PREMIER UNIVERSITY, CHATTOGRAM

Department of Computer Science & Engineering

Course Code: CSE 458

Course Title: Machine Learning Laboratory

Final Project Report

on

“Fake News Detection Using

Machine Learning and Deep Learning”

Date of Submission: 26.11.2025

Submitted to

Avishek Chowdhury
Lecturer
Department of Computer Science & Engineering
Premier University

Submitted by

Name	: Bobi Barua
ID	: 2104010202165
Semester	: 8th
Section	: A
Batch	: 40th
Session	: Spring 2025

Abstract

The widespread circulation of fake news on digital platforms has become a major concern, influencing public opinion and undermining trust in online media. This project aims to develop an automated system for detecting fake news using both traditional machine learning algorithms and deep learning models. The Fake and Real News Dataset from Kaggle was used, comprising labeled articles of true and false news. Text preprocessing techniques such as cleaning, normalization, and tokenization were applied to ensure high-quality input data.

Machine learning models including Logistic Regression**,** Naïve Bayes and Random Forest were trained using TF–IDF features, while deep learning architectures-LSTM**,** BiLSTM, and GRU- were implemented with pre-trained GloVe embeddings to capture semantic relationships. Experimental results revealed that deep learning models outperformed traditional ones, with the GRU model achieving the highest accuracy of 99.16%. To evaluate practical usability, the best-performing model was deployed as an interactive web application using Streamlit, allowing users to input text and receive real-time predictions along with confidence scores. The system proved responsive and user-friendly, demonstrating its potential as a supportive tool for combating misinformation.

Overall, the study demonstrates that recurrent neural networks, particularly GRU are highly effective for text-based fake news detection and can significantly improve automated misinformation filtering systems.

1. Introduction

In the digital age, online platforms have revolutionized how people consume and share information. With billions of users actively engaging on social media, news websites, and online forums, information dissemination has become faster and more accessible than ever before. However, this rapid exchange of content has also enabled the widespread circulation of fake news-false or misleading information intentionally created to deceive readers. The uncontrolled spread of such misinformation poses severe threats to society, influencing public perception, distorting facts, and even affecting political, social, and economic stability.

1.1 Background and Motivation

The rise of fake news has become a global concern, especially during major events such as elections, health emergencies, and social movements, where false information spreads rapidly across social media. Manual fact-checking is too slow to keep up with the massive volume of online content, creating the need for automated systems that can detect misinformation in real time.

From a technical standpoint, fake news detection is challenging because deceptive articles are often written to resemble legitimate news in tone, structure, and language. This makes it difficult to distinguish authentic content from fabricated stories, highlighting the need for computational models capable of analyzing subtle linguistic cues to identify misinformation effectively.

1.2 Significance of the Study

The significance of this project lies in its potential to enhance digital trust and information reliability. Fake news can manipulate public opinion, incite violence, and erode confidence in journalism. For example, during the COVID-19 pandemic, misinformation about treatments and vaccines caused widespread panic and confusion. Automated systems that can efficiently detect and flag fake content could help prevent such societal harm.

Moreover, this project contributes to the growing field of Natural Language Processing (NLP) and Artificial Intelligence (AI) by applying advanced text analysis and deep learning techniques to real-world problems. Machine learning (ML) and deep learning (DL) models, when trained on large datasets, can learn patterns in textual data that distinguish factual statements from deceptive ones. Integrating these models into social media monitoring tools or news aggregators could significantly reduce the spread of misinformation online.

1.3 Objectives of the Project

The main objective of this project is to design and evaluate an effective system for automated fake news detection using both traditional machine learning methods and advanced deep learning architectures. The specific goals are as follows:

1. **Data Acquisition and Preprocessing:** To collect and prepare a large-scale dataset of real and fake news articles, applying preprocessing techniques such as text cleaning, normalization, tokenization, and stopword removal to ensure consistency and quality.
2. **Feature Extraction and Representation**: To apply text vectorization methods like TF–IDF for traditional ML models and GloVe word embeddings for deep learning models to capture semantic relationships between words.
3. **Model Implementation and Evaluation:** To develop and compare the performance of machine learning algorithms (Logistic Regression, Naïve Bayes, Random Forest) and deep learning models (LSTM, BiLSTM, GRU) in classifying fake and real news.
4. **Performance Analysis:** To evaluate models based on metrics such as accuracy, precision, recall, and F1-score, identifying the most effective approach for text-based fake news classification.

1.4 Scope of the Project

This project focuses exclusively on text-based fake news detection, meaning that the models analyze only the written content of news articles. Non-textual cues such as images, videos, or user metadata are not included in this study. The dataset used originates from verified Kaggle sources containing labeled real and fake articles, ensuring reliability in model training and evaluation.While the project aims for high accuracy, it also recognizes limitations such as language bias, dataset imbalance, and contextual ambiguity in news reporting. Despite these challenges, this research contributes a practical framework that demonstrates the potential of machine and deep learning in combating misinformation at scale.

2. Dataset Description

The dataset used in this project is the Fake and Real News Dataset obtained from Kaggle. It contains labeled news articles categorized as either real or fake, making it suitable for supervised text classification tasks.

- **Dataset name:** Fake and Real News Dataset
- **Source:** Kaggle [1]
- **Language:** English
- **Type:** Text dataset

2.1 Dataset Size and Features

The dataset consists of two separate CSV files - one containing real news and the other containing fake news. These were combined into a single dataset for model training and evaluation.

File Name	Number of Records	Label
True.csv	21,417	Real news (label = 1)
Fake.csv	23,502	Fake news (label = 0)

After merging, the combined dataset contains 44,919 records**.** Each record includes the following features:

Feature	Description
title	The headline or title of the news article.
text	The main body of the article.
subject	The category of the news (e.g., politics, world news, tech).
date	The date when the article was published.
label	The target variable indicating whether the article is fake (0) or real (1).

The title and text columns were used as input features while the label column served as the target variable for classification.

2.2 Data Preprocessing

Before training the models, the dataset was preprocessed to ensure that it was clean, consistent, and suitable for both machine learning and deep learning models. The preprocessing steps implemented in this project are as follows:

- **Combining and Labeling:**
  The two CSV files (True.csv and Fake.csv) were combined into a single dataset. A new column named label was added to mark real articles as 1 and fake articles as 0.

- **Removing Duplicates and Missing Values:**
  Duplicate records and rows containing null values in the text or title fields were identified and removed to prevent bias during training.
    - **Class Distribution**: After preprocessing, the dataset consisted of 386727 unique samples, nearly balanced between fake news (17,433 samples) and real news (21,194 samples).
- **Dropping Irrelevant Columns:**
  The date and subject columns were dropped since they did not contribute meaningfully to the text-based classification.
- **Text Cleaning:**
  Each article's text was cleaned using regular expressions to remove unwanted elements such as punctuation, special characters, numbers, and URLs.
  All text was also converted to lowercase for consistency.
- **Tokenization and Sequence Preparation:**
    - For machine learning models**,** TF–IDF vectorization was applied to transform the cleaned text into numerical feature vectors.
    - For deep learning models**,** the text was tokenized and padded to a fixed sequence length to ensure uniform input size for neural networks.
- **Data Splitting:**
  The dataset was divided into training (80%) and testing (20%) sets to evaluate model performance objectively.

3. Methodology

This project focuses on developing a robust fake news detection system using both traditional machine learning (ML) techniques and advanced deep learning (DL) architectures. The methodology is divided into multiple stages: data preprocessing, feature extraction, model selection, training, and evaluation. Each stage is designed to optimize the performance of the models while ensuring generalization to unseen data.

3.1 Experimental Setup

- **Train-Test Split**: The dataset was split into training and testing sets with an 80:20 ratio, stratified by label to maintain class balance.

- **Hyperparameters**:
  - Maximum vocabulary size (MAX_WORDS): 20,000
  - Sequence length (MAX_LEN): 300
  - Embedding dimension: 100 (GloVe)
  - Batch size: 128
  - Dropout rate: 0.5
  - Number of epochs: 10 with early stopping based on validation loss
- **Class Weights**: Applied in ML models and sample weighting in Naive Bayes to mitigate minor class imbalance.
- **Optimizer and Loss**: For deep learning models, Adam optimizer with binary cross-entropy loss was used.
- **Evaluation Metrics**: Models were evaluated using standard classification metrics:
  - **Accuracy:** Overall fraction of correctly classified samples.
  - **Precision:** Fraction of correctly predicted positive instances among all predicted positives.
  - **Recall:** Fraction of correctly predicted positive instances among all actual positives.
  - **F1-score:** Harmonic mean of precision and recall, providing a balanced metric.
  - **Confusion Matrix:** Visual representation of true vs. predicted labels to analyze misclassifications.

3.2 Feature Extraction and Representation

To convert textual data into numerical features suitable for ML and DL models, the following techniques were used:

- **TF-IDF Vectorization**: For classical machine learning models, the Term Frequency–Inverse Document Frequency (TF-IDF) representation was computed with a maximum of 50,000 features. This approach captures the importance of each word in the corpus relative to its frequency across all documents, helping the models identify discriminative terms for fake versus real news.
- **Tokenization and Padding**: For deep learning models, a tokenizer was applied to the full_text column, converting each article into a sequence of integer tokens. All sequences were padded to a fixed length of 300 tokens to ensure uniform input dimensions.

- **Pre-trained Word Embeddings (GloVe)**: Deep learning models utilized 100-dimensional GloVe embeddings to initialize the embedding layer. These embeddings capture semantic relationships between words learned from large corpora, enhancing the ability of the models to understand contextual meaning.

3.3 Machine Learning Models

Three traditional machine learning algorithms were implemented using TF–IDF vectorized features. Class weights were used to account for minor class imbalance.

Logistic Regression

- A linear model suitable for binary classification.
- Maximum iterations were set to 1,000 to ensure convergence.

Naïve Bayes

- Multinomial Naïve Bayes was employed, which is suitable for count-based features such as TF–IDF vectors. Sample weighting was used.

Random Forest

- Random Forest is an ensemble learning algorithm that constructs multiple decision trees and averages their predictions.
- 200 estimators were used, with class weights to balance the dataset.

3.4 Deep Learning Models

To capture semantic and sequential patterns in the text, deep learning models with embeddings were used.

LSTM (Long Short-Term Memory)

- LSTM networks are designed to handle long-term dependencies in sequences.
- Architecture:
    - Embedding layer with GloVe weights
    - LSTM layer (128 units, dropout 0.3, recurrent dropout 0.2)
    - Dropout layer (0.5)
    - Dense output layer with sigmoid activation

BiLSTM (Bidirectional LSTM)

- BiLSTM extends LSTM by processing sequences in both forward and backward directions, allowing the model to capture past and future context simultaneously.

- Architecture is similar to LSTM but with a Bidirectional wrapper around the LSTM layer.

GRU (Gated Recurrent Unit)

- GRU is a variant of LSTM with fewer parameters offering faster training while maintaining the ability to capture sequential dependencies.
- Architecture mirrors the LSTM network but replaces the LSTM layer with a GRU layer.

3.5 Justification of Approach

- Classical ML models were implemented to establish baseline performance due to their interpretability and efficiency on TF-IDF features.
- Deep learning models were chosen to leverage sequential dependencies in textual data, capturing semantic context that ML models may miss.
- GloVe embeddings were incorporated to improve semantic understanding without requiring training embeddings from scratch.
- Using multiple models allows performance comparison, demonstrating that sequence-based models (GRU, BiLSTM) outperform traditional ML approaches for this task.

4. Results and Analysis

The performance of both classical machine learning (ML) models and deep learning (DL) models was evaluated on the preprocessed Fake and Real News dataset. The primary objective was to compare model performance in terms of accuracy, precision, recall, and F1-score for detecting fake versus real news.

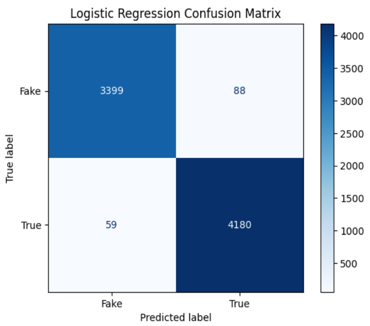
4.1 Machine Learning Model Performance

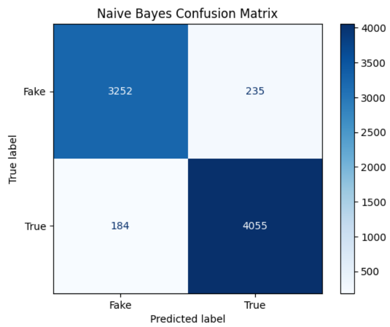
Three classical ML models were evaluated using TF-IDF features: Logistic Regression, Naive Bayes, and Random Forest. The models were trained on the training set (80%) and evaluated on the test set (20%).

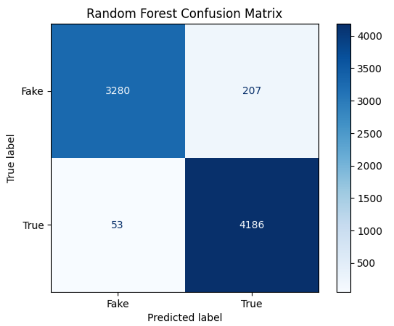
Model	Accuracy	Precision (Fake/Real)	Recall (Fake/Real)	F1-Score (Fake/Real)
Logistic Regression	0.9810	0.98 / 0.98	0.97 / 0.99	0.98 / 0.98
Naive Bayes	0.9458	0.95 / 0.95	0.93 / 0.96	0.94 / 0.95
Random Forest	0.9663	0.98 / 0.95	0.94 / 0.99	0.96 / 0.97

- Logistic Regression achieved the highest accuracy among ML models (98.1%) and performed consistently across both classes.
- Naive Bayes while efficient showed slightly lower recall for fake news (93%), suggesting some fake news articles were misclassified.
- Random Forest performed well, especially in capturing true news articles but slightly underperformed in detecting fake news.

Confusion Matrices for each model visually confirmed these observations, showing minimal misclassifications, especially for Logistic Regression.







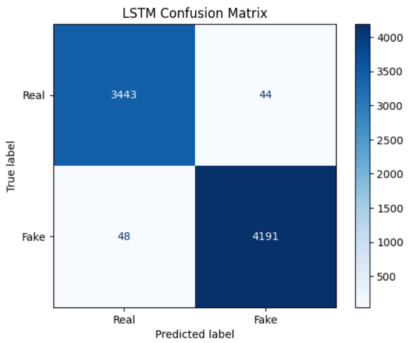
4.2 Deep Learning Model Performance

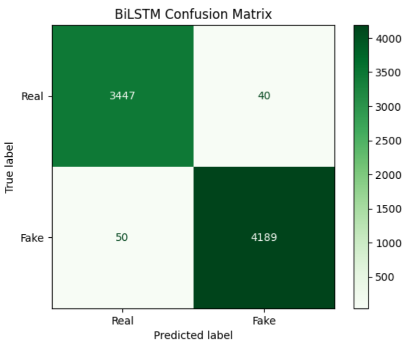
Three deep learning architectures-LSTM, BiLSTM, and GRU were trained using pre-trained GloVe embeddings, padded sequences, and early stopping to prevent overfitting.

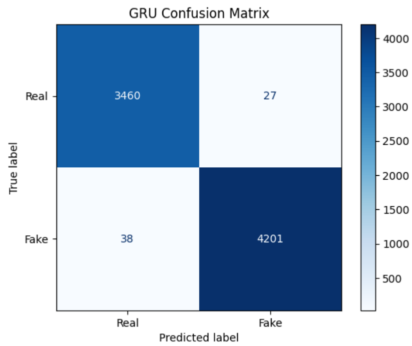
Model	Test Accuracy	Precision (Fake/Real)	Recall (Fake/Real)	F1-Score (Fake/Real)
LSTM	0.9881	0.99 / 0.99	0.99 / 0.99	0.99 / 0.99
BiLSTM	0.9884	0.99 / 0.99	0.99 / 0.99	0.99 / 0.99
GRU	0.9916	0.99 / 0.99	0.99 / 0.99	0.99 / 0.99

- GRU achieved the highest accuracy (99.16%), demonstrating its efficiency in capturing sequential dependencies with fewer parameters compared to LSTM.
- BiLSTM performed similarly to LSTM but did not significantly improve performance suggesting that unidirectional context provided by LSTM and GRU was sufficient for this dataset.

Confusion Matrices showed almost perfect classification, with negligible misclassifications in both fake and real news categories.

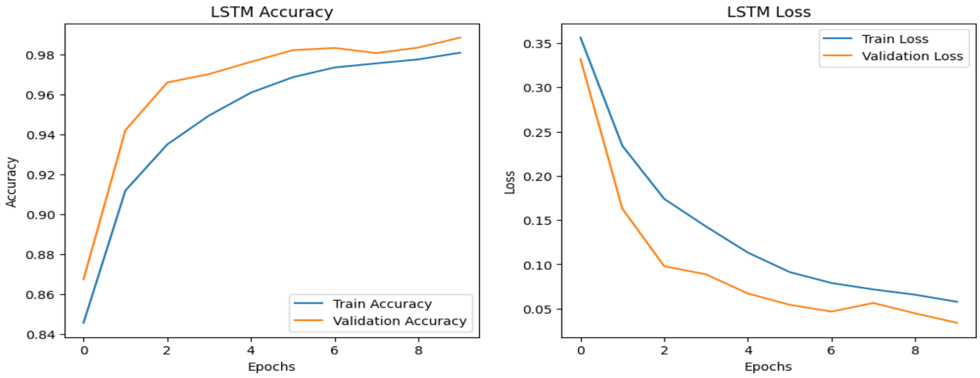




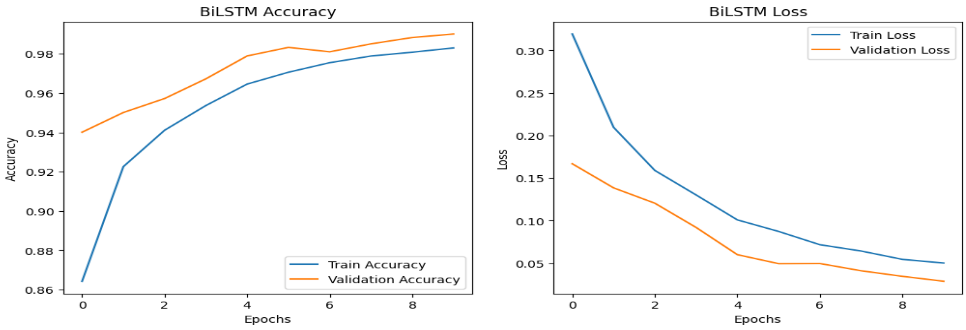


4.2.1 Training and Validation Curves

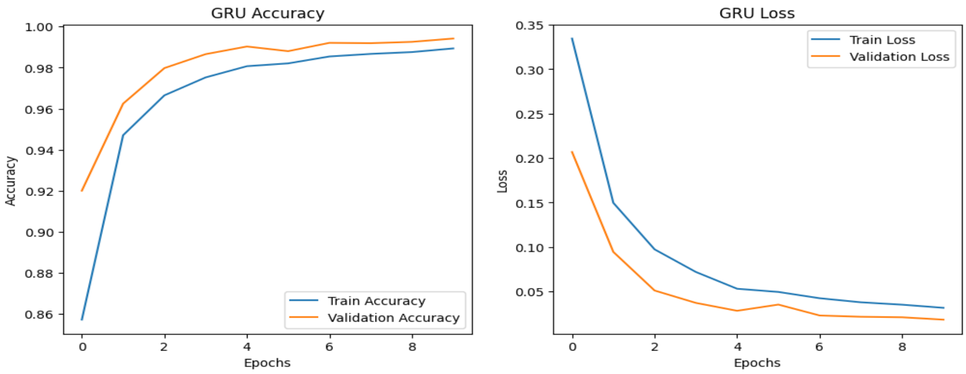
- Accuracy and loss curves for each deep learning model indicated smooth convergence with minimal overfitting.



- Early stopping based on validation loss prevented excessive training particularly in BiLSTM and GRU models.



- GRU achieved the fastest convergence among DL models while maintaining superior performance.



4.3 Comparative Analysis

Classical ML vs Deep Learning: ML models (Logistic Regression, Naive Bayes, Random Forest) achieved high accuracy (94–98%), demonstrating that TF-IDF features capture discriminative information effectively. DL models further improved performance (98–99%) leveraging sequential patterns and contextual embeddings.

Model Efficiency: Naive Bayes is the fastest to train but slightly less accurate. Logistic Regression and Random Forest provide strong baselines. LSTM, BiLSTM and GRU require more computational resources but offer superior predictive power.

4.4 Unexpected Observations:

Despite using a relatively simple architecture, the GRU-based model achieved accuracy levels significantly higher than anticipated for a real-world text classification task. This was unexpected because deep learning models trained on news datasets typically require larger corpora, extensive tuning, or pretrained embeddings to reach comparable performance. The result suggests that the dataset contained strong linguistic or stylistic patterns that the model could easily learn, allowing it to generalize well even with modest model complexity.

5. Web System

The best performing deep learning model GRU was deployed as a fully interactive web-based Fake News Detection System using Streamlit Cloud [2] . The purpose of this system is to allow users to test the model in real time by entering any news headline or article and receiving an instant prediction and enabling public access without requiring local installation.

Web Application Link: https://fake-news-system-app.streamlit.app/

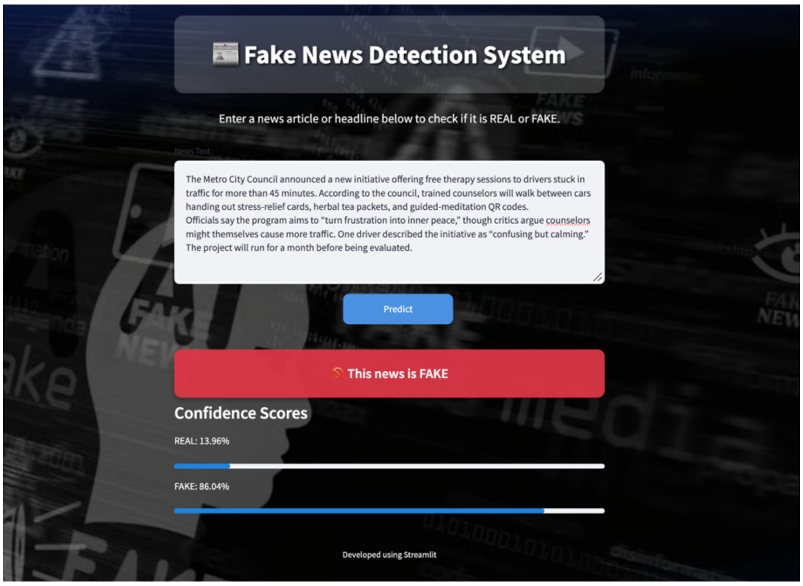
5.1 System Workflow

- **User Input:** The user enters a news headline or full article into the text box.
- **Model Inference:** The backend converts the text into padded sequences and uses the GRU model to compute the probability of the news being real. All computations are performed instantly on the server-side.
- **Output Display:** The system returns a binary prediction (REAL or FAKE) along with confidence scores for both classes

5.2 Sample Snapshots

Real News Prediction: Shows the system identifying an authentic news input and returning a high real-news confidence score.

Fake News Prediction: Shows the system accurately detecting misleading content assigning higher fake-news confidence score.



6. Conclusion and Future Work

This project successfully demonstrated the development of an effective fake news detection system using deep learning techniques. By preprocessing a balanced dataset of real and fake news articles, training multiple models, and evaluating their performance, the study showed that sequence-based deep learning architectures-particularly the GRU model achieved exceptionally high accuracy and strong generalization on unseen data. The model was able to capture linguistic patterns, writing styles and contextual cues that consistently distinguish real information from fabricated content. Furthermore, deploying the model as an interactive web application validated its practical usefulness by enabling users to test news content in real time.

The impact of this work lies in its ability to provide an accessible, automated tool that supports users in identifying potentially misleading information. As misinformation continues to circulate rapidly across digital platforms, tools like this can offer timely assistance in reducing the spread of false content. Although the system is not meant to replace professional fact-checking, it can support users by offering quick, data-driven assessments based on learned patterns in news language.

6.1 Limitations

Despite its strong performance, the project has several limitations. First, the dataset although balanced, is limited to specific sources meaning the model may struggle with news formats or writing styles not represented in the training data. Second, the model analyzes only the textual content and does not consider external factors such as publication credibility, metadata, or multimedia context. Additionally, high accuracy may partly reflect clear linguistic differences in the dataset between real and fake news rather than universally generalizable patterns. The model may also misclassify legitimate articles that use sensational or emotional language, as observed in several test cases.

6.2 Future Enhancements

Future work can address these limitations in several ways. One direction is to expand the dataset using more diverse sources, languages, and writing styles to improve generalization. Incorporating transformer-based models such as BERT or RoBERTa could further enhance contextual understanding and robustness. Another promising extension is to develop a multimodal detection system that analyzes images, headlines, article sources and user comments alongside text. Finally, improving explainability-such as highlighting which words influenced the prediction-could make the system more transparent and trustworthy for end-users.

Overall, the project demonstrates that deep learning models can play a meaningful role in strengthening information reliability and combating misinformation online while also showing clear opportunities for continued enhancement and broader real-world applicability.

References

[1] C. Bisaillon, "Fake and Real News Dataset," Kaggle, 2019. [Online]. Available: https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

[2] Streamlit, "Streamlit Documentation," 2024. [Online]. Available: https://docs.streamlit.io

Appendices

Appendix A – Detailed Model Performance

Table A1: Performance Metrics of All Models

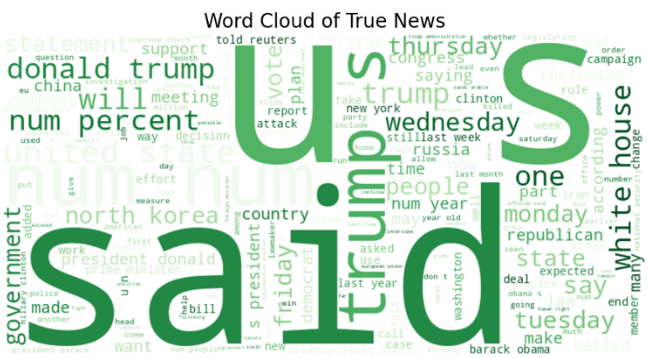
Model	Accuracy	Precision (Fake)	Recall (Fake)	F1-score (Fake)	Precision (Real)	Recall (Real)	F1-score (Real)
Logistic Regression	0.9810	0.98	0.97	0.98	0.98	0.99	0.98
Naive Bayes	0.9458	0.95	0.93	0.94	0.95	0.96	0.95
Random Forest	0.9663	0.98	0.94	0.96	0.95	0.99	0.97
LSTM	0.9881	0.99	0.99	0.99	0.99	0.99	0.99
BiLSTM	0.9884	0.99	0.99	0.99	0.99	0.99	0.99
GRU	0.9916	0.99	0.99	0.99	0.99	0.99	0.99

Note: Metrics are evaluated on the test dataset.

Appendix B – Sample Word Clouds

Figure B1: Word Cloud of Fake News

Figure B2: Word Cloud of Real News



Note: Word clouds visualize the most frequent terms in fake vs real news articles.