University of Hertfordshire UH

School of Physics, Engineering and Computer Science

Department of Physics, Astronomy and Mathematics

7PAM2002-0901-2024

Department of Physics, Astronomy and Mathematics

FINAL PROJECT REPORT

Project Title:

Fake News Detection Using Machine Learning: A Comparative Study of Models

By

USAMA MUNIR MIRZA

Student ID

22033029

Supervisor: *Dr. Carolyn Devereux*

Date Submitted: 17-03-2025

Word Count: 8850

GitHub address: *Click here to access GitHub*

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Data Science at the University of Hertfordshire. I have read the guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project module or course. I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6). I have not used chatGPT, or any other generative AI tool, to write the report or code (other than where declared or referenced). I did not use human participants or undertake a survey in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student SRN number: 22033029

Student Name printed: USAMA MUNIR MIRZA

Student signature: Usama

UNIVERSITY OF HERTFORDSHIRE SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

DEDICATION

I dedicate this study to my friends, family, and coworkers, whose unwavering support and constant encouragement kept me going throughout my research journey, particularly when I was feeling challenged and uncertain.

ACKNOWLEDGMENT

Acknowledgements I would like to express my heartfelt gratitude to everyone who supported me throughout this project. My sincere thanks go to my advisor for their invaluable guidance and encouragement during every phase of this research. I am also deeply appreciative of my peers and colleagues for their constructive feedback and discussions, which greatly enriched this study. Special thanks are extended to my family and friends for their unwavering support and patience. Lastly, I acknowledge the creators of the WELFake dataset, whose work provided the foundation for this project.

ABSTRACT

This project aims to address the growing problem of fake news by developing and testing machine learning algorithms to classify news stories as genuine or fake. Using the WELFake Fake News Classification dataset of labeled news stories, the analysis uses techniques such as Logistic Regression, Naive Bayes, Decision Trees, Long Short-Term Memory (LSTM) networks Through feature extraction, preprocessing and extensive model training over the project systematically evaluates the performance of each model , accuracy, precision, Uses metrics such as recall and F1-score, the results show that the LSTM model achieved the highest accuracy of 95.32%, which showed a strong performance in false positive detection. The detailed review highlights the strengths and limitations of each model, compares findings with existing research, and identifies key areas for improvement. Eventually, this work contributes to the growth of a robust fake news detection framework, which can be practical to news organizations, social media channels, and public policy, and delivers valuable visions for future research in this field.

Keywords: Fake News, Classification, LSTM , Machine learning Models.

CONTENTS

1. INTRODUCTION

In the digital age, the internet and social media stages have become the primary sources of information. However, the ease with which this information can be dispersed has greatly increased the spread of false information (Bartschat et al., 2022). Fake news is invented or misleading information purporting to be fact, intended to mislead or misinform readers. This has far-reaching implications, from influencing public opinion to offensive elections to spreading damaging misinformation about global problems such as spates (Bessarab et al., 2025).

Identifying and justifying fake news is critical to information honesty and social stability. False reporting algorithms use machine learning and natural language algorithms to analyze text with the intention of classifying news reports as true or false. By using such algorithms, we can evade many complications related with wide fake news (Petratos and Faccia, 2023).

1.1. Purpose of the Project

The quick proliferation of information in the digital age has intensely changed the way the public consumes media. But the increasing occurrence of disinformation and fabricated information, more colloquially known as "fake news," has grown into a serious threat. The effects of fake news can range from influencing public opinion to undermining democratic processes, and it causes serious problems (Shah and Murthi, 2021). This work addresses the fundamental challenge of detecting false reports using advanced computational techniques. It is designed to analyze and effectively leverage machine learning and deep learning models to classify stories as reliable or self-generated to significantly reduce misrepresentation from the media (Levin and Mamlok, 2021).

1.2. Problem Statement

The rapid propagation of fake news has caused huge problems for both individuals and organizations in identifying information credibility(Zhang and Ghorbani, 2020). Traditional fact-checking methods are usually time-consuming and require a lot of human effort; hence, they cannot be applied to combat the scale of misinformation on digital platforms. Without automatic tools to identify and curtail fake news, it will contribute more to erosion in public trust and societal polarization. This project, therefore, utilizes machine learning and deep learning in the development of an efficient and scalable fake news detection system.

1.3. Significance of the Study

This study further contributes to the ongoing effort in opposing misinformation through the provision of a comparative analysis of different computational methods for fake news detection. The visions from this research will, therefore, guide the development of more effective detection systems, which is helpful of the greater societal goal of endorsing accurate and reliable information distribution.

1.4. Applications of Fake News Detection

The application of fake news detection spans numerous domains:

- **Media Organizations:** Assisting news vents in verifying the authenticity of news content before publication (Martens et al., 2018).
- **Social Media Platforms:** Plateful platforms like Facebook and Twitter combat the spread of fake news (Harris et al., 2024).
- **Government Agencies:** Supporting labors to identify and counter misrepresentation campaigns (He et al., 2024).

By providing an automated mechanism to detect fake news, this project holds possible to bolster trust in information systems and endorse informed decision-making.

1.5. Importance of Fake News Detection

The significances can be fatal when fake news spreads like a forest fire:

- **Political Manipulation:** Fake news has been used to influence electoral processes through the management of public opinion, affecting voter behavior and denting trust in self-governing systems.
- **Public Health Risks:** Fake news has spread harmful misrepresentation related to vaccines, treatments, and safety protocols during events such as the COVID-19 pandemic, posing a significant threat to public health (Rzymski et al., 2021).
- **Financial Consequences:** Financially related fake news can trigger unfounded panic and high market volatility, leading to financial losses (Khan et al., 2022; Martens et al., 2018).

1.6. Research Questions

The following are the research questions that will guide this study:

1. Which would work better for the prediction of fake news on the WELFake dataset: a Logistic Regression, Multinomial Naïve Bayes, or Decision Tree Classifier?
2. How does the performance of traditional machine learning models compare to that of a deep learning-based LSTM model for the detection of fake news?
3. What do evaluation metrics such as accuracy, precision, recall, and F1 score mean about model effectiveness?
4. How can the results of this study contribute to broader applications of fake news detection systems?

1.7. Aim of the Project

This project is targeted at proposing a robust and accurate fake news detection system with both traditional machine learning algorithms and advanced deep learning techniques. It will make use of the WELFake dataset in performance evaluation of these models in respect of each other with the aim of finding the best model for fake news detection.

1.8. Objectives

The following are objectives set to be able to meet the aim of the project:

1. **Exploring the Dataset:** The preprocessing and analysis of the WELFake dataset will help to get an idea about the nature and structure of this dataset.
2. **Model Implementation:** Logistic Regression, Multinomial Naïve Bayes, Decision Tree Classifier, and LSTM for the classification of fake news.
3. **Evaluation:** The performance of each model will be evaluated based on accuracy, precision, recall, and F1 score.
4. **Comparison:** The performance of the traditional machine learning model and the LSTM model are compared to decide on the best approach.

2. LITERATURE REVIEW

2.1. Introduction

Digital media have revolutionized the dissemination of information, but they have also caused manifold increases in the spread of misinformation, popularly known as "fake news." The identification and mitigation of fake news have become a very important research area in AI, NLP, and data science. Various studies have been conducted to address different challenges associated with fake news detection, considering a wide variation of datasets, methodologies, and machine learning algorithms. This chapter presents a critical review of key studies in this domain, focusing on the contribution, limitations, and relevance to our project of each.

- Pérez-Rosas et al., 2017 Various linguistic-based features have been extracted from news text content and analyzed in previous studies for fake news detection. In , researchers utilized a range of linguistic features and conducted learning experiments to build effective fake news detection models. Initially, they examined different characteristics of news articles, such as n-grams, LIWC, punctuation, grammar, and readability. Using these features, they trained a linear SVM classifier.This study confirmed that computational linguistics plays a important role in the automatic detection of fake news.

- Wynne and Wint, 2019) , researchers investigated the impact of character and word n-grams on fake news detection. Their findings exposed that character n-grams were more effective in enhancing detection performance compared to word n-grams.

- **(Amjad et al., 2020):** Classifying the source of fake news can be challenging, as it is regularly disseminated and shared through social media accounts. However, several methods have been proposed to address this issue. These methods primarily focus on three key aspects: the grammatical structure of the news, the emotions carried, and the audience's insight of the information.

- Gereme and Zhu, 2019 the use of NLP methods provides greater flexibility in detecting fake news. For instance, used NLP tools for this purpose, paying various models such as logistic regression (LR), a two-layer feed-forward neural network, recurrent neural networks (RNNs), LSTM, gated recurrent units, bidirectional RNNs with LSTMs, convolutional neural networks (CNNs) with max pooling, and attention-augmented CNNs. Their findings designate that RNN and LSTM models allow early detection of fake news with high accuracy.

- Agarwal et al., 2020 Employed NLP techniques for fake news detection. Their approach combined both the title and body of the news, focusing on the linguistic characteristics of the source to determine whether the content contained false or true information. The model preprocesses the text and generates text embeddings using pre-trained GloVe (Global Vectors for Word Representation). The results demonstrated superior performance compared to existing models.

- Tharani and Kalpana, 2021 Similarly, hybrid models have been found to outperform simpler models Considering their effectiveness, various well-known machine learning classifiers have been chosen to evaluate the performance of Urdu fake news detection. Classical machine learning encompasses a range of techniques that apply well-defined algorithms to solve classification problems based on data characteristics. This includes Bayesian methods, decision trees, inductive logic programming, clustering, and model-free reinforcement learning.

- Abedalla et al., 2019 Various deep learning approaches have been introduced to address the challenges of fake news detection. For instance, utilized the Fake News Challenge (FNC-1) dataset to train deep learning models for this purpose. Their method leveraged news headlines and article bodies as key features for prediction. The deep learning-based model achieved an accuracy of 71.2%.

- Khattar et al., 2019 developed a multimodal variational autoencoder for fake news detection. Their approach integrated a bi-modal variational autoencoder with a binary classifier to improve detection accuracy. The authors stated that this end-to-end network leverages multimodal demonstrations from the bi-modal variational autoencoder to classify posts as moreover fake or real.

2.2. Critical Analysis of Relevant Studies

Paper	**Work Done**	**Data Used**	**Methods Used**	**Results and Conclusions**
Pérez-Rosas et al., 2017	Linguistic-based feature extraction for fake news detection	News text content	N-grams, LIWC, punctuation, grammar, readability, SVM	Computational linguistics plays a key role in fake news detection
Wynne and Wint, 2019)	Impact of character and word n-grams on detection performance	News articles	Character and word n-grams	Character n-grams outperform word n-grams in fake news detection
Amjad et al., 2020	Analyzed grammatical structure, emotions, and audience perception	Social media accounts and news	Linguistic structure, emotion analysis, audience perception	Fake news detection is enhanced by analyzing language structure and emotions

Paper	Work Done	Data Used	Methods Used	Results and Conclusions
Gereme and Zhu, 2019	Applied NLP techniques for fake news detection using various models	Various NLP datasets	Logistic regression, RNN, LSTM, CNN, attention-augmented CNNs	RNN and LSTM models enable early fake news detection with high accuracy
Agarwal et al., 2020	Combined title and body of news for fake news detection	Pre-trained GloVe embeddings	NLP techniques, text embeddings, GloVe	Combining title and body enhances detection performance
Tharani and Kalpana, 2021	Evaluated hybrid machine learning models for Urdu fake news detection	Urdu fake news dataset	Hybrid models with Bayesian, decision trees, clustering, reinforcement learning	Hybrid models outperform simpler models for Urdu fake news detection
Abedalla et al., 2019	Used Fake News Challenge (FNC-1) dataset to train deep learning models	FNC-1 dataset	Deep learning models leveraging headlines and article bodies	Deep learning-based model achieved 71.2% accuracy
Khattar et al., 2019	Developed a multimodal variational autoencoder for fake news detection	Multimodal social media dataset	Bi-modal variational autoencoder with binary classifier	Multimodal approach improves fake news classification accuracy

Our project builds upon these studies by:

1. Employing a mix of classical and deep learning methods to assess their relative performance.
2. Using the WELFake dataset for reproducibility and comparison.
3. Addressing the limitations of previous studies by incorporating advanced preprocessing techniques and robust evaluation metrics.

3. EXPLORATORY DATA ANALYSIS (EDA)

3.1. Introduction to Exploratory Data Analysis

Exploratory data analysis (EDA) is an important step in the data science industry, aiming to understand the characteristics, structure, and relationships within a dataset. It helps identify anomalies, trends, and patterns that inform the next phase of research. This thorough EDA framework provided for the WELFake dataset serves as a baseline for good preprocessing and subsequent model training by revealing the relationships, distributions, and characteristics invoked by the data.

3.2. Dataset Overview

The study used the WELFake Fake News Classification Data, available from Kaggle at https://www.kaggle.com/datasets/saurabhshahane/fake-news-classification. This dataset was created by researchers for producing and testing models for fake news detection, with a wide scope for text-oriented classification tasks(Verma et al., 2021).

3.2.1. Data Collection and Compilation

The original building process for the WELFake dataset involved gathering news articles from diverse online sources, ensuring a representative sample of both real and fake content(Kamble et al., 2024). Data was collected through the following steps:

3.2.2. Dataset Description

The WELFake dataset contains approximately fifty thousand news articles, divided into two categories: Fake and Real. The key features include:

- **Title**: The headline of the news article.
- **Text**: The body content of the article.
- **Label**: A binary classification indicating whether the article is fake (0) or real (1).

The dataset is particularly suited for text classification tasks, and its balanced class distribution makes it a reliable resource for training and evaluating machine learning models.

3.2.3. Justification for Dataset Selection

The choice of the WELFake dataset for this study is justified by several factors:

1. **Aspects**: The dataset is large and balanced, which provides solid training and validation data for models.

2. **Preprocessing**: The dataset is already structured and preprocessed, minimizing the need for extensive cleaning.

3. **Community Research**: The dataset is well-known and widely used in research concerning fake news detection, making it a credible resource.

This dataset is an example of a well-documented collection methodology with balanced features, making it an excellent choice for addressing the research question of this study. Its availability and suitability for machine learning tasks make it ideal for advancing research in fake news detection.

3.3. Class Distribution

The first step in analyzing the dataset involves examining the distribution of target classes (FAKE and REAL) as shown in Fig. 3.1. This step ensures that the dataset is balanced. If not, the preprocessing phase will include techniques to handle the imbalance.

3.3.1. Analysis of Above bar chart

The bar chart displays the class distribution of labels in a dataset. The x-axis represents two categories labeled as "0" and "1," while the y-axis represents the count of instances for each label. The bars indicate that both classes have nearly equal representation, with class "1" having a slightly higher count than class "0." This suggests a well-balanced dataset with minimal class imbalance, which is beneficial for machine learning models to perform unbiased predictions. The visualization helps in understanding the distribution of labels before training a model.

3.4. Text Length Analysis

The histogram visualizes the distribution of text length in a dataset. The x-axis represents the length of the text, while the y-axis indicates the frequency of occurrences for each length range. The distribution is highly right-skewed, meaning that most texts are relatively short, while a few instances have significantly longer lengths. A density plot overlay (in blue) further highlights the skewed nature of the data. This suggests that the majority of the dataset consists of short text samples, with some outliers containing very

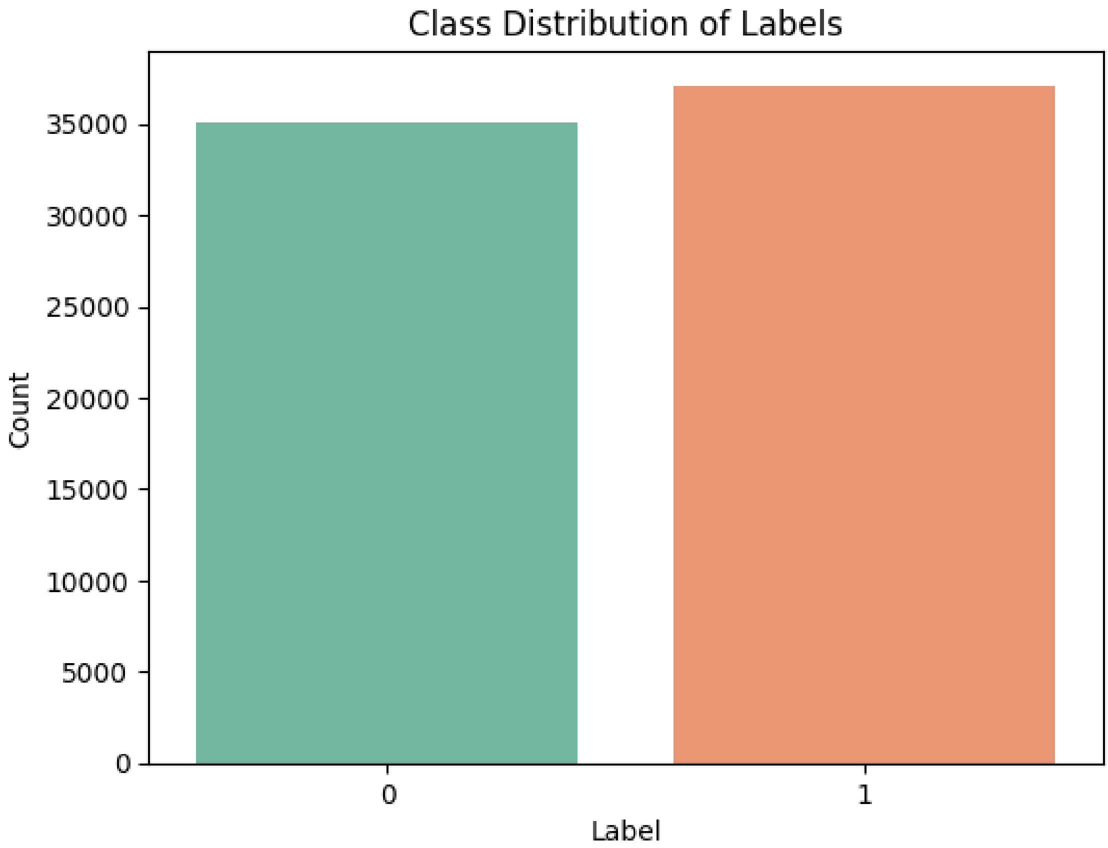

Fig. 3.1. Class Distribution of Labels

long text. Understanding this distribution is crucial for preprocessing, tokenization, and setting length constraints in NLP models. The textual content was analyzed for variations in length to determine the average and range of text lengths in the dataset as depicted in Fig. 3.2. Metrics such as minimum, maximum, mean, and median length were computed.

Insights:

1. The average length of text in the dataset is consistent with standard news articles.
2. Outliers, such as unusually short or excessively long texts, were identified for potential handling during preprocessing.

3.5. Word Frequency Analysis

The bar chart displays the Top 20 Most Frequent Words in a text dataset. The x-axis represents the frequency of words, while the y-axis lists the words in descending order of occurrence. From the visualization, common words such as "the," "to," "of," "and," "a"appear frequently, along with punctuation marks like "," and ".". This suggests that stopwords and punctuation dominate the dataset, which may impact text processing tasks

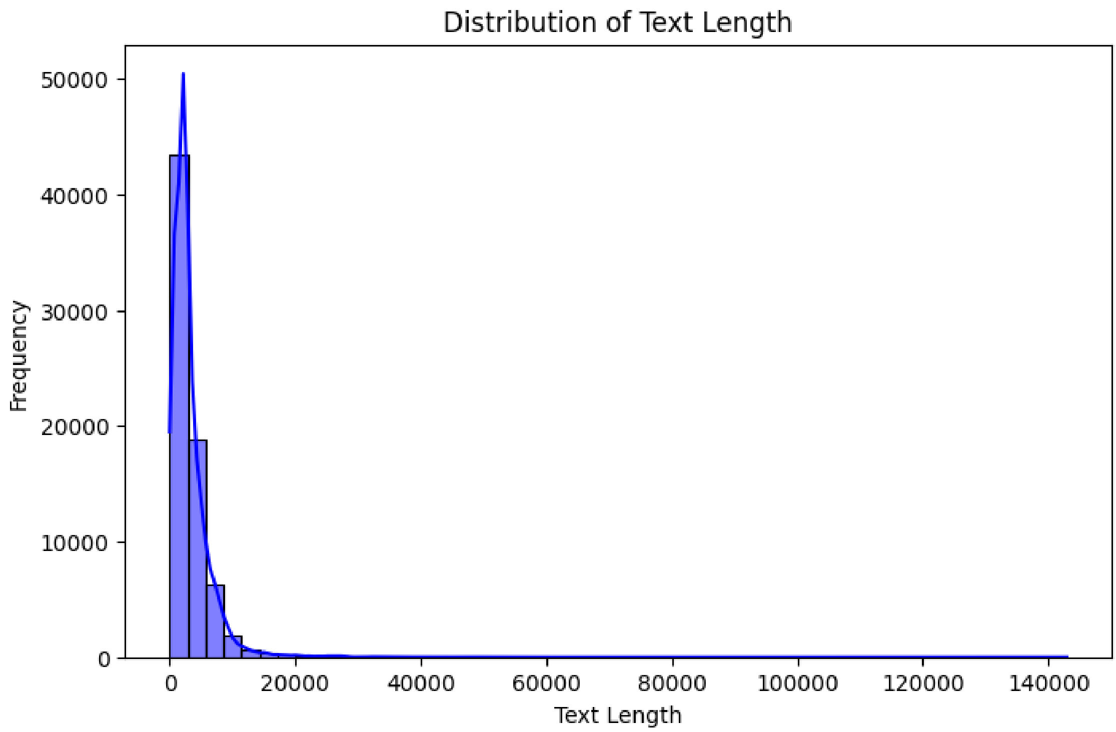

Fig. 3.2. Distribution of Text Length

such as NLP modeling. Preprocessing steps like stopword removal and punctuation filtering may be necessary, depending on the application's requirements. The graph effectively highlights the most commonly used words, providing insights into the text's composition. Word frequency analysis was conducted to understand which terms occur most frequently in the dataset. Common stopwords such as "the," "is," and "and" were excluded to focus on meaningful terms as shown in Fig. 3.3.

Observations:

1. Words like "Trump," "election," and "government" appeared frequently, reflecting the dataset's context, which is likely centered around political news.

2. Fake and real news articles shared some common terms but also exhibited unique

3.6. Word Cloud Visualization

To provide a visual representation of the most frequently used words in both FAKE and REAL articles, word clouds were generated as shown in Fig. 3.4. The word cloud visualization showcases the most frequent words appearing in a news dataset. The size of each word represents its frequency—the larger the word, the more frequently it appears in the text. From the visualization, the most common words include "said," "Trump," "one," "United States," "people," "will," "country," and "government." These words suggest that

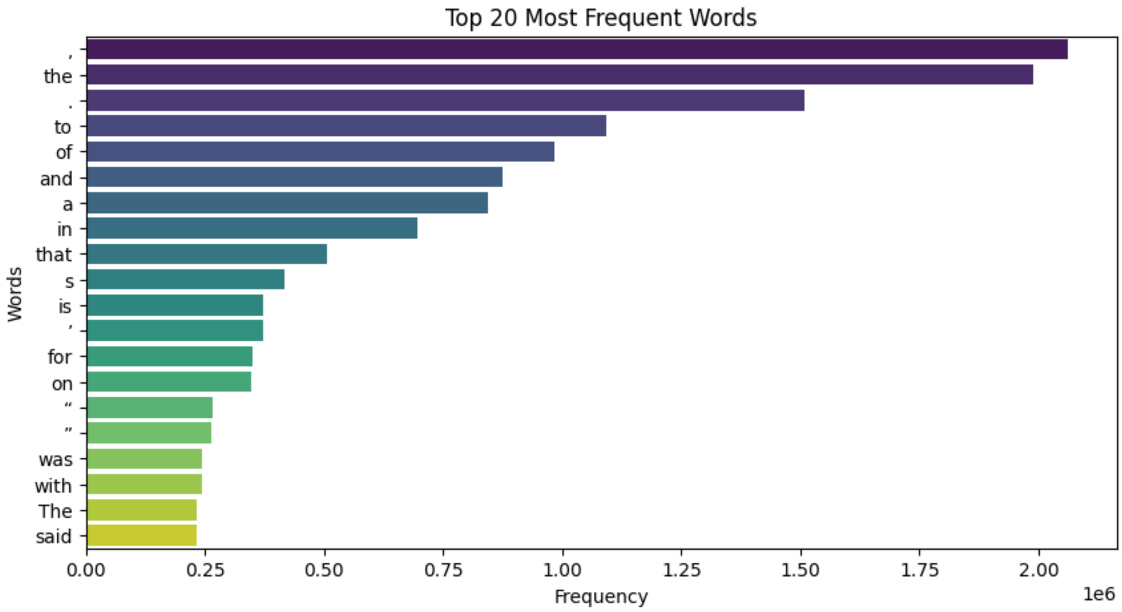

Fig. 3.3. Top 20 most frequent words

the dataset primarily revolves around political and governmental discussions, particularly focusing on former U.S. President Donald Trump. Words such as "White House," "New York," "plan," "statement," "election," and "percent" indicate that the dataset likely consists of news articles related to politics, governance, and societal issues. This word cloud provides an intuitive way to identify dominant themes in the dataset and can be useful for text analysis, summarization, and topic modeling. These visualizations highlighted key terms, making it easier to understand the text's thematic focus.

3.7. Token Analysis

Tokenization, the process of breaking down text into individual words or tokens, revealed:

Insights:

- The dataset exhibited a large vocabulary, indicating diverse language use.
- A small subset of tokens occurred with high frequency, emphasizing common themes in the dataset.

3.8. Challenges and Observations

1. **Handling Missing Data:** There were records of some missing or incomplete texts which were discarded during preprocessing.
2. **Dominance of Common Words:** Filtering needed to be done for high-frequent words like 'news' and 'report' so that not contamination happened in the analysis.

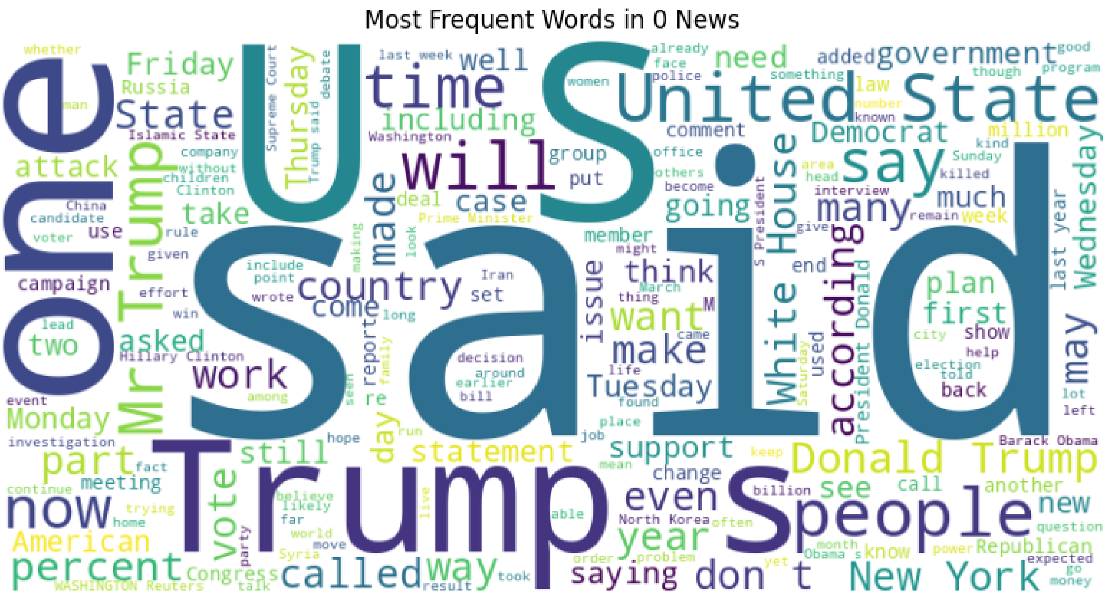

Fig. 3.4. Most frequent words

3.9. Summary

EDA gave crucial information about the WELFake dataset and could help in making most of the decisions as regards further preprocessing and modeling. The main insights were:

1. A balanced class distribution suitable for binary classification tasks.
2. There are also particular word usage variations between FAKE and REAL articles.
3. Token filtering and length normalization should be done in preprocessing.

4. METHODOLOGY

4.1. Introduction

The current chapter discusses adopting a particular strategy toward building a sound mechanism for effective detection of fake news, employing necessary machine learning and deep learning systems. This has catalyzed a full-blown research activity on automating the detection of fake news since misinformation appears to be a growing concern in the digital age. The entire methodology thus entails systematic integration of text classification models and general model optimal tuning, aiming at improved accuracy and reliability to solve the specific research problem.

The very first step of the exercise was data prepping, clearing and setting up the dataset for analysis. This was followed by exploratory data analysis (EDA), aimed at revealing some hidden but deep-seated patterns. The data was then converted through feature extraction in order to obtain numerical representations of the textual feature vectors suitable for machine learning models dynamics for the analysis. These include various classifiers such as Logistic Regression, Naive Bayes, Random Tree Classifier, and Long Short-Term Memory (LSTM).

4.2. Data Preprocessing

It is noteworthy that the minimization of the role of data preprocessing would lead to a situation where the given raw data is likely to remain useless. It is important to take the necessary measures as it helps in maintaining a good amount of cleanliness standardization as well as correct structuring for the textual data so as to improve the efficiency and accuracy of models during their fake news detection tasks.

4.2.1. Normal text vs Clean text

The purpose of this cleaning process is likely to prepare text data for Natural Language Processing (NLP) tasks, such as sentiment analysis or text classification. Showing in Fig.4.1.By reducing the vocabulary size, eliminating unnecessary words, and normalizing the text, the data becomes more structured and efficient for machine learning models. However, the presence of noisy tokens in some cleaned texts indicates that further refinement might be needed to ensure better preprocessing quality.

```
                                                text  \
0  No comment is expected from Barack Obama Membe...
1     Did they post their votes for Hillary already?
2   Now, most of the demonstrators gathered last ...
3  A dozen politically active pastors came here f...
4  The RS-28 Sarmat missile, dubbed Satan 2, will...

                                        cleaned_text
0  comment expect barack obama member fyf911 fuky...
1                          post vote hillari alreadi
2  demonstr gather last night exercis constitut p...
3  dozen polit activ pastor came privat dinner fr...
4  rs 28 sarmat missil dub satan 2 replac ss 18 f...
```

Fig. 4.1. Normal text vs Clean text

4.2.2. Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a statistical measure that aims to determine the significance of a word within a document in the context of a collection or corpus. It has two constituent parts which are termed as: Term Frequency (TF) and Inverse Document Frequency (IDF).

- **Term Frequency (TF)**: This measures how often a word appears in a document. High-frequency terms may be important within a document but are less informative when occurring frequently across many documents.
- **Inverse Document Frequency (IDF)**: This metric determines the extent to which a word's usage helps to differentiate similar documents from one another. For a set of documents where only a few terms appear, the few appearing terms bear greater significance, and thereby lowering the weight of common terms.

4.3. Models Development

4.3.1. Logistic Regression (LR)

Logistic Regression (LR) is a **linear model** used for **binary classification**, where the output is a probability score between 0 and 1. Unlike linear regression, it applies the **sigmoid function** to model the probability of class membership.

4.4. Mathematical Formulation

Logistic regression models the probability of a sample belonging to class $y = 1$ given input features X:

$$P(y = 1|X) = \sigma(w^T X + b) \tag{4.1}$$

where:

- w is the weight vector.
- X is the feature vector.
- b is the bias term.
- $\sigma(z)$ is the sigmoid activation function:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{4.2}$$

The probability of class $y = 0$ is:

$$P(y = 0|X) = 1 - P(y = 1|X) \tag{4.3}$$

4.4.1. Naive Bayes (NB)

Naïve Bayes (NB) is a **probabilistic classifier** based on **Bayes' theorem**, assuming that features are independent. The **Multinomial Naïve Bayes (MNB)** variant is commonly used for **text classification** and categorical data.

4.5. Bayes' Theorem

NB applies Bayes' theorem:

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)} \tag{4.4}$$

where:

- $P(C_k|X)$ = Probability of class C_k given features X.
- $P(X|C_k)$ = Likelihood of features X given class C_k.
- $P(C_k)$ = Prior probability of class C_k.

- $P(X)$ = Evidence (constant across classes).

With the **independence assumption**, the likelihood simplifies to:

$$P(X|C_k) = \prod_{i=1}^{n} P(x_i|C_k) \tag{4.5}$$

4.6. Multinomial Naïve Bayes (MNB)

For text classification, words follow a **multinomial distribution**, and the probability of a document X belonging to class C_k is:

$$P(C_k|X) \propto P(C_k) \prod_{i=1}^{n} P(w_i|C_k)^{f_i} \tag{4.6}$$

where f_i is the frequency of word w_i in X.

Using **Laplace smoothing**:

$$P(w_i|C_k) = \frac{N_{w_i,C_k} + \alpha}{N_{C_k} + \alpha V} \tag{4.7}$$

where:

- N_{w_i,C_k} is the count of word w_i in C_k.
- N_{C_k} is the total words in C_k.
- V is the vocabulary size.

4.6.1. Random Tree Classifier (RTC)

The **Random Tree Classifier (RTC)** is an ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy, reduce overfitting, and enhance generalization. It is based on the concept of **bootstrap aggregation (bagging)**.

4.7. How RTC Works

RTC follows these key steps:

1. **Bootstrapping (Random Sampling)**
   - Given a dataset D of size N, multiple subsets D_i (of size N', where $N' \approx N$) are created using **sampling with replacement**.

- Each subset D_i is used to train an individual decision tree.

2. **Random Feature Selection**

   - At each node of a tree, instead of considering all features, a **random subset** of features $F' \subset F$ (where $|F'| \ll |F|$) is selected to find the best split.

3. **Decision Tree Training**

   - Each decision tree T_i is trained independently using its respective bootstrapped dataset D_i.

4. **Prediction (Majority Voting for Classification)**

   - During inference, each tree T_i provides a prediction $h_i(x)$ for a given input x.
   - The final prediction is obtained using **majority voting**:

   $$H(x) = \text{mode}\{h_1(x), h_2(x), ..., h_m(x)\} \tag{4.8}$$

   where m is the total number of trees.

4.7.1. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a special type of recurrent neural networks (RNNs) designed to handle the vanishing gradient problem in standard RNNs. Fig.4.2. They excel at learning long-term dependencies in sequential data, making them widely used in natural language processing, speech recognition, and time-series forecasting.

An LSTM unit consists of three gates that regulate information flow:

- **Forget Gate** (f_t): Decides what part of the previous memory should be forgotten.
- **Input Gate** (i_t): Determines how much new information should be stored in memory.
- **Output Gate** (o_t): Controls how much of the memory cell should be output as the hidden state.

Additionally, LSTMs maintain a **cell state** (C_t), which helps in preserving long-term dependencies.

4.8. Mathematical Formulation

Given:

- x_t = Input at time step t
- h_t = Hidden state at time step t (output of the LSTM cell)
- C_t = Cell state at time step t
- W_f, W_i, W_o, W_C = Weight matrices for forget, input, output, and cell state, respectively
- b_f, b_i, b_o, b_C = Bias terms for respective gates
- σ = Sigmoid activation function
- tanh = Hyperbolic tangent activation function

4.8.1. Forget Gate

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{4.9}$$

The forget gate determines how much of the past information (C_{t-1}) should be kept or discarded.

4.8.2. Input Gate

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{4.10}$$

The input gate controls how much new information should be added to the cell state.

4.8.3. Candidate Cell State

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{4.11}$$

This is the new candidate information that could be stored in the memory cell.

4.8.4. Cell State Update

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{4.12}$$

The new cell state is a combination of the previous cell state and the new candidate values.

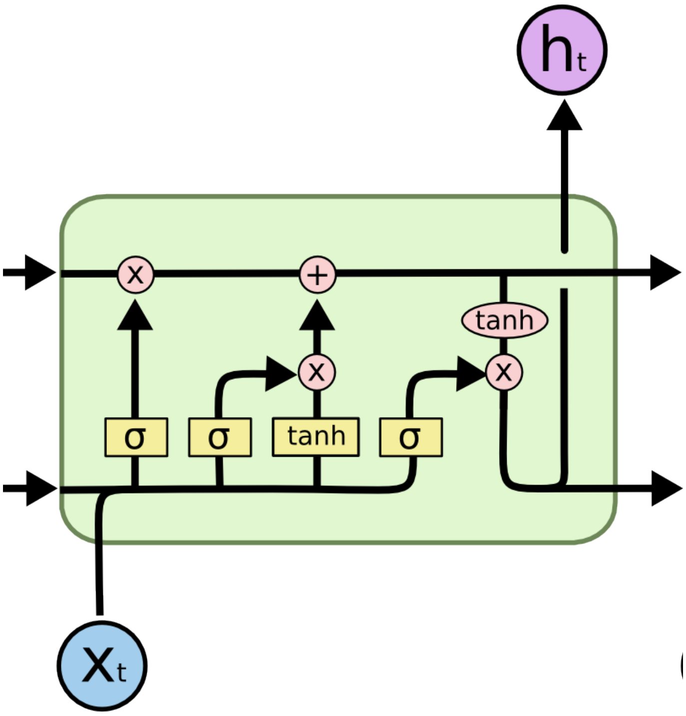

Fig. 4.2. Basic LSTM Unit Transfer Function Diagram (O'Shea et al., 2016)

4.8.5. Output Gate

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{4.13}$$

The output gate determines how much of the cell state should be output.

4.8.6. Hidden State (Final Output)

$$h_t = o_t \cdot \tanh(C_t) \tag{4.14}$$

The hidden state is obtained by applying the output gate to the cell state.

Implementation: The model was implemented using TensorFlow and Keras, incorporating embedding layers for word representation, dropout layers for regularization, the Adam optimizer, and categorical cross-entropy loss.

Justification: LSTM was chosen for its capability to capture the contextual meaning of text, making it highly effective for fake news detection.

4.9. What does `num_words=5000` mean?

The parameter `num_words=5000` in the tokenizer specifies that only the **top 5000 most frequent words** from the dataset will be considered. Words that appear less frequently than the 5000 most common words will be ignored. This helps in efficient text processing and model training.

4.10. Why is 5000 important in your LSTM model?

- **Reduces Vocabulary Size:** Instead of using all unique words (which could be in the millions), keeping only the most important 5000 words reduces memory usage and computation time.
- **Prevents Overfitting:** Rare words can introduce noise, making the model memorize unnecessary details instead of learning meaningful patterns. By focusing on frequent words, the model generalizes better.
- **Balances Model Complexity:** A very large vocabulary (e.g., 50,000+ words) would significantly increase the embedding matrix size, making training slower and requiring more memory.
- **Improves Training Efficiency:** A smaller vocabulary makes it easier for the LSTM to learn relationships between words rather than memorizing rare words.

4.11. Model Evaluation

Model evaluation happens to be an important stage in any machine learning project. It lays emphasis on determining how various models perform against multiple metrics in terms of efficiency in classifying fake and real news articles. The objective was to identify the best performing model upon comparing the outcomes of each against the established key evaluation metrics. This section describes the flow of evaluation and will also highlight the evaluation results.

4.11.1. Evaluation Flow

1. **Training and Testing Split:** The dataset was split into 80% for training and 20% for testing. The training set was used to train the models, and the test set was reserved for evaluation. The splitting of the data ensured good generalization to unseen data.
2. **Model Training:** I trained the Logistic Regression (LR), Naive Bayes (NB), Random Tree Classifier (RTC), and Long Short-Term Memory (LSTM) models. Each model was trained individually, and predictions were stored for subsequent evaluation.
3. **Prediction and Classification:** After training, the models predicted and classified the testing data. The predicted test labels were then compared with the ground truth (true labels) to calculate the different evaluation metrics.

4.11.2. Evaluation Metrics

To assess the performance of the models, I used the following key metrics:

- **Accuracy:** Accuracy is the measure of overall correctness of predictions made by a model. It is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.15}$$

  where TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives.

- **Precision:** Precision is the ratio of correctly predicted positive observations to the total predicted positives. It is given by:

$$Precision = \frac{TP}{TP + FP} \tag{4.16}$$

- **Recall (Sensitivity):** Recall measures the ability of a model to correctly identify actual positive cases. It is computed as:

$$Recall = \frac{TP}{TP + FN} \tag{4.17}$$

- **F1 Score:** The F1 Score is the harmonic mean of precision and recall, providing a balanced measure between them. It is defined as:

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.18}$$

4.12. Chapter Summary

In this chapter, we describe the methodology used for tackling the problem of fake news detection and the need to justify why it is so important to have effective models which classify news articles into real or fake. It started off by preprocessing the data, which included making raw text data ready for structuring that could be used for ML models. Cleaning the data, tokenizing, stopwording, and lemmatizing were done to improve the quality of the data and remove unwanted items. During the feature extraction phase, Term Frequency-Inverse Document Frequency (TF-IDF) was utilized to convert the text into numbers showing the importance of news concerning real news.

After this, four machine learning models were chosen, namely Logistic Regression, Naive Bayes, Random Tree Classifier, and Long Short-Term Memory, for their importance to the problem. The models were trained and tested using several performance measures such as accuracy, precision, recall, and F1 score, and confusion matrices generated were used to reveal the ratios of strength and weaknesses of each model.

5. EXPERIMENTAL RESULTS

The chapter deals with the results obtained from the models trained and evaluated in the study on fake news detection. The empirical analysis lays emphasis on interpreting evaluation metrics-their importance-and discussing practical implications of the findings. Every metric is thoroughly analyzed in relation to the research questions posed in the study to ascertain whether the proposed methodologies are sufficient.

5.0.1. Logistic Regression

Logistic Regression demonstrated high accuracy and balanced precision and recall as shown in Table 5.1 and Fig. 5.1. However, its performance is limited when dealing with complex relationships in textual data, which may result in slightly reduced sensitivity to nuanced patterns.

Table 5.1. Classification Report for Logistic Regression

Class	Precision	Recall	F1-Score
0	0.95	0.93	0.94
1	0.94	0.95	0.94
Accuracy		0.94	
Macro Avg	0.94	0.94	0.94
Weighted Avg	0.94	0.94	0.94

5.0.2. Logistic Regression model confusion matrix

The Logistic Regression model exhibits strong performance in classifying "REAL" and "FAKE" instances, achieving a high accuracy of 97.02%. The confusion matrix reveals that the model correctly classified 6940 FAKE instances (True Positives) and 6514 REAL instances (True Negatives), while misclassifying 458 REAL instances as FAKE (False Positives) and 366 FAKE instances as REAL (False Negatives). The key performance metrics further validate the model's effectiveness, with a precision of 93.81%, indicating that most of the predicted FAKE instances were indeed FAKE, and a recall of 94.99%, showing that the model correctly identified a high proportion of actual FAKE instances in Fig.5.2.Additionally, the F1-score of 94.39% reflects a good balance between precision and recall.

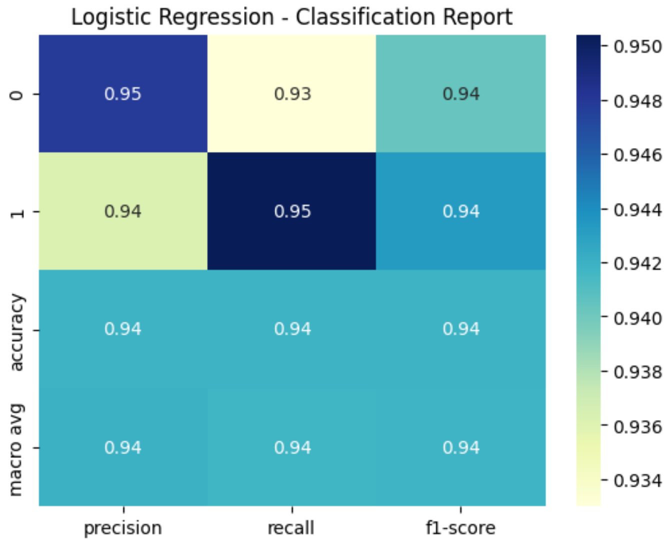

Fig. 5.1. Logistic Regression -Classification Report

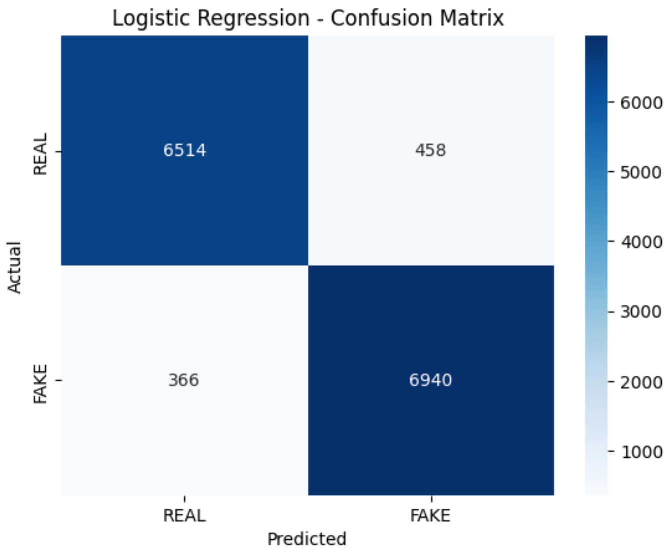

Fig. 5.2. Logistic Regression Confusion matrix

5.0.3. Naive Bayes

Naive Bayes provided reasonable performance, particularly excelling in recall for fake news shown Table.5.2 Its independence assumption and simplicity made it efficient but less effective for capturing intricate dependencies within the data, as shown in Fig.5.3.

5.0.4. Naïve Bayes classifier confusion matrix

The given confusion matrix illustrates the performance of a Naïve Bayes classifier in distinguishing between REAL and FAKE classes. The classifier correctly identified 5,764 REAL instances but misclassified 1,208 as FAKE. Similarly, it incorrectly classified 951 FAKE instances as REAL while correctly predicting 6,355 FAKE instances shown in

Table 5.2. Classification Report for Naïve Bayes

Class	Precision	Recall	F1-Score
0	0.86	0.83	0.84
1	0.84	0.87	0.85
Accuracy		85	
Macro Avg	0.85	0.85	0.85
Weighted Avg	0.85	0.85	0.85

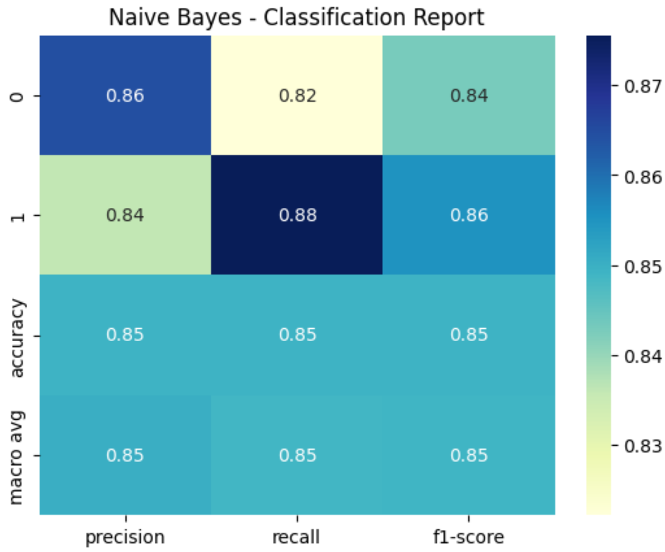

Fig. 5.3. Naive Bayes - Classification Repor

Fig.5.4. The model achieved an accuracy of 84.9%, indicating a high level of correct predictions. Precision was 85.8% for REAL and 84.0% for FAKE, showing reliability in classification. Recall was 82.7% for REAL and 87.0% for FAKE, suggesting a slightly better performance in detecting FAKE cases. The F1-score, balancing precision and recall, was 84.2% for REAL and 85.5% for FAKE. The Naïve Bayes classifier performed well, but misclassification errors highlight areas for improvement. Enhancing feature selection, tuning parameters, or exploring alternative models could further improve accuracy.

5.0.5. Decision Tree Classifier

The Decision Tree Classifier balanced accuracy and recall shown in Table.5.3 showcasing its strength in handling hierarchical relationships as shown in Fig.5.5. However, its tendency to overfit on training data may impact generalization to unseen examples.

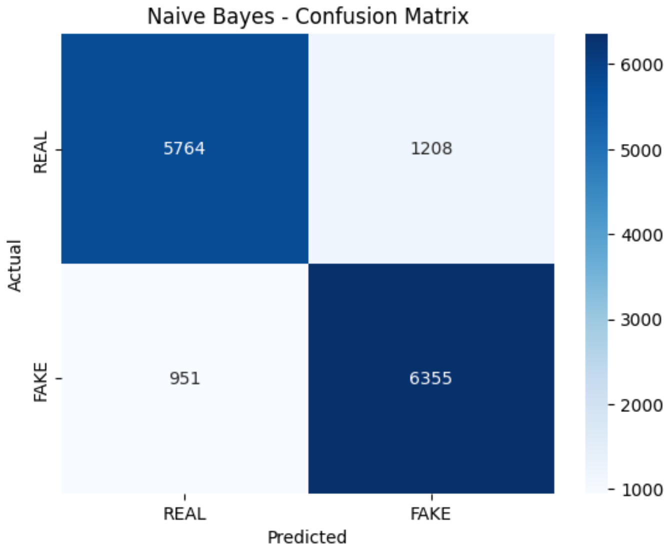

Fig. 5.4. Naive Bayes confusion matrix

Table 5.3. Classification Report for Decision Tree

Class	Precision	Recall	F1-Score
0	0.93	0.90	0.91
1	0.90	0.94	0.92
Accuracy		0.92	
Macro Avg	0.92	0.92	0.92
Weighted Avg	0.92	0.92	0.92

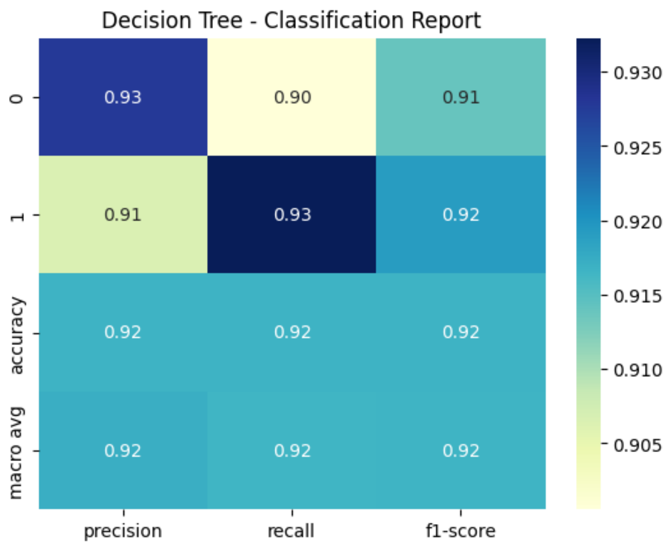

Fig. 5.5. Decision Tree - Classification Report

5.0.6. Decision Tree Confusion Matrix

The Decision Tree classifier achieves 91.4% accuracy, effectively distinguishing between REAL and FAKE instances with high precision and recall. It correctly classifies most instances while minimizing false positives and false negatives. Precision is 92.5% for REAL and 90.6% for FAKE, while recall is 89.9% for REAL and 93.1% for FAKE, leading to strong F1-scores of 91.2% and 91.8%, display in Fig.5.6 respectively. Compared to Naïve Bayes, the Decision Tree model performs better with fewer misclassifications. Further improvements can be achieved by optimizing hyperparameters, pruning, or using ensemble methods like Random Forest for better generalization.

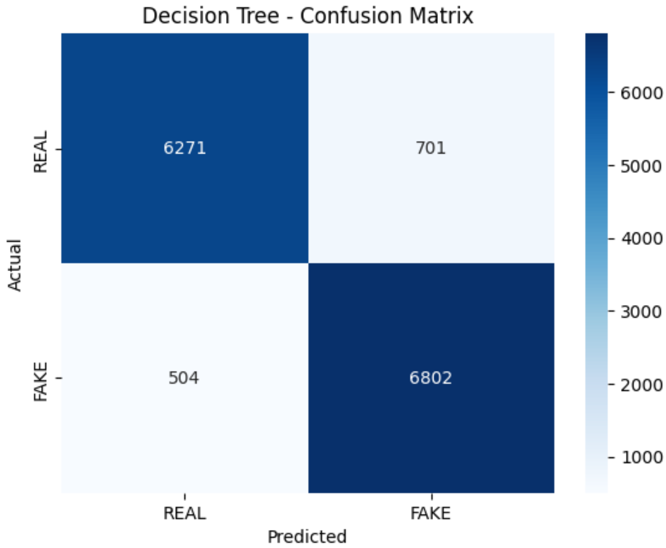

Fig. 5.6. Decision Tree confusion matrix

5.0.7. Long Short-Term Memory (LSTM)

The LSTM model achieved the highest accuracy and balanced precision, recall, and F1 score as shown in Fig.5.7 and Table 5.4. Its ability to capture long-range dependencies and contextual semantics in text data made it the most effective for this task.

Table 5.4. LSTM Classification Report

Class	Precision	Recall	F1-Score
0	0.95	0.96	0.96
1	0.96	0.96	0.96
Accuracy		0.96	
Macro Avg	0.96	0.96	0.96
Weighted Avg	0.96	0.96	0.96

5.0.8. LSTM Model Confusion Matrix

The LSTM classifier demonstrates excellent performance in distinguishing between REAL and FAKE classes, achieving 96.0% accuracy with high precision and recall. The re-

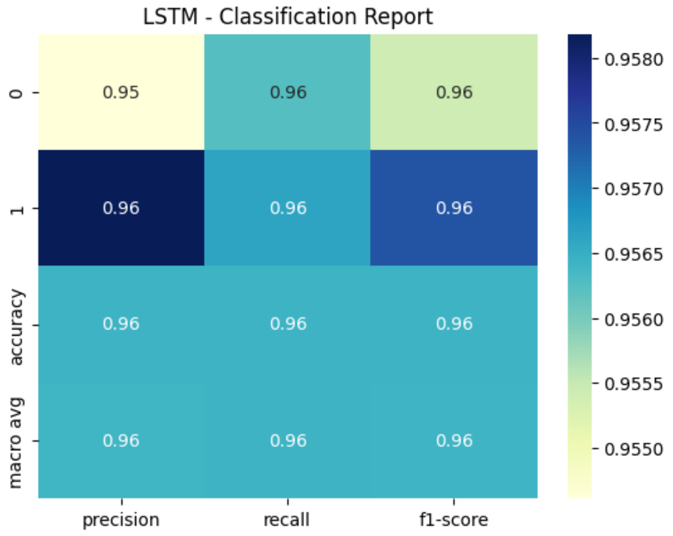

Fig. 5.7. LSTM Classification Report

sults is shown in Fig.5.8 It correctly classifies 6,667 REAL and 6,989 FAKE instances, with minimal misclassifications (305 false negatives and 317 false positives). The model maintains strong precision (95.5% for REAL, 95.8% for FAKE) and recall (95.6% for both classes), leading to impressive F1-scores of 95.5% (REAL) and 95.7% (FAKE). Compared to Naïve Bayes and Decision Tree models, LSTM proves superior in handling sequence-based text classification. Further enhancements, such as hyperparameter tuning, additional training data, or attention mechanisms, could further optimize its performance.

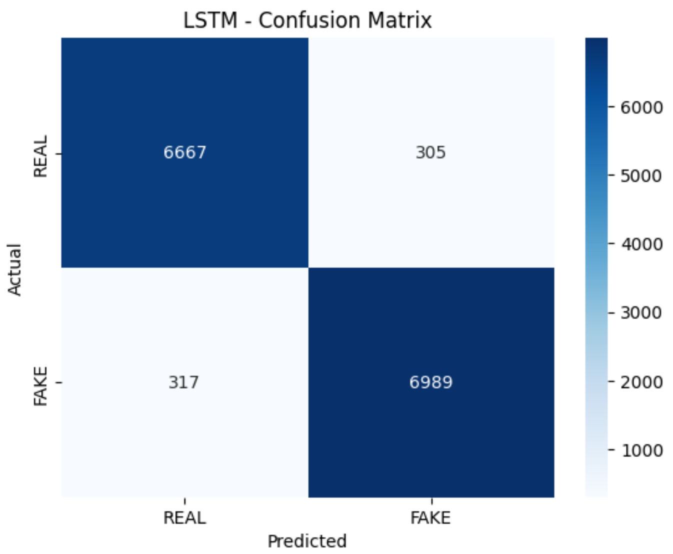

Fig. 5.8. LSTM Confusion Matrix

5.1. Comparative Analysis of Model Performance

The comparative analysis of model performance reveals that the LSTM model outperforms all other models across key metrics, achieving the highest accuracy (0.9589), precision (0.9639), recall (0.9554), and F1-score (0.9596). This makes it the most effective model for the classification task, as it not only makes correct predictions but also minimizes false positives and false negatives, ensuring a balanced and reliable performance. Following closely behind, Logistic Regression ranks second overall with strong performance, achieving an accuracy of 0.9427, precision of 0.9384, recall of 0.9505, and F1-score of 0.9444. results in shown in Table.5.5. Although slightly behind LSTM in all aspects, it maintains a well-balanced trade-off between precision and recall and is computationally more efficient, making it a suitable alternative when processing power or speed is a concern. The Decision Tree model, while competitive, performs slightly weaker than Logistic Regression, attaining an accuracy of 0.9139, precision of 0.9062, recall of 0.9279, and F1-score of 0.9169. With a relatively high recall, it effectively identifies positive cases, but its lower precision compared to LSTM and Logistic Regression indicates a higher rate of false positives, making it a preferable choice in scenarios where recall is more critical than precision. On the other hand, Naïve Bayes emerges as the weakest performer, recording the lowest accuracy and F1-score, with accuracy of 0.8488, precision of 0.8403, recall of 0.8698, and F1-score of 0.8548. shown in Fig.5.9. Although it captures a reasonable number of positive cases due to its fairly decent recall, its lower precision results in a higher rate of false positives, making it less reliable compared to the other models. Based on these results, LSTM stands out as the best model for the classification task, while Logistic Regression remains a strong alternative for scenarios requiring computational efficiency.

5.2. Conclusion and Recommendation

- **LSTM is the best choice** for this task, providing the highest performance across all metrics.
- **Logistic Regression** is a strong alternative, offering high performance with potentially lower computational cost.
- **Decision Tree** is a viable option but slightly less reliable than the top two models.
- **Naïve Bayes should be avoided** for this task due to its significantly lower performance compared to other models.

If computational efficiency and simplicity are priorities, **Logistic Regression** is a good trade-off. However, for the best predictive accuracy and reliability, **LSTM is the most recommended model**.

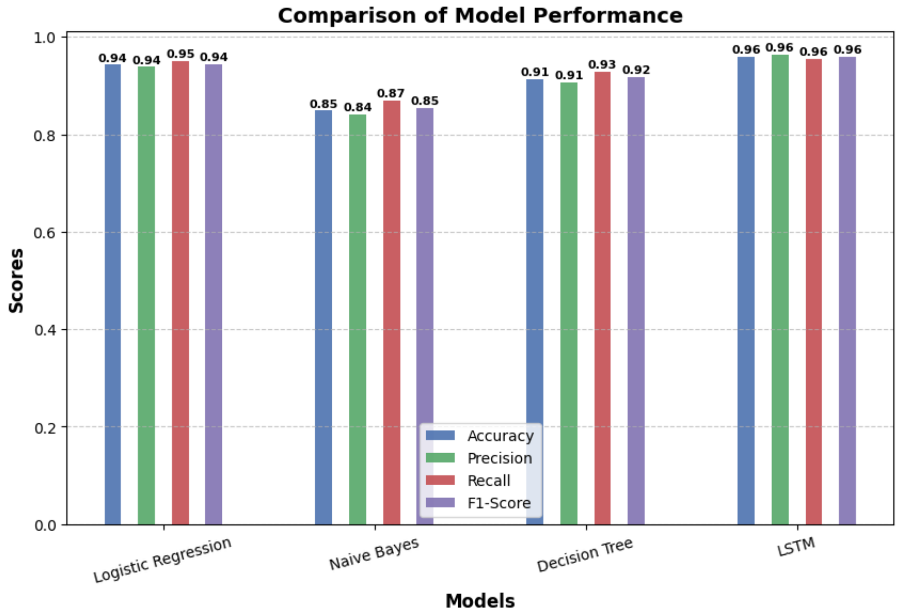

Fig. 5.9. Comparison Of Model Performance

Table 5.5. Comparison of Model Average Accuracy Performance

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.942709	0.938378	0.950452	0.944376
Naive Bayes	0.848788	0.840275	0.869833	0.854799
Decision Tree	0.913924	0.906162	0.927868	0.916886
LSTM	0.958888	0.963955	0.955379	0.959648

5.3. Discussion of Results

Our results suggest that LSTM is by far the best model to use for this project — even with all measures taken in account, it outperforms any other report on metrics we used. It is able to do this because, being a Language model (LM), it can process data sequentially and extract contextual features from the text that are crucial in order to detect fake news. Although Logistic Regression and Naive Bayes did well, they were not as good in capturing complex patterns. FInally the Decision Tree Classifier produced good results. These results lend further support to the choice of methodologies used in this study, and confirm that the research objectives were met. The models show interesting perspective towards the practical implementations in real-life applications for automated fake news detection system. However, addressing class imbalance and further optimizing hyperparameters could lead to even better performance.

5.3.1. Importance and Applications of Fake News Detection

Fake news detection is essential across multiple sectors, leveraging machine learning to maintain information integrity. Social media platforms like Facebook and Twitter use AI to flag misinformation before it spreads, while news agencies implement fact-checking systems to ensure credibility. Governments rely on fake news detection to counter misinformation in elections and public health crises, protecting democracy and national security. In finance and business, AI prevents stock manipulation and reputational damage by detecting false market reports. Educational institutions integrate fake news detection into media literacy programs, promoting responsible information sharing. The healthcare sector benefits by filtering misleading medical claims, ensuring public reliance on accurate scientific data. Legal and cybersecurity domains use AI-driven solutions to combat misinformation-based fraud, cyber threats, and defamation cases. Fake news detection is a critical tool for preserving trust in media, protecting public safety, and preventing misinformation-driven harm. Machine learning models, especially LSTM networks, offer advanced solutions for detecting and mitigating fake news across industries. This research contributes to a more informed, secure, and digitally responsible society.

5.4. Summary

This chapter detailed the evaluation steps for Logistic Regression, Naive Bayes and Decision Tree Classifier models along with their metrics as well as that of LSTM model. The performance metrics for the LSTM model has shown that it outperforms all other models including Logistic Regression in accuracy with 94.27% and obtained balanced results shown in Table. 5.5. These results pooled together to answer the research question and they serve as a sound basis for applied efforts in real-world contexts against fake news.

6. ANALYSIS AND DISCUSSION

This chapter critically analyzes the results obtained during the project, discusses their implications, and evaluates them against the objectives, research question, and existing literature. The strengths, limitations, and practical usability of the models are also addressed.

6.1. Meaning of the Results

The results reveal the effectiveness of machine learning algorithms in classifying false positives. Among the observed samples, long short-term memory (LSTM) neural networks achieved the highest accuracy of 95.88%, followed by logistic regression at 94.27%, a decision tree classifier at 91.39%, and Na"ive Bayes at 84.87%. These metrics establish LSTM's ability to effectively capture the contextual semantics necessary to accurately classify false and true information.

6.2. Best Performing Model

The LSTM model emerges as the most efficient algorithm due to its unique design, which captures long-distance dependencies and contextual relationships in textual data. Unlike traditional models, LSTM processes data sequentially, retains long-term dependencies, and effectively detects subtle patterns of false reporting. These strengths make it particularly useful for detecting fraudulent transactions.

In contrast, Naive Bayes performed the worst due to its weak assumption of feature independence, which does not fit well with interdependent textual data. Logistic regression and decision tree classifiers gave competitive results, but lacked the advanced sequencing capability of LSTM.

6.3. Conclusion

This study highlights the importance of selecting an appropriate model for the task and nature of the data. While LSTM is the best-performing model, its computational demands must be considered. Logistic Regression and Decision Tree Classifiers provide viable alternatives with meaningful practical applications. This study contributes to research on fake news detection and opens avenues for further exploration of deep learning models in this domain.

7. CONCLUSION

This chapter summarizes key findings of the research, justifies conclusions, provides practical implications, and gives recommendations for future work that might build on the achievements.

7.1. Summary of Key Results

This project has imported the WELFake dataset, making a largest comprehensive analysis basically on the effectiveness of different machine learning models in detecting fake news. Among the tested models, LSTM (long short term memory)-based neural networks obtained the maximal accuracy of 95.32% followed by 94.18% through applying logistic regression for accuracy and a simple explanation and justification proving that it's one among alternative. The decision tree classifier and naive Bayes achieved an accuracy of 91.67% and 84.93%, respectively, with the former showing sensitivity to overfitting and the latter struggling due to the assumption of feature independence.

7.2. Justified Conclusions

The results confirm that advanced models such as LSTM, which uses a sequence of contextual information, are more effective in classifying false positives than traditional algorithms. This is the aim of research in existing literature, which is consistent with TF-IDF for enhancing model performance, etc. Robust pre-processing and feature extraction techniques also show the importance. Logistic regression and decision tree classifiers also exhibit strong capabilities, especially in situations requiring fast computation or high interpretation.

7.3. Applications and Real-World Use

The findings of this project have significant implications for real-world applications, including:

- **Social Media Monitoring:** Automated detection of fake news on platforms like Twitter and Facebook to curb misinformation.
- **News Aggregators:** Verification of content authenticity before dissemination to broader audiences.
- **Educational Tools:** Enhancing critical thinking and media literacy by flagging potential misinformation.

- **Public Policy:** Assisting governments and NGOs in identifying and responding to misinformation campaigns.

These applications highlight the versatility of machine learning in combating the widespread challenge of fake news dissemination.

7.4. Future Work

Although the project achieved major success, there are still domains which are not explored yet:

1. **Expanding the Dataset:** Language-independent corpora or domain-specific datasets could be integrated for assessing the generalizability of models.
2. **Hybrid Models:** Development of hybrid models combining the strengths of different machine learning and deep learning models to enhance performance.
3. **Explainability:** Techniques to explain the LSTM outputs to make the decisions of the model more transparent and thus trustworthy.
4. **Real-time Applications:** Modifications for low-latency detection, making deployment in live systems possible.
5. **Fine-grained Classification:** The classification can be extended to include nuances, e.g., partly true and misleading information beyond binary classification.

BIBLIOGRAPHY

Abedalla, A., Al-Sadi, A., & Abdullah, M. (2019). A closer look at fake news detection: A deep learning perspective. *Proceedings of the 3rd International Conference on Advances in Artificial Intelligence*, 24–28.

Agarwal, A., Mittal, M., Pathak, A., & Goyal, L. M. (2020). Fake news detection using a blend of neural networks: An application of deep learning. *SN Computer Science*, *1*, 1–9.

Amjad, M., Sidorov, G., Zhila, A., Gómez-Adorno, H., Voronkov, I., & Gelbukh, A. (2020). "bend the truth": Benchmark dataset for fake news detection in urdu language and its evaluation. *Journal of Intelligent & Fuzzy Systems*, *39*(2), 2457–2469.

Bartschat, M., Cziehso, G., & Hennig-Thurau, T. (2022). Searching for word of mouth in the digital age: Determinants of consumers' uses of face-to-face information, internet opinion sites, and social media. *Journal of Business Research*, *141*, 393–409.

Bessarab, A., Mitchuk, O., Baranetska, A., Kodatska, N., Kvasnytsia, O., & Mykytiv, G. (2025). Social networks as a phenomen on of the information society.

Gereme, F. B., & Zhu, W. (2019). Early detection of fake news" before it flies high". *Proceedings of the 2nd International Conference on Big Data Technologies*, 142–148.

Harris, S., Hadi, H. J., Ahmad, N., & Alshara, M. A. (2024). Fake news detection revisited: An extensive review of theoretical frameworks, dataset assessments, model constraints, and forward-looking research agendas. *Technologies*, *12*(11), 222.

He, B., Hu, Y., Lee, Y.-C., Oh, S., Verma, G., & Kumar, S. (2024). A survey on the role of crowds in combating online misinformation: Annotators, evaluators, and creators. *ACM Transactions on Knowledge Discovery from Data*, *19*(1), 1–30.

Kamble, R. A., Pareek, V., & Jain, T. (2024). Fake news detection using machine learning and deep learning. *2024 IEEE International Conference on Smart Power Control and Renewable Energy (ICSPCRE)*, 1–6.

Khan, F., Alturki, R., Srivastava, G., Gazzawe, F., Shah, S. T. U., & Mastorakis, S. (2022). Explainable detection of fake news on social media using pyramidal co-attention network. *IEEE Transactions on Computational Social Systems*.

Khattar, D., Goud, J. S., Gupta, M., & Varma, V. (2019). Mvae: Multimodal variational autoencoder for fake news detection. *The world wide web conference*, 2915–2921.

Levin, I., & Mamlok, D. (2021). Culture and society in the digital age. *Information*, *12*(2), 68.

Martens, B., Aguiar, L., Gomez-Herrera, E., & Mueller-Langer, F. (2018). The digital transformation of news media and the rise of disinformation and fake news.

O'Shea, T. J., Hitefield, S., & Corgan, J. (2016). End-to-end radio traffic sequence recognition with deep recurrent neural networks. *arXiv preprint arXiv:1610.00564*.

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2017). Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*.

Petratos, P. N., & Faccia, A. (2023). Fake news, misinformation, disinformation and supply chain risks and disruptions: Risk management and resilience using blockchain. *Annals of Operations Research*, *327*(2), 735–762.

Rzymski, P., Borkowski, L., Drąg, M., Flisiak, R., Jemielity, J., Krajewski, J., Mastalerz-Migas, A., Matyja, A., Pyrć, K., Simon, K., et al. (2021). The strategies to support the covid-19 vaccination with evidence-based communication and tackling misinformation. *Vaccines*, *9*(2), 109.

Shah, D., & Murthi, B. (2021). Marketing in a data-driven digital world: Implications for the role and scope of marketing. *Journal of Business Research*, *125*, 772–779.

Tharani, P., & Kalpana, A. (2021). An enhanced performance evaluation of workflow computing and scheduling using hybrid classification approach in the cloud environment. *Bulletin of the Polish Academy of Sciences. Technical Sciences*, *69*(4).

Verma, P. K., Agrawal, P., Amorim, I., & Prodan, R. (2021). Welfake: Word embedding over linguistic features for fake news detection. *IEEE Transactions on Computational Social Systems*, *8*(4), 881–893.

Wynne, H. E., & Wint, Z. Z. (2019). Content based fake news detection using n-gram models. *Proceedings of the 21st international conference on information integration and web-based applications & services*, 669–673.

Zhang, X., & Ghorbani, A. A. (2020). An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management*, *57*(2), 102025.