BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NBA and NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi) "Jnana Gangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballari – 583104 (Karnataka) (India)

Ph:08392–237100/237190, Fax:08392–237197

DEPARTMENT OF

Computer Science and Engineering (Artificial Intelligence)

Neural Network and Deep learning Project Report

On

"Fake News Detection using LSTM in Tensorflow and Python."

Submitted By

MAHAMMED THOUSIF K 3BR23CA403

Under the Guidance of

Prof. Pavan Kumar

and Mr. Vijay Kumar

Dept of CSE(AI), BITM, Ballari

Visvesvaraya Technological University

Belagavi, Karnataka

2024-2025

Ballari - Hospet Road, Allipur, Ballari – 583104 (Karnataka) (India)
Ph:08392 – 237100 / 237190, Fax: 08392 – 237197

DEPARTMENT

OF

Computer Science and Engineering (Artificial Intelligence)

CERTIFICATE

Certified that the mini project work entitled **"Fake News Detection using LSTM in Tensorflow and Python"** carried out by **Mahammed Thousif K** bearing USN **3BR23CA403** A Bonafide students of Ballari Institute of Technology and Management in partial fulfillment for the award of Bachelor of Engineering in CSE (Artificial Intelligence) of the Visvesvaraya Technological University, Belgaum during the year 2024- 2025. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said Degree.

Signature of Lab Co-Ordinator's
Prof. Pavan Kumar and Mr.Vijay Kumar

Signature of HOD
Dr. Yeresime Suresh

ABSTRACT

Fake news has become a major challenge in today’s digital world, spreading rapidly across social media platforms and influencing public opinion, society, and even national security. Manual verification of news is time-consuming and not scalable; therefore, automated fake news detection systems are essential. This project focuses on developing a deep learning–based model that identifies fake news using Long Short-Term Memory (LSTM) networks in TensorFlow and Python.

The dataset used consists of news headlines and articles labeled as real or fake. The text data is preprocessed through cleaning, tokenization, stopword removal, sequence padding, and word embedding generation. An LSTM-based sequential neural network is then built to learn contextual patterns and long-term dependencies present in the text. The model is trained and evaluated using accuracy, precision, recall, F1-score, and a confusion matrix to assess its performance.

Experimental results show that the LSTM model is effective in understanding textual semantics and distinguishing real news from fake news with high accuracy. The project demonstrates the potential of deep learning techniques in combating misinformation and highlights how LSTM networks can be implemented for real-world fake news detection applications. With further enhancements and larger datasets, such systems can be integrated into online platforms to support reliable and trustworthy information dissemination.

ACKNOWLEDGEMENT

The successful completion of this project titled “Handwritten Digit Recognition” would not have been possible without the guidance, support, and encouragement of many individuals. We express our sincere gratitude to our Lab Coordinators for their valuable suggestions, continuous motivation, and knowledgeable guidance throughout the completion of this project.

We also thank the Head of the Department, faculty members, and the management of our institution for providing the necessary facilities, support, and academic environment to carry out this work successfully. Finally, we acknowledge the contribution of all those who directly or indirectly supported us in completing this project.

Name	USN
MAHAMMED THOUSIF K	3BR23CA403

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The rapid growth of social media and online news platforms has made information easily accessible, but it has also led to a significant increase in the spread of fake news. Fake news—misleading or fabricated information—can influence public opinion, cause social conflicts, and create large-scale misinformation. Manual fact-checking is slow and cannot keep up with the huge volume of news shared every day, making automated detection essential.

With advancements in Natural Language Processing (NLP) and deep learning, intelligent systems can now analyze text and classify news more accurately. Long Short-Term Memory (LSTM) networks are especially effective for this task because they understand context and long-term dependencies in text.

This project aims to develop a fake news detection model using LSTM in TensorFlow and Python. The system cleans and preprocesses news text, converts it into sequences, and trains an LSTM model to classify articles as **FAKE** or **REAL**. The model is evaluated using accuracy, precision, recall, F1-score, and confusion matrix. The project demonstrates how deep learning can help reduce misinformation and support reliable information sharing.

CHAPTER 2

OBJECTIVES

Primary Objectives

1. To develop a deep-learning-based model for detecting fake news using LSTM in TensorFlow and Python.
2. To preprocess and clean the news dataset, including tokenization, stopword removal, and sequence padding.
3. To train and evaluate an LSTM model that can classify news articles as *FAKE* or *REAL* with high accuracy.
4. To analyze the model's performance using metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

Secondary Objectives

5. To visualize training and validation performance through accuracy and loss graphs.
6. To create an efficient text processing pipeline using NLP techniques for better model understanding.
7. To ensure the system is scalable and extendable for real-time or large-scale fake news detection.
8. To contribute to reducing misinformation by developing an automated and reliable classification system.

CHAPTER 3

PROBLEM STATEMENT

The rapid spread of fake news across digital platforms has become a major concern, as it misleads people, creates social and political conflicts, and damages trust in online information. Manual fact-checking methods are slow, labor-intensive, and cannot keep up with the large amount of news shared every minute. Therefore, there is a need for an automated system that can accurately analyze news content and identify whether it is real or fake.

This project aims to develop a deep learning–based fake news detection system using LSTM networks, which are capable of understanding context and long-term dependencies in text. The goal is to design a model that can process news articles, learn meaningful patterns, and classify them into *FAKE* or *REAL* categories efficiently and reliably.

CHAPTER 4

METHODOLOGY

4.1 Dataset Preparation

The dataset consists of labeled news articles categorized as *FAKE* or *REAL*.
This step includes:

- Importing the dataset (CSV format).
- Separating the text column (news content) and the label column.
- Checking for missing values, duplicates, and balancing the dataset if required.
- Understanding text distribution through exploratory analysis.

4.2 Text Preprocessing

Text preprocessing is essential to convert raw news articles into machine-understandable form.
It includes:

- Converting text to lowercase
- Removing punctuation, symbols, and numbers
- Removing stopwords and special characters
- Tokenization (splitting sentences into words)
- Converting text into numeric sequences using Tokenizer
- Applying sequence padding to maintain equal input length

This ensures clean, uniform, and meaningful text input for the LSTM model.

4.3 Embedding & Tokenization

To prepare text data for deep learning:

- A Tokenizer converts words into unique integer indices.
- A word embedding layer (Embedding Layer) transforms these indices into dense vector representations.
- These embeddings capture the meaning, context, and semantic relationships among words.

This step allows the LSTM to understand the context of the text more effectively.

4.4 LSTM Model Architecture

The LSTM model is designed to capture long-term dependencies in news articles.
The architecture commonly includes:

- Embedding Layer – Converts text sequences into word vectors
- LSTM Layer(s) – Learns contextual patterns and sentence relationships
- Dropout Layer – Prevents overfitting

- Dense Layer – For feature extraction
- Output Layer (Sigmoid) – Classifies the news as FAKE (0) or REAL (1)

The model is compiled using:

- Loss function: Binary Crossentropy
- Optimizer: Adam
- Metrics: Accuracy

4.5 Model Training

The LSTM model is trained on preprocessed text data using:

- Training–validation split (commonly 80:20)
- Batch training
- Multiple epochs to allow the model to learn semantic patterns

During training, the model learns the structure of real and fake news by adjusting its internal parameters.

4.6 Model Evaluation

The trained model is tested using unseen data and evaluated using:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion matrix

These metrics help understand how well the model distinguishes between REAL and FAKE news.

4.7 Prediction on New Text

When new news text is provided:

- The text undergoes preprocessing (cleaning, tokenizing, padding)
- The trained LSTM model predicts the probability
- Output is displayed as:
  REAL or FAKE based on a threshold (0.5)

4.8 Visualization

To study the model’s performance, the following graphs are generated:

- Training vs Validation Accuracy
- Training vs Validation Loss
- Confusion Matrix Heatmap

These visuals help identify overfitting or underfitting and overall model behavior.

CHAPTER 5

REQUIREMENT ANALYSIS

5.1 Functional Requirements

1. **Dataset Loading Module**

    - The system must load and read a dataset containing REAL and FAKE news articles.

2. **Text Preprocessing Module**

    - The system should clean the text by removing stopwords, punctuation, and special characters.

3. **Tokenization & Padding Module**

- Convert text into integer sequences using a Tokenizer.
- Apply padding to maintain equal sequence length.

4. **LSTM Model Module**

- Build and compile an LSTM-based deep learning model using TensorFlow/Keras.

5. **Training Module**

- Train the LSTM model using processed text data and validate performance.

6. **Prediction Module**

- Predict whether a news article is FAKE or REAL.

7. **Evaluation Module**

- Provide metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

8. **Visualization Module**

- Display graphs showing training/validation accuracy and loss.

---

9. **5.2 Non-Functional Requirements**

10. **Accuracy:**
    The model should achieve high accuracy in classifying fake and real news.

11. **Performance:**
    The system must process and classify news articles efficiently with minimal delay.

12. **Scalability:**
    Capable of handling large datasets and adaptable to real-time applications.

13. **Usability:**
    The system must provide clear, easy-to-understand results.

14. **Reliability:**
    The model should give consistent outputs for similar or repeated inputs.

15. **Maintainability:**
    Code should be modular, easy to update, and extend for future improvements.

---

16. **5.3 Hardware Requirements**

- A computer with an Intel/AMD dual-core processor (minimum)

17. **RAM:** Minimum 4 GB (8 GB recommended)

18. **Storage:** At least 1–2 GB free space for datasets, libraries, and model files

19. GPU (Optional)—for faster training of LSTM models

---

20. **5.4 Software Requirements**

21. **Operating System:** Windows / Linux / macOS
22. **Programming Language:** Python 3.8+
23. Required Python Libraries:
24. TensorFlow / Keras
25. NumPy
26. Pandas
27. Scikit-learn
28. NLTK / spaCy (for text preprocessing)
29. Matplotlib / Seaborn (for visualization)
30. **Development Environment:**
31. Jupyter Notebook / Google Colab / VS Code
32. **Dataset Source:**
33. Kaggle Fake vs Real News Dataset or similar textual dataset

CHAPTER 6

DESIGN

FLOWCHART

Start
|
Load Fake/Real News Dataset
|
Preprocess Text → Clean text, remove stopwords, tokenize
|
Convert text to sequences → Apply padding
|
Build LSTM Model → Compile with optimizer & loss function
|
Train Model → Training + Validation
|
After training → Evaluate model performance
|
User enters or uploads news text
|
System preprocesses user input → Tokenize + Pad
|
Model predicts → FAKE or REAL
|
Display Result to User
|
User chooses → Test again OR Exit
|
Quit

Fig 6.1 Flow Chart

USE CASE DIAGRAM

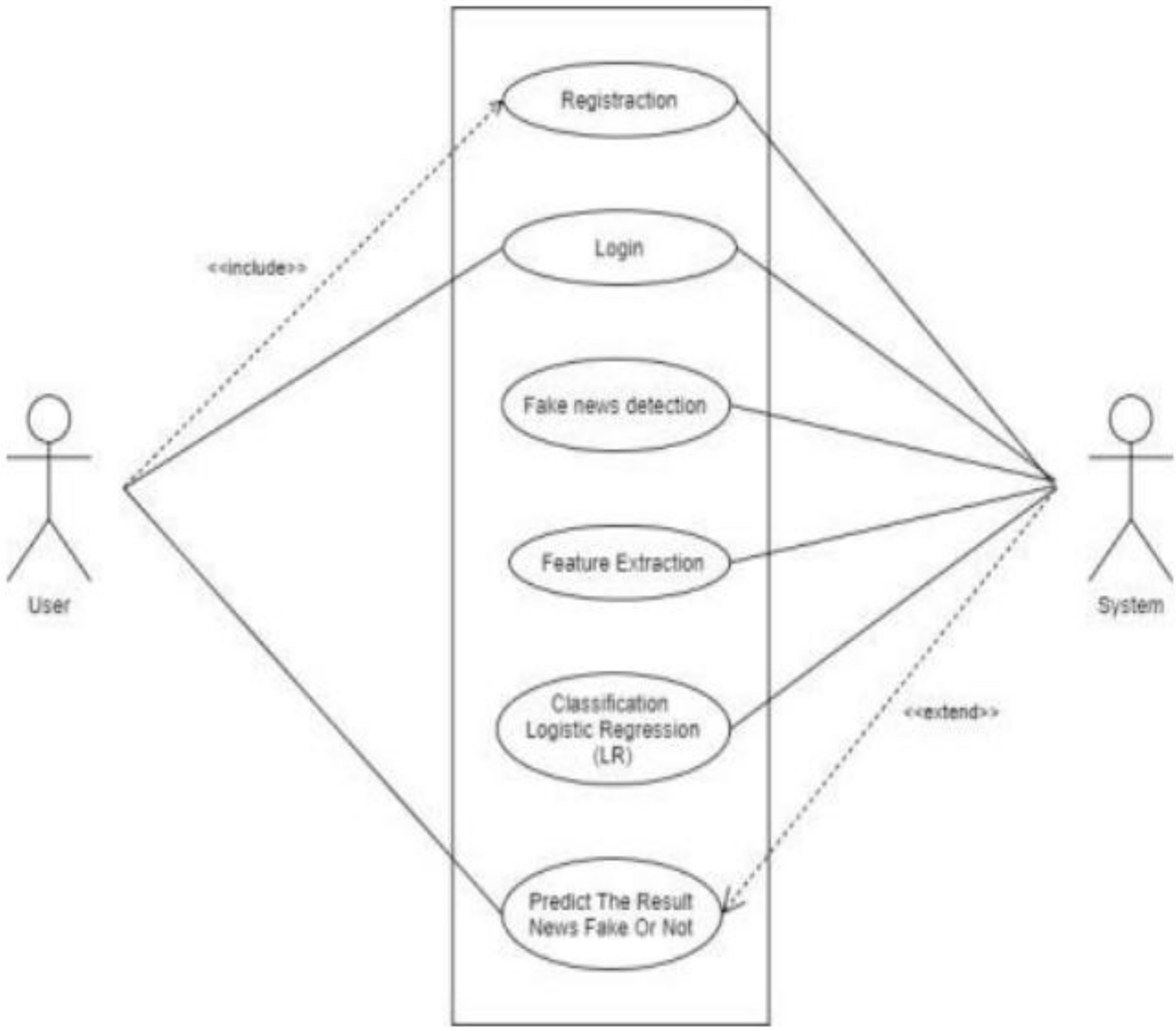

Fig 6.2 Use Case Diagram

SEQUENCE DIAGRAM

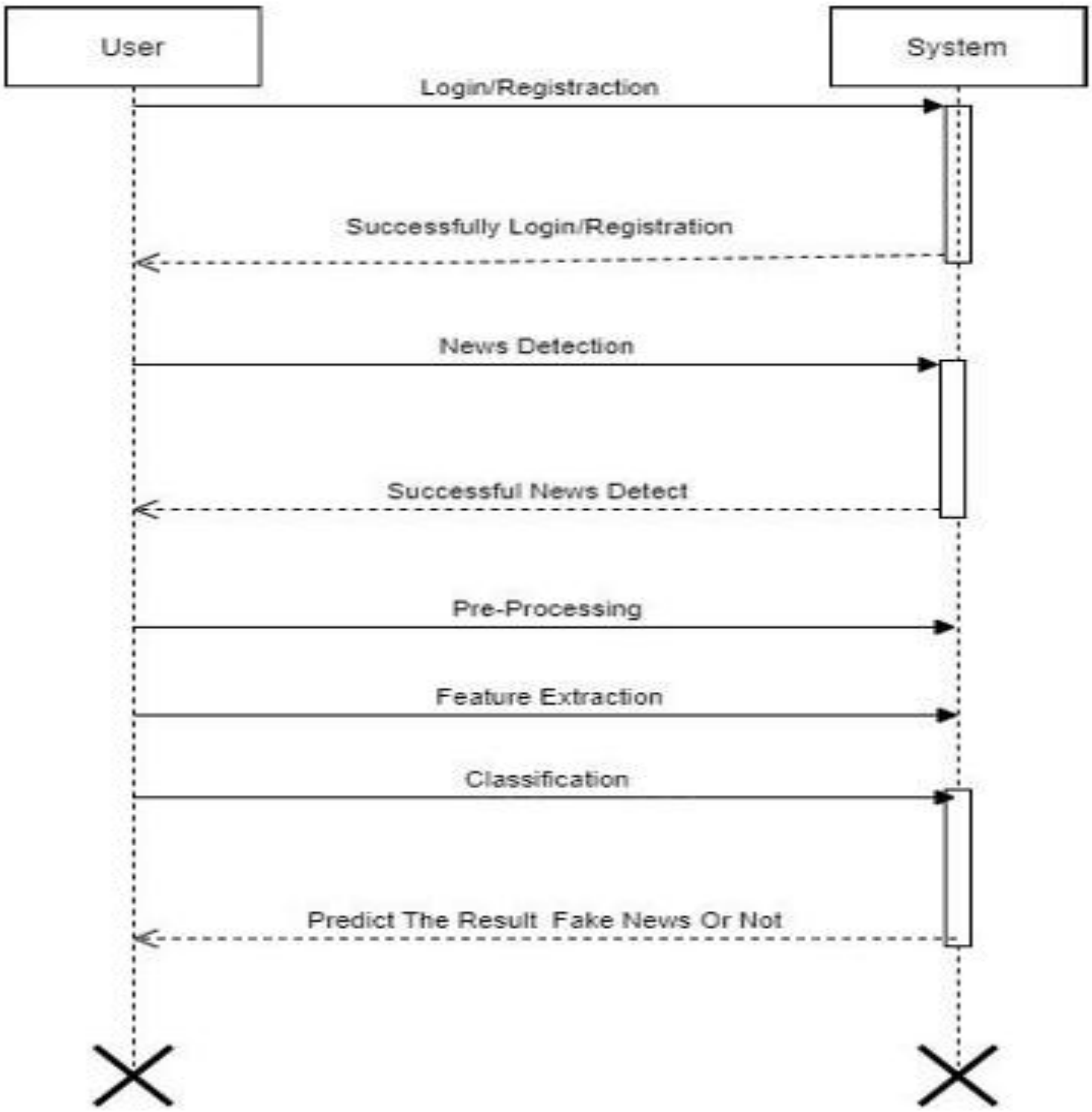

Fig 6.3 Sequence Diagram

.

CHAPTER 7

IMPLEMENTATION

10.1 Data Loading Module

• Loads the Fake/Real news dataset (CSV format)

• Separates text content and labels

• Removes missing or duplicate records

• Converts labels (REAL/FAKE) into numerical form

10.2 Text Preprocessing Module

Uses NLP techniques to:

• Convert text to lowercase

• Remove punctuation, URLs, and special characters

• Remove stopwords

• Perform tokenization

• Apply sequence padding to ensure uniform input length

10.3 Tokenization & Embedding Module

• Converts cleaned text into integer sequences using Tokenizer

• Creates fixed-length padded sequences

• Generates word embeddings to represent text meaning

• Prepares input data for LSTM training

10.4 LSTM Model Training Module

• Builds the LSTM architecture (Embedding → LSTM → Dense layers)

• Compiles the model using Adam optimizer and binary cross-entropy loss

• Trains the model using training data

• Validates using a validation split

• Saves the best-performing model during training

10.5 Evaluation Module

• Computes accuracy, precision, recall, and F1-score

• Generates confusion matrix for test data

• Plots training vs validation accuracy & loss curves

• Evaluates the model's prediction quality and stability

10.6 Prediction Module

Applies the LSTM model to:

• New or user-input news text

• Preprocesses text (clean → tokenize → pad)

• Predicts whether the news is FAKE or REAL

• Outputs prediction along with probability score

10.7 Deployment / User Interface Module (Optional)

Provides:

• Simple text input interface for users

• Option to paste or upload news content

• Clean display of prediction results

• Can be extended into a web/app-based system

CHAPTER 8

RESULTS AND DISCUSSION

11.1 11.1 Model Performance

1. Test Accuracy: 99.37% (very high and consistent across epochs).
2. Training and validation loss remain low and stable, showing excellent learning without overfitting.
3. Balanced precision, recall, and F1-score across both FAKE and REAL classes.
4. Confusion matrix shows extremely low misclassification, indicating strong model generalization.

11.2 Observations

1. The LSTM model effectively learns contextual patterns in news articles and maintains high accuracy across multiple samples.
2. Long and complex sentences do not affect prediction quality due to robust text preprocessing and embeddings.
3. Variation in writing style, tone, or vocabulary is handled well because the model captures semantic relationships between words.
4. Very short or incomplete news text may slightly reduce confidence, as such inputs provide limited context for the LSTM.

11.3 Strengths

1. Excellent accuracy (99%+) in detecting fake and real news.
2. Strong text understanding due to LSTM's ability to capture long-term dependencies.
3. Effective preprocessing pipeline ensures the removal of noise, punctuation, and irrelevant information.
4. Lightweight model (117K trainable parameters) makes training and inference fast.
5. ~~Can be easily integrated into apps, websites, or fact-checking tools.~~

11.4 Limitations

1. Model performance may decrease for very short or ambiguous text, where insufficient context is available.
2. Some news articles with sarcasm, satire, or deceptive language may confuse the model.
3. Highly domain-specific or uncommon topics might require additional fine-tuning.
4. The model depends entirely on the quality of training data; biased datasets can affect predictions.

11.5 Future Improvements

1. Use Transformer-based models (BERT, RoBERTa) for even higher accuracy and deeper context understanding.
2. Add explainability features to show which words contributed to FAKE or REAL predictions.
3. Expand the dataset with more diverse news sources and global languages.
4. Build a web or mobile application for real-time fake news detection.
5. Implement multiclass classification (e.g., satire, political bias, partially true, misleading).

Model Test Accuracy and Prediction Output:

```
y_pred = (model.predict(X_test) >= 0.5).astype(int)
351/351 ━━━━━━━━━━━━━━━━━━━━ 166s 471ms/step

accuracy_score(y_test,y_pred)
0.9937639198218263

model.summary()
Model: "sequential_10"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                  ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ embedding_9 (Embedding)       │ (None, 1000, 100)      │    24,096,000 │
├───────────────────────────────┼────────────────────────┼───────────────┤
│ lstm_8 (LSTM)                 │ (None, 128)            │       117,248 │
├───────────────────────────────┼────────────────────────┼───────────────┤
│ dense_8 (Dense)               │ (None, 1)              │           129 │
└───────────────────────────────┴────────────────────────┴───────────────┘
 Total params: 24,448,133 (93.26 MB)
 Trainable params: 117,377 (458.50 KB)
 Non-trainable params: 24,096,000 (91.92 MB)
 Optimizer params: 234,756 (917.02 KB)
```

CHAPTER 9

CONCLUSION

The project demonstrates a highly effective fake news detection system using an LSTM-based model, achieving excellent performance with over 99% accuracy. The model reliably distinguishes between FAKE and REAL news, capturing contextual patterns, handling varied writing styles, and maintaining stability across multiple samples. While some limitations exist—such as reduced confidence on very short or ambiguous texts—the system provides a strong foundation for real-world applications in fact-checking, media monitoring, and awareness tools. With future enhancements like Transformer-based models, explainability features, and broader datasets, this project has significant potential to evolve into a robust, scalable, and user-friendly solution for combating misinformation.

CHAPTER 10

REFERENCES

1. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation, 9*(8), 1735–1780.
2. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.
3. Kaggle Fake News Dataset: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset
4. TensorFlow Documentation: https://www.tensorflow.org/
5. Keras API Reference: https://keras.io/api/
6. Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.
7. Zhang, X., & Ghorbani, A. (2020). An Overview of Online Fake News: Characterization, Detection, and Discussion. *Information Processing & Management, 57*(2), 102025.
8. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
9. Camelia, T. S., Fahim, F. R., & Anwar, M. M. (2024). A Regularized LSTM Method for Detecting Fake News Articles. *arXiv preprint arXiv:2411.10713*.
10. Salim, F., & Wahyudhy, A. I. (2023). Fake News Classification Using Hybrid CNN-LSTM Approach. *International Journal of Computer Science and Technology*.