSTM we imported LSTM,Dense, Embedding and SPatialDropout from Keras library. After importing the library we performed preprocessing steps including tokenization to convert text data to numerical form .Following tokenization we applied padding by setting a maximum length of 150 ,ensuring sequence has the same length. Our model architecture included embedding and LSTM layers designed to capture complex temporal dependencies in the data. Following design, we built the model using binary cross-entropy loss, which is ideal for binary classification tasks. The Adam optimizer allowed for efficient parameter optimization during training. The model's summary is shown in fig(), and it is implemented by assigning values to the parameters shown in table(). The model is applied to tarin and test data by assigning different batch sizes and evaluating the model.

Parameter	Values
Max_len	150
Embedded_dim	100
Lstm_units	150
Dropout	0.4
Batch_size	32,64,128
epochs	5

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 150, 100)          9595200

 spatial_dropout1d (Spatial  (None, 150, 100)          0
 Dropout1D)

 lstm (LSTM)                 (None, 150)               150600

 dense (Dense)               (None, 1)                 151

=================================================================
Total params: 9745951 (37.18 MB)
Trainable params: 9745951 (37.18 MB)
Non-trainable params: 0 (0.00 Byte)
```

4.8.2 Convolutional neural network (CNN)

To implement Convolutional neural network we imported Conv1D and GlobalMaxPooling1D from keras library, which are commonly used in CNN architectures . After importing the libraries, we performed preprocessing similar to Long-short term memory by tokenizing and padding to ensure that the sequences were of equal length. After preprocessing, the model is defined using various layers such as embedding, COnvid1D, and GlobalMaxPooling1D layers, compiled with binary_crossentropy, and optimised by using Adam optimizer. The summary of model is shown in fig(). .

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 150, 100)          9595200

 conv1d (Conv1D)             (None, 146, 128)          64128

 global_max_pooling1d (Glob  (None, 128)               0
 alMaxPooling1D)

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 9659457 (36.85 MB)
Trainable params: 9659457 (36.85 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Model is implemented by assigning values to the parameters shown in table(). The model is applied to tarin and test data by assigning different batch sizes and evaluated the model.

Max_length	100
Embedding_dim	150
Conv1D	Number of filters:128 Size of window: 5 Activation: Rectified Linear Unit (ReLU)
GlobalMaxPooling1D	Performs global max pooling over the temporal axis of input sequence
adam	Optimizer for training model parameters

5 EVALUATION

5.1.1 logistic regression:

After fitting the model to both train and test data , in order to evaluate the model , the model is provided with different iterations (5,10,15,20) and for each iteration classification report and accuracy and best parameters has been noted as shown in table(). In classification report ‘0’ indicated fake news and 1 indicates true news.

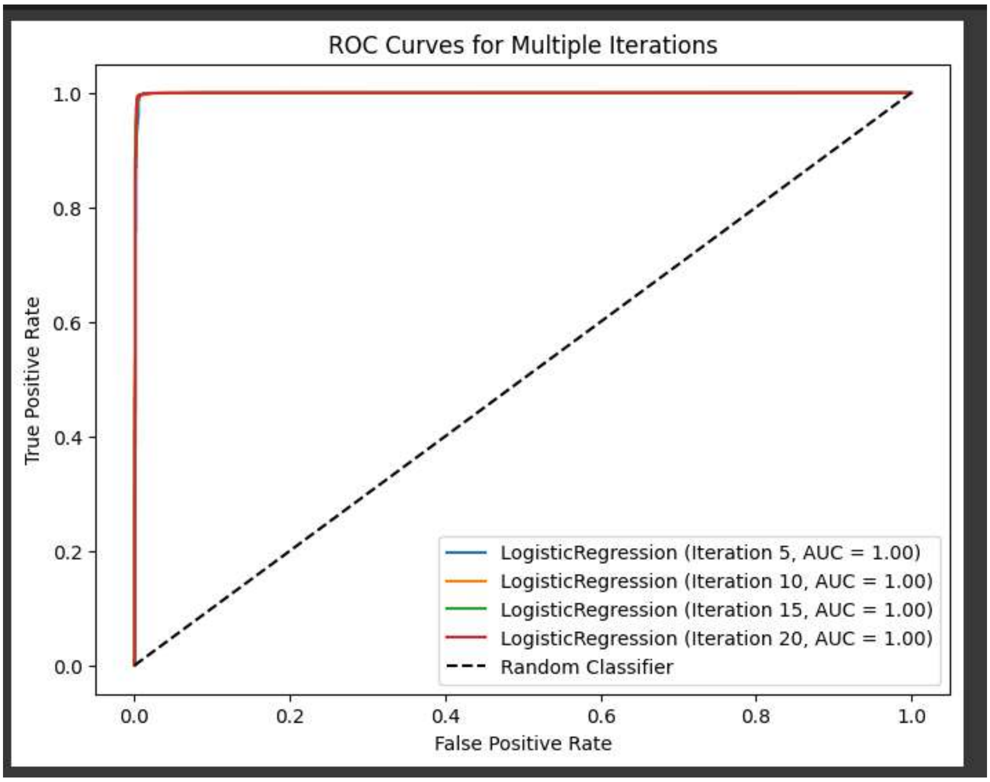
Iterations	Parameters	Time (seconds)	Accuracy	Precision		Recall		F1-score	
				0	1	0	1	0	1
5	{'C': 1.1782775586626744, 'penalty': 'l1', 'solver': 'liblinear'}	109.82	0.9938	1.00	0.99	0.99	1.00	0.99	0.99
10	{'C': 3.0081248582450746, 'penalty': 'l1', 'solver': 'saga'}	339.82	0.9945	0.99	1.00	1.00	0.99	0.99	0.99
15	{'C': 3.6711925775005443, 'penalty': 'l1', 'solver': 'liblinear'}	731.07	0.9946	0.99	1.00	1.00	0.99	0.99	0.99
20	{'C': 3.47413212965694, 'penalty': 'l1', 'solver': 'liblinear'}	622.32	0.9947	0.99	1.00	1.00	0.99	0.99	0.99

Observations:

From above table we can clearly observe that for each iteration, accuracy has been increased which shows overall success of model and at the iteration 20 its shows the high accuracy of 99.47% compared to other and the execution time is more compared to other iterations. The classification report shows high scores in both classes with precision and recall scores varying 99% and 100%, whereas for F1-score it is 99% for both classes.

ROC curve analysis:

Model performance is analysed by plotting a ROC curve for all iterations and compared them in single plot as shown in the figure(). From plot we can observe that AUC is 1.00 for all iterations, which shows that model is performing well in differentiating between fake and true news.



5.1.2 Naive Bayes (MultinomialNB)

After fitting the model to the training and test data, the model was evaluated using the accuracy and classification reports (precision, recall, and F1-score) and the ROC curve.The model undergoes by several iterations (5, 10, 15, and 20), and the best parameter for each iteration is identified. Each iteration is evaluated by comparing the accuracy and classification reports(precision,recall,F1-score) as shown in the table().

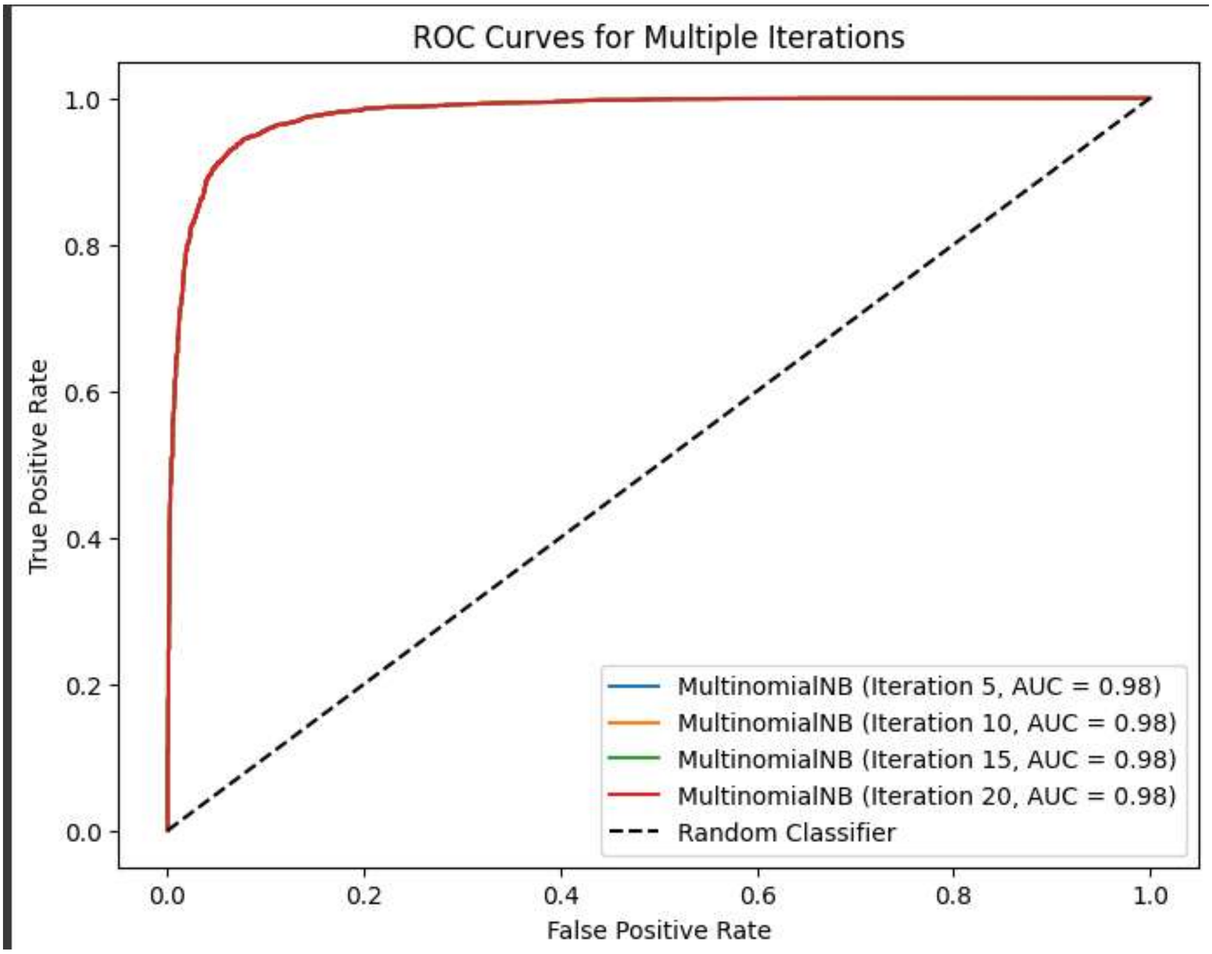
Iterations	Parameters	Accuracy	Training Time (seconds)	Precision		Recall		F1-score	
				Class 0	Class 1	Class 0	Class 1	Class 0	Class 1
5	{'fit_prior': False, 'class_prior': None, 'alpha': 0.1}	0.9324	4.0398	0.93	0.93	0.94	0.93	0.94	0.93
10	{'fit_prior': True, 'class_prior': [0.5, 0.5], 'alpha': 0.1}	0.9324	5.1984	0.93	0.93	0.94	0.93	0.94	0.93
15	{'fit_prior': False, 'class_prior': [0.5, 0.5], 'alpha': 0.1}	0.9324	9.4375	0.93	0.93	0.94	0.93	0.94	0.93
20	{'fit_prior': True, 'class_prior': [0.5, 0.5], 'alpha': 0.1}	0.9324	11.2794	0.93	0.93	0.94	0.93	0.94	0.93

Observation:

The above table clearly shows that accuracy remains constant across each iterations, with an accuracy of 93.24%, but the time taken to execute each iteration has increased. The classification report indicates that the precision score for fake and true news is 93%, while the recall and F1-score for fake and true news are 94% and 93%, respectively. This shows the model's ability to distinguish between fake and true news.

ROC curve analysis:

The model's performance is analysed by plotting a ROC curve based on accuracy scores at various iterations. Figure() shows an AUC score of 0.98, which shows that model can effectively distinguish between true and fake news.



5.1.3 Random forest

After implementing the model on train and test data, the model is evaluated using accuracy, classification reports, and by ROC curve analysis. To evaluate the model, various iterations are provided (5, 10, 15, and 20). By executing each iteration and comparing the best parameters, we collected the accuracy and classification report(precision, recall,F1-score) as shown in table()

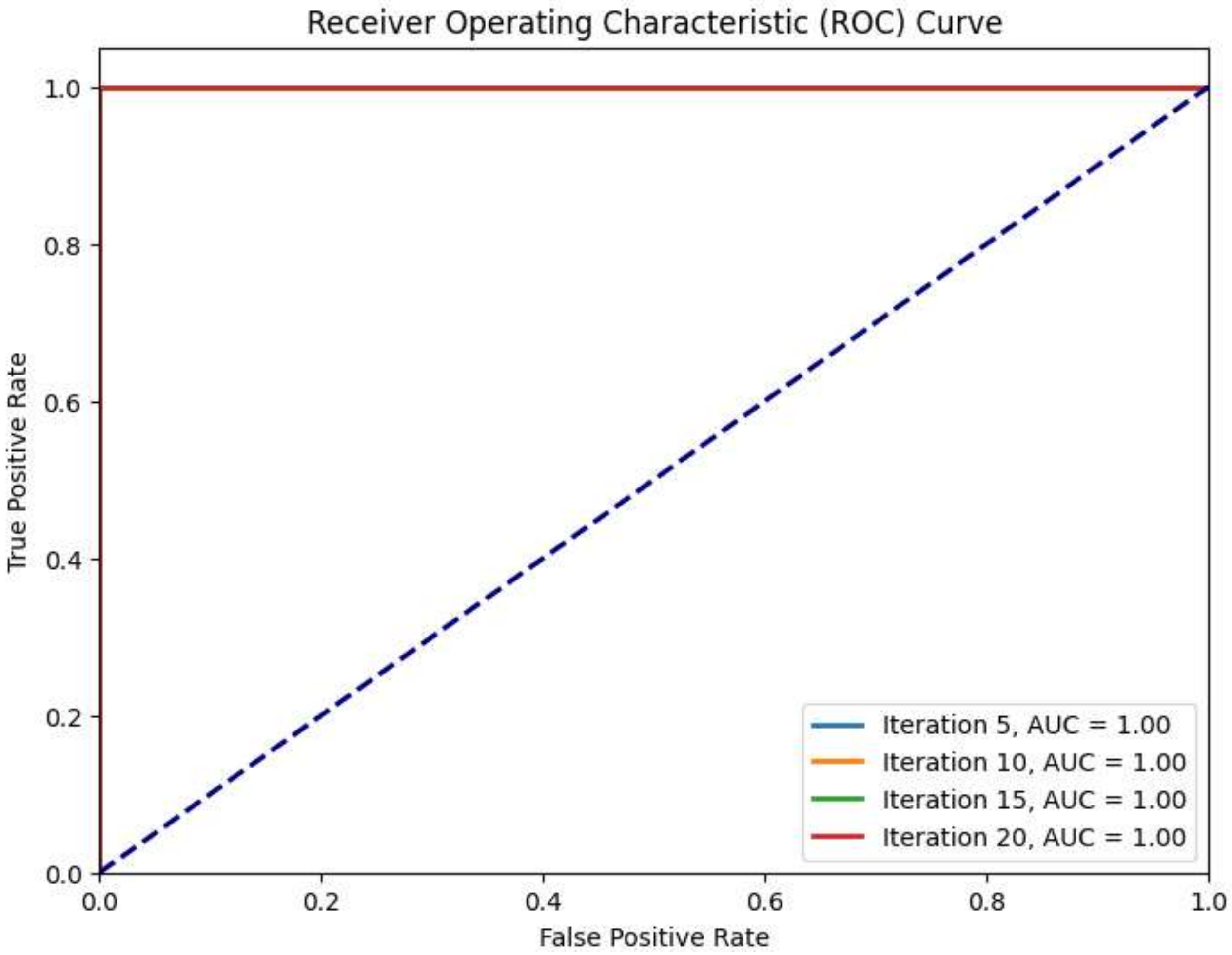
Iterations	Parameters	Accuracy	Time (seconds)	Precision		recall		F1-score	
				class 0	Class 1	class 0	class 1	class 0	Class 1
5	{'bootstrap': False, 'max_depth': 40, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 8, 'n_estimators': 182}	0.9967	2000.88	1.00	1.00	1.00	1.00	1.00	1.00
10	{'bootstrap': False, 'max_depth': 40, 'max_features': 'auto', 'min_samples_leaf': 3, 'min_samples_split': 7, 'n_estimators': 158}	0.9968	3549.01	1.00	1.00	1.00	1.00	1.00	1.00
15	{'bootstrap': True, 'max_depth': 50, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 7, 'n_estimators': 173}	0.9972	5814.54	1.00	1.00	1.00	1.00	1.00	1.00
20	{'bootstrap': True, 'max_depth': None, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 93}	0.9984	3419.68	1.00	1.00	1.00	1.00	1.00	1.00

Observation:

From the table above, we can clearly see that the accuracy has been increasing gradually with each iteration, and for iteration 20, the accuracy score is 99.86%, indicating high accuracy when compared to other iterations. From the classification report, such as precision, recall, and F1-scores, shows 100% for both fake and true news which shows that the model is more accurate to distinguishing between fake and true news.

ROC curve analysis:

Model performance is evaluated by plotting a roc curve, which shows the model's performance. The ROC is plotted for each iteration, as shown in figure(). from figure we can observe that AUC is 100%, indicating that the model can differentiate between fake and true news precisely.



5.1.4 Long-Short term memory (LSTM)

Following successful tokenization and creating a architect for lstm, the model is provided with different batches (32,64,128) and an epoch of 5 in order to avoid overfitting . The model is applied to train and test data, and the results are shown in table().

Batch Size	Accuracy	Precision		Recall		F1-score	
		Class 0	class 1	Class 0	Class 1	class 0	Class 1
32	0.9827	0.98	0.99	0.99	0.98	0.98	0.98
64	0.9521	0.98	0.96	0.96	0.98	0.97	0.97
128	0.9774	0.99	0.98	0.98	0.99	0.98	0.98

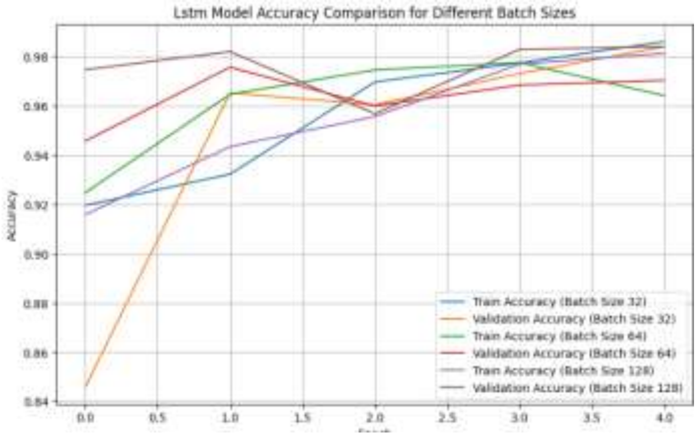
Observation:

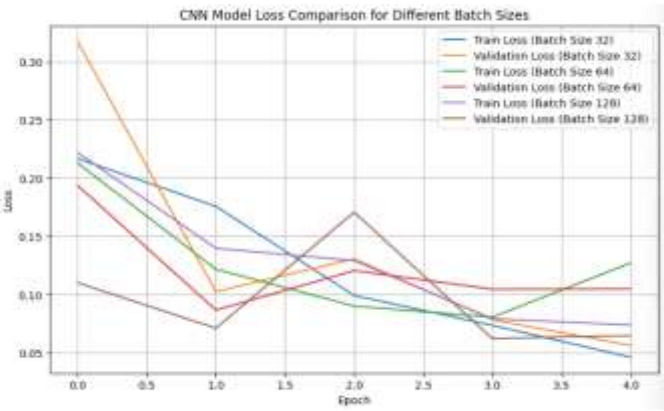
The above table shows that accuracy increases with batch size, and batch 128 has an high accuracy of 98.41% when compared to other batches. From Table() clearly shows that classification report scores for precision, recall, and F1-score change as batch size changes.

Analysis:

To evaluate tarin and val_ accuracy, we created a plot for each batch, as shown in figure(). The figure clearly shows that val_accuracy for 32 (batch size) is low compared to other batches. Accuracy and val_accuracy increased randomly with increase in batch size.

During model performance, there is some data loss; the loss and val_loss for each batch are shown in figure(), From plot we can observe that batch 32 has a higher loss than the others. As loos decreases with increase in batch size, which shows that model can differentiate between fake and true news with a high accuracy of 98.41 percent.





5.1.5 Convolutional neural network(CNN)

After implementing the model on tarin and test data, the model's performance is evaluated by giving a different batch size (32,64,128) and fitting an epoch of 5 to avoid overfitting, and the results are shown in table().

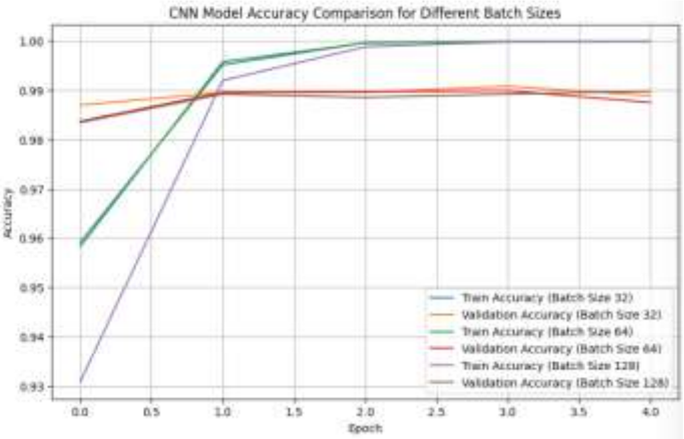
Batch Size	Accuracy	precision		Recall		F1-score	
		class 0	class 1	class 0	Class 1	class 0	Class 1
32	0.9889	0.99	0.99	0.99	0.99	0.99	0.99
64	0.9876	0.98	0.99	0.99	0.98	0.99	0.99
128	0.9898	0.99	0.99	0.99	0.99	0.99	0.99

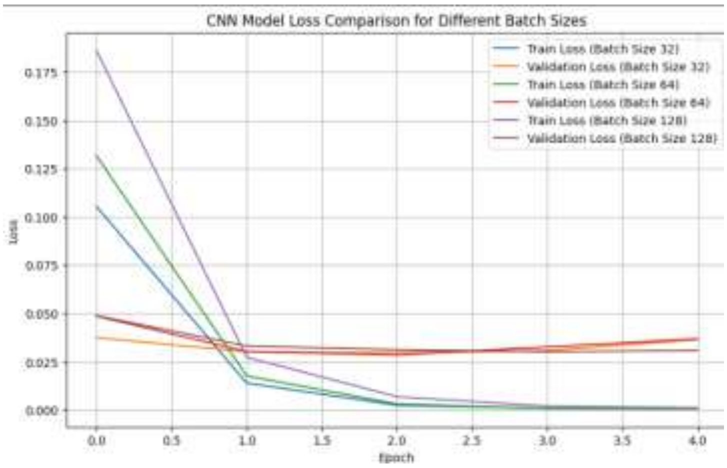
Observation:

From above table we can observe that at batch size of 32 and 128 shows more accuracy compare to batch size 64. As high accuracy is 98.98%, it shows that model is more accurate to differentiate between fake and true news and the classification report shows that the scores for precision, recall, and f1-score are 99% and 98%, respectively which indicates that model performance is good.

Analysis :

In order to understand the model performance, we made a plot for accuracy and val_accuraacy as shown in figure(), which shows that accuracy and val_accuraacy increased randomly with increase in batch size. In order to address loss in the model, we made a plot between loos and val_loss as shown in figure(), which shows there is no much loss in the model as high loss and val_loss in the model are 0.1% and 0.3%, respectively. This shows that the model is more accurate in predicting both fake and true news.





5.2 Comparison of models

We used three different classification models and two neural network models to differentiate between fake and true news. The classification models is given with parameters and performed using various iterations, and the best iteration results are noted in table (*). Similar to the classification model, we used two different neural network models, LSTM and CNN, and we trained and tested the data with various batch sizes. After comparing it to the batch size, the best size was chosen, and the results are noted as shown in table(), along with the scores for precision, recall, and F1-score. Based on the accuracy, a bar graph was plotted to compare all models, as shown in figure().

Models	Accuracy	Precison		recall		F1-score	
		class 0	Class 1	Class 0	Class 1	Class 0	Class 1
logistic regression	99.47%	0.99	1.00	0.99	0.99	0.99	0.99
Multinomial NB	93.20%	0.93	0.94	0.94	0.93	0.94	0.93
random forest	99.84%	1.00	1.00	1.00	1.00	1.00	1.00
LSTM	98.40%	1.00	1.00	1.00	1.00	1.00	1.00
CNN	98.98%	0.99	0.99	0.99	0.99	0.99	0.99

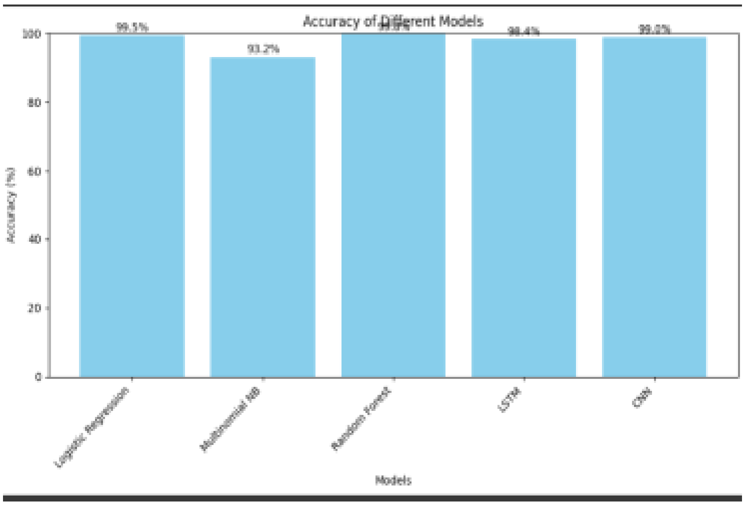


Figure () clearly shows that among classification models, random forest has a high accuracy of 99.8%. When comparing neural network models, the convolutional neural network exceeds Long-Short term memory with an accuracy of 99%.

6 CONCLUSION

In this study, we developed and evaluated machine learning models for efficiently distinguishing between fake and true news articles. While previous research focused mainly on linguistic and syntactic features, our methodology identified additional features important to improving model efficiency. . Through preprocessing steps and the exploration of text length, we gained valuable insights into our dataset.

We used three classifiers: logistic regression, Multinomial Naïve Bayes, and Random Forest. We trained the model using hyperparameter tuning with various iterations and evaluated there performance using accuracy metrics and ROC curves. Our results showed that all models achieved high accuracy scores, with Random Forest consistently outperforming the other models, achieving an accuracy of 99.84%.

Furthermore, we researched a deep learning approach using Long-Short term memory (LSTM) and Convolutional Neutral Network (CNN) by providing varying batch sizes. We observe that larger batches provide higher accuracy. These neural network models performed well, but CNN performed well in distinguishing between fake and true news with an accuracy of 98.98%.

While the results of this project are positive, there are numerous chances for future research to improve the effectiveness and adaptability of the fake news detection system. Advanced deep learning architectures, such as BERT (Bidirectional Encoder Representations from Transformers) and GPT(Generative Pre-trained Transformer) hybrid models that combine CNNs and RNNs, could yield higher precision. Increasing beyond textual data to include other data sources such as user profiles, and social-media data could provide useful context.

Using transfer learning methods to fine-tune pre-trained language models for this task may improve performance while reducing training time and by applying advanced techniques may provide insights into model limitations and decision-making processes and by continuous monitoring a model By addressing these areas of focus, we can help to develop more accurate and reliable fake news detection systems, which will lead to the reduction of misinformation in online environments.

REFERENCES

Shu, K., Sliva, A., & Tang, J. (2017). "Fake news detection on social media: A data mining perspective." ACM SIGKDD Explorations Newsletter, 19(1), 22-36.

Allcott, H., & Gentzkow, M. (2017). "Social media and fake news in the 2016 election." Journal of Economic Perspectives, 31(2), 211-236.

Chen, Y., & Conroy, N. J. (2015). "Towards automatic detection of deception in written language." In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers) (pp. 185-189).

Fake news detection on social media: A data mining perspective
K Shu, A Sliva, S Wang, J Tang, H Liu - ACM SIGKDD explorations …, 2017 - dl.acm.org

SOCIAL MEDIA AND FAKE NEWS IN THE 2016 ELECTION NATIONAL
Hunt Allcott and Matthew Gentzkow NBER Working Paper No. 23089 January 2017, Revised April 2017 JEL No. C52,C53,D7,H0,J60 ABSTRACT Following the 2016 U.S. presidential election, many

Accuracy of Deception Judgments - Charles F. Bond, Bella M ...
Charles F. Bond, Jr. and Bella M. DePaulo View all authors and affiliations. Volume 10, Issue 3. https://doi.org/10.1207/s15327957pspr1003_2

Automatic deception detection: Methods for finding fake news
Fake news detection is defined as the prediction of the chances of a particular news article (news report, editorial, expose, etc.) being intentionally deceptive (Rubin, Conroy & Chen, 2015)

Detecting Deceptive Opinions with Profile Compatibility
of detecting deceptive opinions. In the absence of gold-standard datasets, they trained models using features from the review texts,

Fake news detection on social media: A data mining perspective
K Shu, A Sliva, S Wang, J Tang, H Liu - ACM SIGKDD explorations …, 2017 - dl.acm.org

Improving fake news classification using dependency grammar
Kitti Nagy ,Jozef Kapusta

Fake News or Truth?
Using Satirical Cues to Detect Potentially Misleading News.
Victoria L. Rubin, Niall J. Conroy, Yimin Chen, and Sarah Cornwell

1703[1703.06959] CSI: A Hybrid Deep Model for Fake News Detection
Mar 20, 2017 · CSI: A Hybrid Deep Model for Fake News Detection. Natali Ruchansky, Sungyong Seo, Yan Liu. The topic of fake news has drawn attention both from the public and the academic

Development of Fake News Model using Machine...
Jan 19, 2022 · Development of Fake News Model using Machine Learning through Natural Language Processing. Sajjad Ahmed, Knut Hinkelmann, Flavio Corradini. Fake news detection

APPENDIX A

[The text within the square brackets must be deleted along with the square brackets themselves when finalising this part.

Each Appendix should be identified with a letter (A, B, C, …).

You should not assume that people will read an appendix unless they are directed to do so in the main body of the report, and even then you should make the Main Body self-contained as the appendix may not be read in detail; the appendix is not a way of subverting the word count limit of the Main Body of the report. You can direct a reader to an appendix with a phrase such as “see appendix A for further details”.

The following list of documents **must** be included as appendices:

- Project Proposal including your original plan
- A copy of Completed and approved BU Research Ethics Checklist
- First Progress Review Report
- List of the contents of the Large File (code/artefact) submission on Brightspace
- Artefact (e.g., System Requirements Specifications, Company Reports, Network Designs, Software Designs, Test Plans and Results, Business Plans, etc.)

The following list of documents **could** be included as appendices:

- Important communications between you and your client (private data should be deleted and all data should be anonymised)]
- Extra background information that is not included in the main body but is helpful to understand the project such as a brief summary of the used methods or techniques.
- Revised project proposals and plans.
- Any other relevant information, please discuss with your supervisor first.