Fake News Verifier

Submitted in partial fulfillment of the requirements of the degree

BACHELOR OF ENGINEERING IN **COMPUTER ENGINEERING**

By

1. E. LOGITHA	**(122A1031)**
2. G.U. GOPIKHA	**(122A1032)**
3. K NISHARANI	**(122A1042)**
4. MAHALAKSHMI	**(122A1053)**

Supervisor

PROF.NAMRATA PATEL

Department of Computer Engineering

SIES Graduate School of Technology

Nerul, Navi Mumbai - 400 706

University of Mumbai

(AY 2024-25)

CERTIFICATE

This is to certify that the Mini Project entitled **"Fake News Verifier"** is a bonafide work of **E. Logitha (122A1031), G.U. Gopikha (122A1032),K Nisharani (122A1042), Mahalakshmi Gounder (122A1053)** submitted to the University Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Computer Engineering" .**

Prof. Namrata Patel
Supervisor

Dr. Aparna Bannore
Head of Department

Dr. K. Lakshmi Sudha
Principal

Mini Project Approval

This Mini Project entitled **" Fake News Verifier "** by **E. Logitha (122A1031), G.U. Gopikha (122A1032), K Nisharani (122A1042),Mahalakshmi Gounder (122A1053)** is approved for the degree of Bachelor of **Engineering in Computer Engineering.**

Examiners

1……………………………………..
(Internal Examiner Name & Sign)

2………………………………………
(External Examiner name & Sign)

Date:

Place:

Contents

ACKNOWLEDGEMENT

We would like to express our thanks to the people who have helped us throughout our project. We are grateful to our guide Prof. Namrata Patel and coordinator, Prof. Sunil Punjabi for nonstop support for the project.

A special thanks goes to each other who worked together as a team in completing the project, where we all exchanged our own interesting ideas, thoughts and made it possible to complete our project with all accurate information. We also wish to thank our parents for their personal support and attention who inspired me to go my own way.

We would also like to extend our sincere gratitude to our Principal , Dr. K. Lakshmi Sudha and our Head of the Department, Dr. Aparna Bannore for their continuous support and encouragement.

We also would like to thank our other faculty members for providing us with all the required resources and references for the project.

ABSTRACT

The Fake News Verifier system is a multilingual tool designed to detect misinformation using machine learning. It supports English, Hindi, and Tamil, with data collected via web scraping (BeautifulSoup). The system uses preprocessing techniques including data cleaning, tokenization, stop word removal, and TF-IDF for feature extraction. Models like Logistic Regression, SVM, Random Forest, Gradient Boosting, and ensemble methods are employed for high accuracy. Built with frameworks like Pandas, Flask, and Indic NLP, this system ensures reliable detection across regional languages, enhancing digital media credibility.

The development process begins with extensive data collection from reliable news sources and websites in multiple languages using web scraping tools like BeautifulSoup. Once collected, the data undergoes thorough preprocessing including cleaning, tokenization, stop word removal, and feature extraction using TF-IDF—to ensure it is suitable for training machine learning models. The Indic NLP library is used to handle regional language processing, particularly for Tamil and Hindi, enhancing the system's multilingual capabilities.

The system is powered by various machine learning algorithms, including Logistic Regression, SVM, Random Forest, and Gradient Boosting, each contributing to robust prediction accuracy. An ensemble approach is also implemented to combine the strengths of individual models for improved performance. The web interface, built using Flask and Pandas, allows users to input news articles and receive instant verification results, making the tool both accessible and efficient for users across education, journalism, and public service sectors.

LIST OF ABBREVIATIONS

General Abbreviations

- **CSV** – Comma-Separated Values
- **UI** – User Interface
- **UX** – User Experience

Machine Learning & Data Processing Abbreviations

- **ML** – Machine Learning
- **NLP** – Natural Language Processing
- **TF-IDF** – Term Frequency-Inverse Document Frequency
- **SVM** – Support Vector Machine
- **LR** – Logistic Regression
- **RF** – Random Forest
- **GB** – Gradient Boosting
- **EDA** – Exploratory Data Analysis

Programming & Frameworks Abbreviations

- **Flask** – A Python web framework
- **Pandas** – Python Data Analysis Library
- **BS4** – BeautifulSoup4 (used for web scraping)
- **Sklearn** – Scikit-learn (Python ML library)

Multilingual & Language Processing Abbreviations

- **Indic NLP** – Indian Language Natural Language Processing
- **ENG** – English
- **HIN** – Hindi
- **TAM** – Tamil

Performance and Testing Abbreviations

- **Acc.** – Accuracy
- **TP** – True Positive
- **FP** – False Positive
- **FN** – False Negative
- **TN** – True Negative
- **ROC** – Receiver Operating Characteristic

–

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In the era of rapid digital communication, the spread of misinformation has emerged as a serious global issue, with fake news becoming one of the most dangerous forms of online content. Social media platforms, messaging apps, and online news portals allow news to travel faster than ever before, but they also serve as breeding grounds for the unchecked circulation of false or misleading information. The consequences of fake news can be severe — from creating public panic and influencing elections to damaging reputations and undermining trust in media institutions.

This problem is especially pronounced in a linguistically diverse country like India, where misinformation in regional languages often goes unnoticed by mainstream fact-checking systems. To address this urgent issue, our project presents a robust and intelligent Fake News Detection System capable of analyzing and classifying news articles written in English, Tamil, and Hindi. Developed using Python and powered by Natural Language Processing (NLP) and Machine Learning algorithms, the system is designed to determine whether a given news article is real or fake with high accuracy.

A significant aspect of the project is its data acquisition strategy — since readily available multilingual datasets are limited, we employed advanced web scraping techniques to collect real-time news articles from credible sources and fact-checking websites in all three target languages. The scraped data was then subjected to thorough preprocessing tailored to each language, including tokenization, stopword removal, stemming, and normalization. We used vectorization techniques such as TF-IDF to convert the cleaned text into numerical format, enabling effective training of classification models like Logistic Regression and Naive Bayes.

The system architecture integrates these components into a seamless workflow, ending with a user-friendly interface that allows users to input news content and receive a prediction on its authenticity in real-time. This project not only contributes a functional and scalable solution to the fight against fake news but also promotes digital literacy and responsible information sharing by addressing the gap in multilingual content verification systems.

1.1 Motivation

The rapid dissemination of fake news has become one of the most pressing challenges in today’s digital world. Misinformation spreads faster than ever before, often without the necessary checks to ensure its truthfulness, leading to a cascade of negative consequences from public confusion and political manipulation to the erosion of trust in news sources. This phenomenon has become especially pronounced in regions with diverse languages, where localized fake news in regional dialects often goes unchecked by traditional verification systems.

Our motivation behind this project is to build a **scalable solution** that addresses this critical problem, especially in countries like India, where misinformation in multiple languages can significantly impact large populations. We are particularly motivated by the **lack of robust tools** available for identifying fake news in regional languages like Hindi and Tamil. By developing a multilingual Fake News Detection System, we aim to bridge the gap between language diversity and the fight against fake news.

Additionally, this project gives us the opportunity to explore technologies like Natural Language Processing (NLP) and Machine Learning, which offer powerful ways to process, analyze, and classify large volumes of textual data. We believe that empowering individuals with the ability to quickly verify the authenticity of the news they consume is a crucial step toward creating a more **informed** and **responsible digital society**. By focusing on regional language support, we are not only addressing a technological challenge but also contributing to the **digital literacy** of people who are most vulnerable to misinformation in their native languages.

1.2 Scope

The scope of this project, titled **Fake News Verifier**, is to develop a system capable of accurately classifying news articles as real or fake in multiple languages, including **English, Hindi, and Tamil**. The system will utilize **Natural Language Processing (NLP)** and **Machine Learning algorithms** to analyze and process news content. This will involve scraping news articles from credible sources and fact-checking websites to create a diverse multilingual dataset. The data will be preprocessed using techniques like **tokenization**, **stopword removal**, and **stemming**. The processed text will then be transformed into numerical features using **TF-IDF vectorization**, enabling the use of classification models like **Logistic Regression** and **Naive Bayes** for fake news detection. The system will provide an intuitive **user interface**, allowing users to input news content and receive real-time predictions on its authenticity. While the current scope focuses on three languages, there is potential for future expansion to include more languages and advanced techniques, such as deep learning models, to further enhance the system's accuracy and effectiveness .

1.3 Architecture

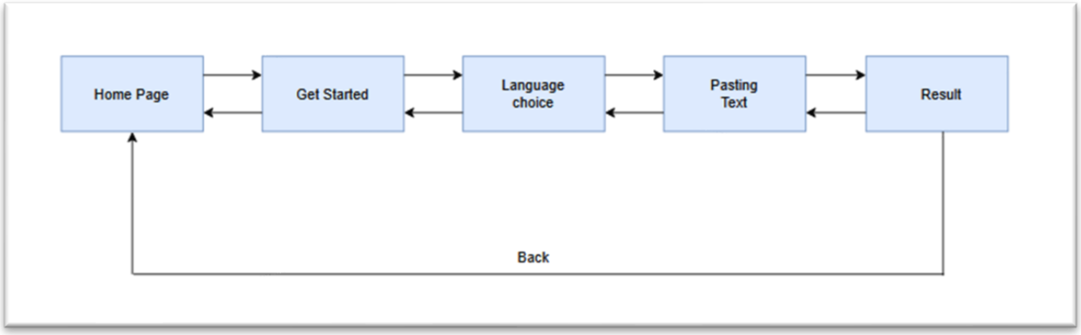


Fig 1.1 Architecture of Fake News Verifier

This flowchart outlines the Fake News Verifier user interface and workflow:

Home Page

- Starting point of the application
- Option: "Get Started"
- No back option (first page)

Get Started

- Guides user to begin verification
- Option: Go to Language Choice
- Back: Returns to Home Page

Language Choice

- User selects a language (English, Hindi, Tamil)
- Option: Go to Pasting Text

- Back: Returns to Get Started

Pasting Text

- User pastes or inputs the news content
- Option: Go to Result Page
- Back: Returns to Language Choice

Result

- Displays whether the news is Real or Fake
- Back: Returns to Pasting Text
- Back: Also has option to return to Home Page

1.4 Organization of the report

This report provides a comprehensive overview of the Fake News Verifier System, detailing its development, implementation, and evaluation:

Chapter 1 : This section introduces the Fake News Verifier project, outlining its objectives and the problem of rising misinformation in the digital age. It highlights the need for accurate and efficient fake news detection across multiple languages (English, Hindi, Tamil). The chapter also describes the key features of the system, such as real-time verification using machine learning and multilingual support.

Chapter 2 : The literature review explores previous research in the areas of fake news detection, Natural Language Processing (NLP), and machine learning. It discusses the effectiveness of various algorithms (like SVM, Logistic Regression, and ensemble models) in identifying misinformation. It also examines the role of tools like TF-IDF and frameworks like Indic NLP for processing content in regional languages.

Chapter 3 : This section outlines the approach used for developing and implementing the system, including web scraping (BeautifulSoup) for data collection, data preprocessing steps (cleaning, tokenization, stop word removal), and TF-IDF-based feature extraction. It details the software tools and frameworks used such as Pandas, Flask, and Scikit-learn, along with the model training and evaluation procedures.

Chapter 4 : This chapter presents the architectural framework of the Fake News Verifier, explaining its backend and frontend design. It details the workflow from input to prediction, describes how the multilingual functionality is supported, and includes diagrams and user interface screenshots to illustrate the system’s intuitive, user-friendly design.

Chapter 5 : This section analyzes the outcomes of the system’s implementation, presenting model performance metrics (accuracy, precision, recall) for various algorithms. It discusses the strengths and limitations of the current system, the effectiveness of the ensemble approach, and challenges in handling multilingual data. Recommendations for future improvements are also provided.

Chapter 6 : The conclusion summarizes the project’s key contributions and findings. It emphasizes the system's potential to combat fake news in a multilingual context and underlines its importance in journalism, education, and public awareness. The chapter also reflects on the broader impact of machine learning in digital media verification.

CHAPTER 2

LITERATURE SURVEY

2.1 Survey of Existing System

In recent years, the rapid spread of misinformation has led to a surge in research and development of fake news detection systems. These systems leverage advancements in Natural Language Processing (NLP), Machine Learning (ML), and Deep Learning (DL) to distinguish between legitimate and fake news content. Traditional approaches have relied heavily on classical ML models like Support Vector Machines, Decision Trees, Logistic Regression, Naive Bayes, and XGBoost. These models primarily use TF-IDF vectorization and other feature engineering techniques, often performing well on English-language datasets but struggling with deeper contextual understanding.

To overcome such limitations, hybrid deep learning models combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks have emerged. These systems incorporate FastText embeddings and transformer-based architectures like BERT, RoBERTa, and XLNet to better capture the semantic nuances of text. Explainability has also become a priority, with the integration of tools like LIME (Local Interpretable Model-Agnostic Explanations) and Latent Dirichlet Allocation (LDA) to interpret model predictions.

Another notable trend is the exploration of social context and propagation patterns for fake news detection. Some systems adopt Graph Neural Networks (GNNs) to analyze how news spreads across social media platforms, offering improved performance in identifying deceptive content based on topological data. Transformers are also being used to capture the sequential flow of news dissemination.

Cross-domain adaptability has also been addressed through Domain Adversarial Neural Networks (DANN), allowing models trained on one social media platform to generalize across others. However, most of these systems are either monolingual—focused predominantly on English—or lack real-time usability through web interfaces. Furthermore, deep learning models, while highly accurate, are often viewed as black boxes with limited interpretability. These gaps highlight the need for multilingual, explainable, and user-friendly fake news detection systems that can serve diverse populations and contexts.

Paper	Authors	Year	Key Contributions	Methodology/ Technique	Limitations
1) Advancing Fake News Detection: Hybrid Deep Learning With FastText and Explainable AI	Ehtesha Mhashmi	2024	A hybrid model combining CNN and LSTM enriched with FastText embeddings, Explainable AI using LIME and LDA,	FastText word embeddings, Machine Learning and Deep Learning methods, transformer-based models	DL-based models operate as black boxes; interpretability is addressed using LIME .
2) Topological and Sequential Neural Network Model for Detecting Fake News	Yoon-sik cho	2023	A Topological and Sequential Neural Network model (TSNN) for detecting fake news by capturing the diffusion patterns between source news and users in social networks.	Supernode approach, two-staged graph neural networks (leafGAT, time-decay GCN), transformer architecture.	Complex architecture; performance evaluated only on PolitiFact and GossipCop; interpretability not deeply explored.
3) Explainable Misinformation Detection Across Multiple	Gargi Joshi	2023	Integration of Domain Adversarial Neural Network (DANN) and	Domain Adversarial Neural Network (DANN), Local Interpretable Model-Agnostic	Focuses on COVID-19 misinformation; limited to two datasets (MiSoVac

Social Media Platforms			LIME for explainable and generalized misinformation detection across multiple social media platforms.	Explanations (LIME), Cross-domain adaptation.	and CoAid); lacks evaluation on misinformation topics.
4) Fake News Detection using Python and Machine Learning	Meenakshi mony	2024	Systematic exploration and evaluation of machine learning models; identifies the optimal ML algorithm for classifying articles as real or fake news.	Passive Aggressive Classifier, Random Forest, Logistic Regression, Decision Tree,	Limited to ML models; absence of deep learning; lacks discussion on multimodal or graph-based detection approaches.
5) Comprehensive Review on Fake News Detection With Deep Learning	M.d. saifur rahman	2021	Investigates advanced and state-of-the-art fake news detection mechanisms; provides comprehensive overview of dl	Attention, Generative Adversarial Networks, BERT, Natural Language Processing, CNN, RNN, LSTM.	Survey only; no new model proposed;

Table – 2.1- Existing literature survey

2.2 Limitation of existing system or research gap

1) **Language Dependency -** Most existing fake news detection systems primarily focus on English content, ignoring regional languages like Hindi, Tamil, etc. This limits their applicability in linguistically diverse regions like India, where misinformation is often spread in local languages.

2) **Lack of Real-Time Detection -** Several systems rely on static datasets and offline models, making them less effective for detecting rapidly evolving fake news in real-time scenarios.

3) **Inability to Handle Code-Mixed Text -** Many users post news in mixed-language formats (e.g., Hindi-English). Existing models often fail to accurately process and classify such content due to limited support for code-mixed NLP.

4) **Limited Use of Ensemble Learning -** Traditional systems rely on a single machine learning model, which may not capture the complexity of linguistic and contextual cues. Ensemble methods, which combine multiple models, are underutilized despite offering better accuracy and robustness.

5) **No Source Validation -** Many systems focus only on textual analysis and do not cross-reference the information with reliable external sources or trusted databases, reducing their verification accuracy.

6) **Scalability Issues -** Systems built for small datasets or single-language use cases may struggle to scale for large volumes of multilingual data from different sources like news sites, social media, and blogs.

7) **Limited User Interaction -** Most fake news detectors are backend tools with minimal or no user-friendly interfaces, making them less accessible to the general public or journalists.

2.3 Problem Statement & Objectives

Problem Statement

The proliferation of fake news across digital platforms, especially in regional languages, threatens public trust and social stability. Despite advancements in fake news detection, existing systems primarily target English content, leaving a major gap in multilingual detection. There is a lack of integrated tools that provide language-specific analysis, credibility scoring, and a user-friendly interface for real-world use .

Objectives

- To design and implement a multilingual fake news detection system for English, Tamil, Telugu, and Hindi.
- To develop custom NLP pipelines tailored to each supported language.
- To train machine learning models capable of classifying news as real or fake with high accuracy.
- To generate a credibility score that reflects the likelihood of a news article being authentic.
- To create a web-based application for public use, including journalists, researchers, and fact-checkers.
- To ensure the system is explainable, scalable, and easily extensible to more languages in the future.

2.4 Mini Project Contribution

1) Developed a multilingual fake news detection system supporting English, Tamil, Telugu, and Hindi.
2) Created dedicated preprocessing scripts for each language to handle tokenization, stop word removal, and normalization.
3) Integrated external NLP resources, including Indic language stopwords and word embeddings.
4) Designed and trained a machine learning model using TF-IDF vectorization and classifiers such as Logistic Regression and SVM.
5) Implemented a credibility scoring mechanism to quantify news authenticity.
6) Built a web application using Flask, providing an intuitive interface for uploading and verifying news.
7) Structured the backend to support multilingual input and return language-aware results.
8) Modularized the codebase using utils/, model/, preprocessing/, and web/ directories for scalability and clarity.
9) Included automated testing scripts for model accuracy, preprocessing correctness, and web route stability.

10) Ensured the system is practical for real-world use by journalists, social media users, and researchers.

CHAPTER 3

FAKE NEWS VERIFIER

3.1 Problem Statement

The proliferation of fake news across digital platforms, especially in regional languages, threatens public trust and social stability. Despite advancements in fake news detection, existing systems primarily target English content, leaving a major gap in multilingual detection. There is a lack of integrated tools that provide language-specific analysis, credibility scoring, and a user-friendly interface for real-world use .

3.2 Objectives

- To design and implement a multilingual fake news detection system for English, Tamil, Telugu, and Hindi.
- To develop custom NLP pipelines tailored to each supported language.
- To train machine learning models capable of classifying news as real or fake with high accuracy.
- To generate a credibility score that reflects the likelihood of a news article being authentic.
- To create a web-based application for public use, including journalists, researchers, and fact-checkers.
- To ensure the system is explainable, scalable, and easily extensible to more languages in the future.

3.3 Architecture / Framework

The architecture of the Fake News Verifier web application is designed to ensure accurate classification of news content, smooth user experience, and secure interaction. The system follows a modular and layered architecture, allowing flexibility, scalability, and ease of maintenance.

The key architectural components are as follows:

1. **User Interface (UI) Layer**

   This layer handles all interactions with the user and ensures a responsive and user-friendly experience.

- **HTML/CSS/JavaScript Web Pages:** Provides pages for the home screen, input form, and result display using attractive and intuitive layouts.
- **Input Forms:** Allows users to paste news content for verification.
- **Real-time Feedback:** Displays results dynamically with labels such as "Fake News" or "Real News", supported by styled alert messages.

2. Presentation Layer

This layer connects the user interface with the core backend logic.

- Flask Routing and Templates: Uses the Flask web framework to handle URL routing, receive user inputs, and return prediction results dynamically using Jinja2 templating.
- **Dynamic Rendering**: Ensures that the results are displayed along with the input news for clarity and user validation.

3. Business Logic Layer

1. This is the core engine of the application, performing all processing and classification tasks.
2. Preprocessing Module: Performs text cleaning operations such as lowercasing, punctuation removal, tokenization, and stopword filtering.
3. Classification Model: Utilizes a Machine Learning or NLP model (e.g., Logistic Regression, Random Forest, or BERT) trained on labeled datasets to classify news as fake or real.
4. Evaluation Metrics: Incorporates metrics such as accuracy, precision, recall, F1-score, and confusion matrix to assess model performance.

4. Data Layer

1. This layer manages all data-related operations and resources.

2. Dataset Storage: Stores the multilingual news datasets (English, Tamil, Hindi) used for training and evaluation.

3. Model Files: Stores the trained models in serialized formats (e.g., .pkl or .joblib) for quick inference.

4. Preprocessed Files: Maintains cleaned and balanced datasets to improve model learning and reduce bias.

5. Integration Layer

a. This layer facilitates the connection with third-party tools and services.

b. Python Libraries: Integrates libraries such as pandas, scikit-learn, nltk for data handling, ML modeling, and visualization.

c. Deployment Platform (Optional): Can be deployed on platforms like Heroku, AWS EC2 for public access and scalability.

CHAPTER 4

DESIGN AND IMPLEMENTATION

4.1 Methodology

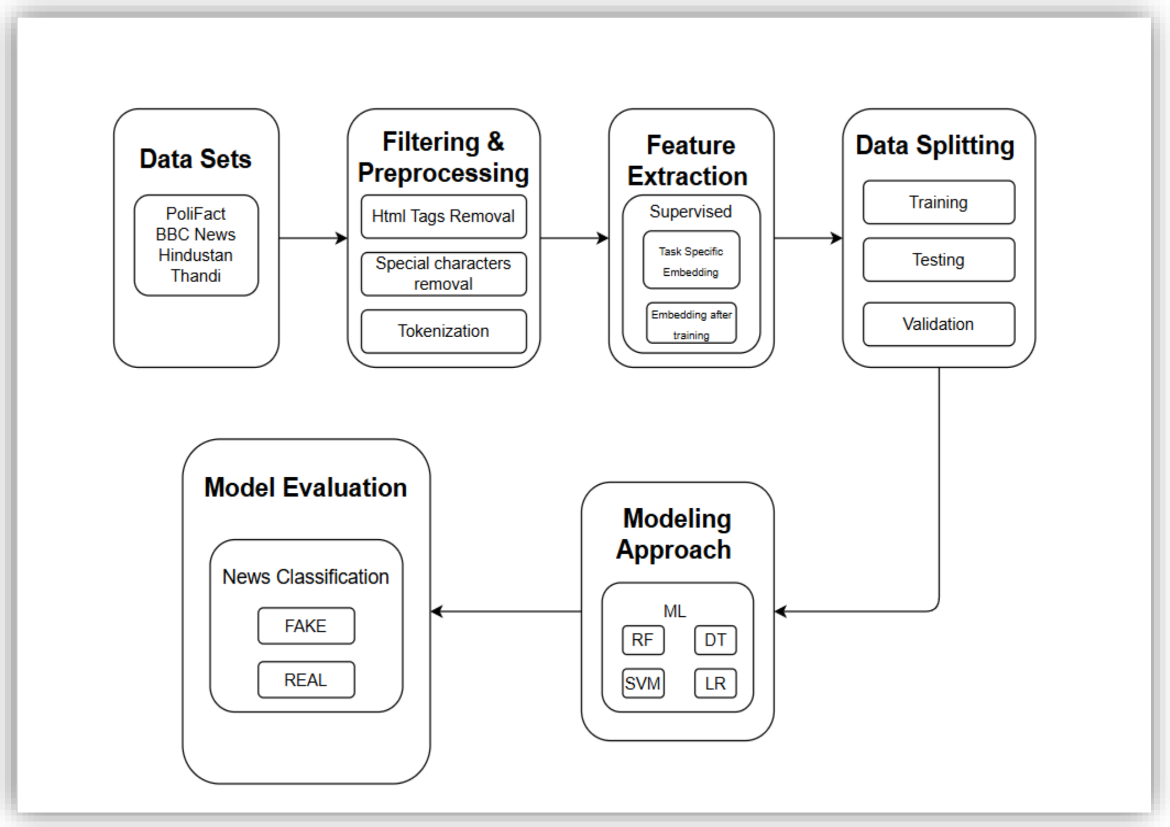


Fig 4.1 Methodology of Fake News Verifier

4.2 Details of Hardware & Software

1. Programming Language

The Fake News Verifier project is built using Python (version 3.8 or above). Python is chosen for its simplicity, readability, and extensive support for data science and machine learning libraries.

2. Web Framework

The web application is developed using **Flask**, a lightweight and flexible web framework in Python. Flask simplifies the development of web applications making it suitable for rapid deployment and testing of the machine learning model.

3. Machine Learning Libraries

Scikit-learn is used for implementing machine learning algorithms like Naive Bayes or Logistic Regression. It also provides tools like TfidfVectorizer for converting raw text data into numerical features, which are essential for model training.

4. Data Processing Libraries

Pandas is used for data loading, cleaning, and manipulation, allowing efficient handling of structured data. **NumPy** supports mathematical operations and array management, which are frequently needed during preprocessing and model evaluation.

5. Model Serialization

To save and load trained models, libraries such as **Pickle** is used. These tools allow the trained model to be stored as a file and reloaded later for inference without retraining

6. Frontend Technologies

The frontend is developed using **HTML** and **CSS** for structuring and styling the web pages. To ensure responsiveness and better design, **Bootstrap**, a front-end CSS framework, is used to style buttons, forms, and layouts effectively.

7. Development Environment

Jupyter Notebook is used during the model development phase for experimenting and visualizing outputs. IDEs like **Visual Studio Code** or **PyCharm** are used for building and maintaining the backend code.

8. Version Control System

Git is used for tracking code changes, and **GitHub** is used as a remote repository to host the project. This setup helps manage versions, share code, and collaborate with others if required.

9. Deployment Tools

The model and web application can be deployed using **AWS EC2**. These platforms make the web application accessible to users over the internet.

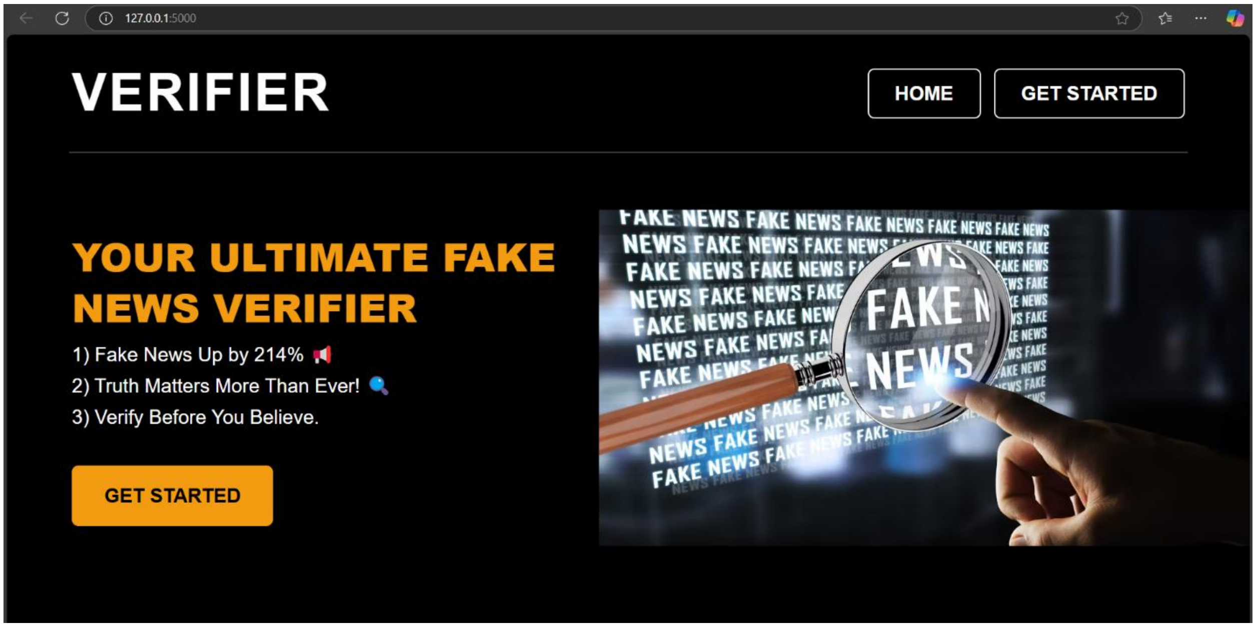
Component	Description	Technologies/Tools	Output
News Input	Users input news text manually. Accepts plain text from social media, news sites, or copy-paste.	Manual Entry, File Upload (TXT), Web Scraping (optional)	Raw text of news article
Text Preprocessing	Cleans and prepares the news text by removing stop words, punctuation, and converting to lowercase. Tokenization and lemmatization are also done.	Python, NLTK, Pandas,	Cleaned and structured news content
Feature Extraction	Converts cleaned text into numerical vectors to make it usable by machine learning algorithms.	Scikit-learn (TfidfVectorizer, CountVectorizer)	Vectorized representation of news text
Classification (AI Model)	Uses a trained machine learning model to classify the input as "Fake" or "Real".	Scikit-learn, Logistic Regression, Naive Bayes, Pickle	Classification label (Fake or Real)
User Interface (UI)	Provides a user-friendly web interface for inputting news and viewing results.	HTML, CSS, Bootstrap, Flask (Jinja2 Templates)	Webpage for news input and result display
Backend	Manages communication between frontend and the ML model;.	Flask, , Python, Gunicorn (optional)	Result from model sent to frontend
Deployment	The system is hosted on a cloud platform to allow anytime access from any device.	AWS EC2 (free tier)	Live hosted application accessible via browser

Table-4.2 Designing of software

CHAPTER 5

EXPERIMENTS AND RESULT

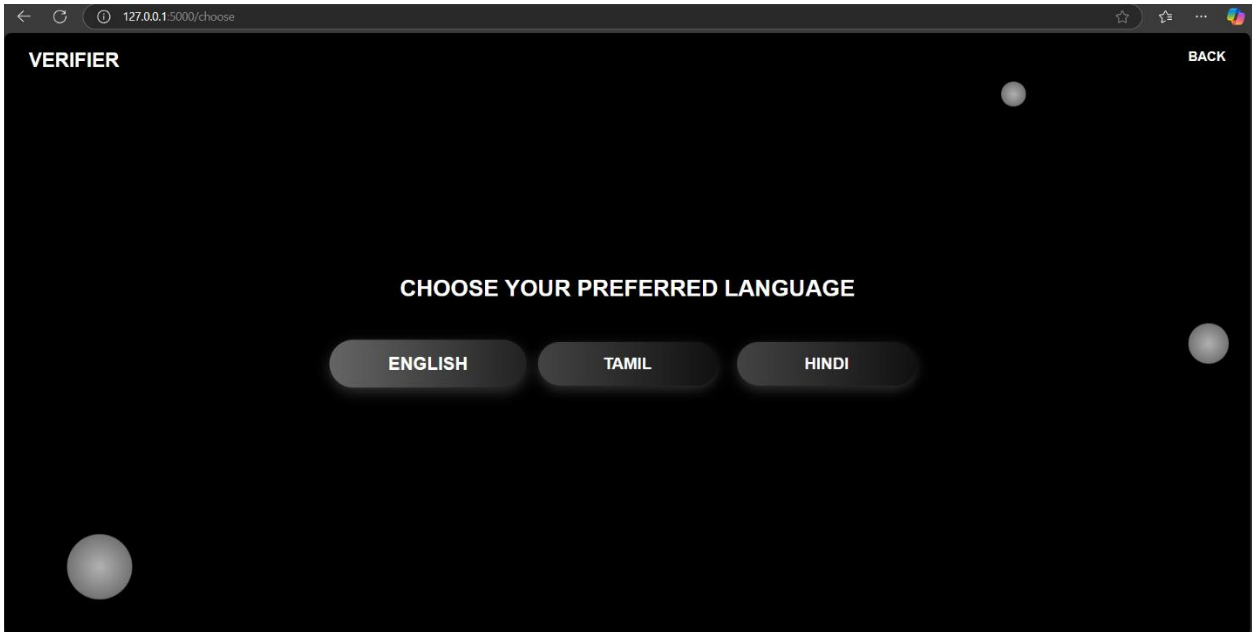
5.1 Experiment



YOUR ULTIMATE FAKE NEWS VERIFIER

1) Fake News Up by 214% 📢
2) Truth Matters More Than Ever! 🔍
3) Verify Before You Believe.

Fig . 5.1. Home Page



CHOOSE YOUR PREFERRED LANGUAGE

Fig . 5.2. Language Options

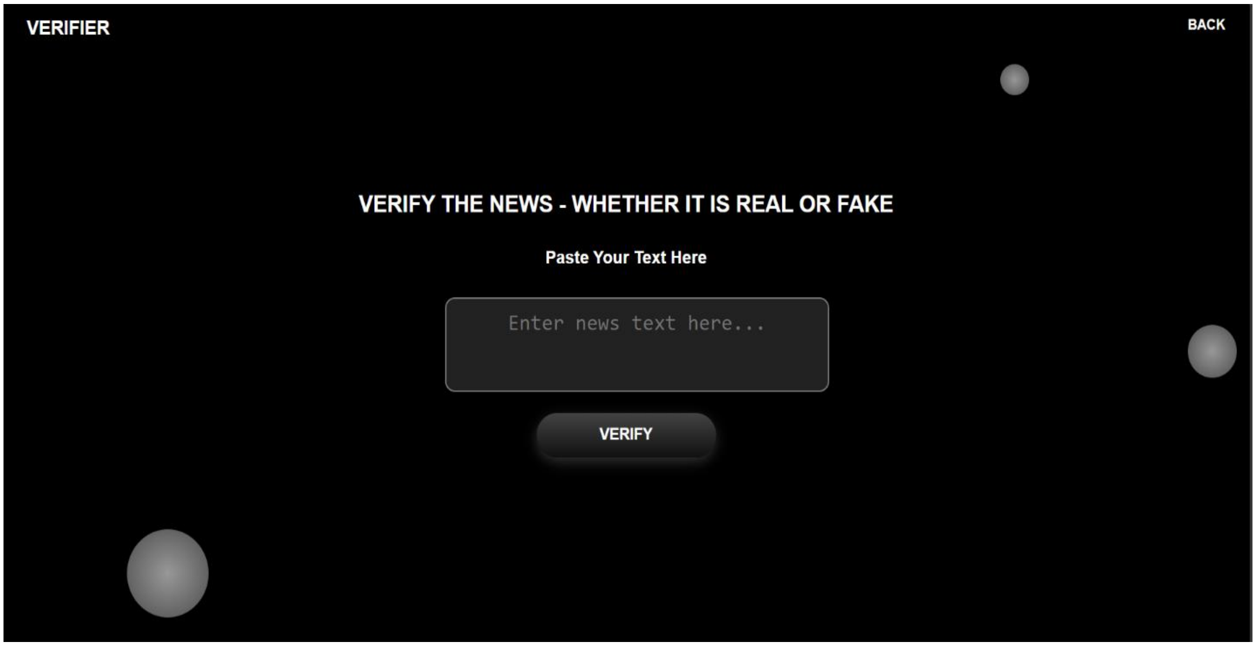
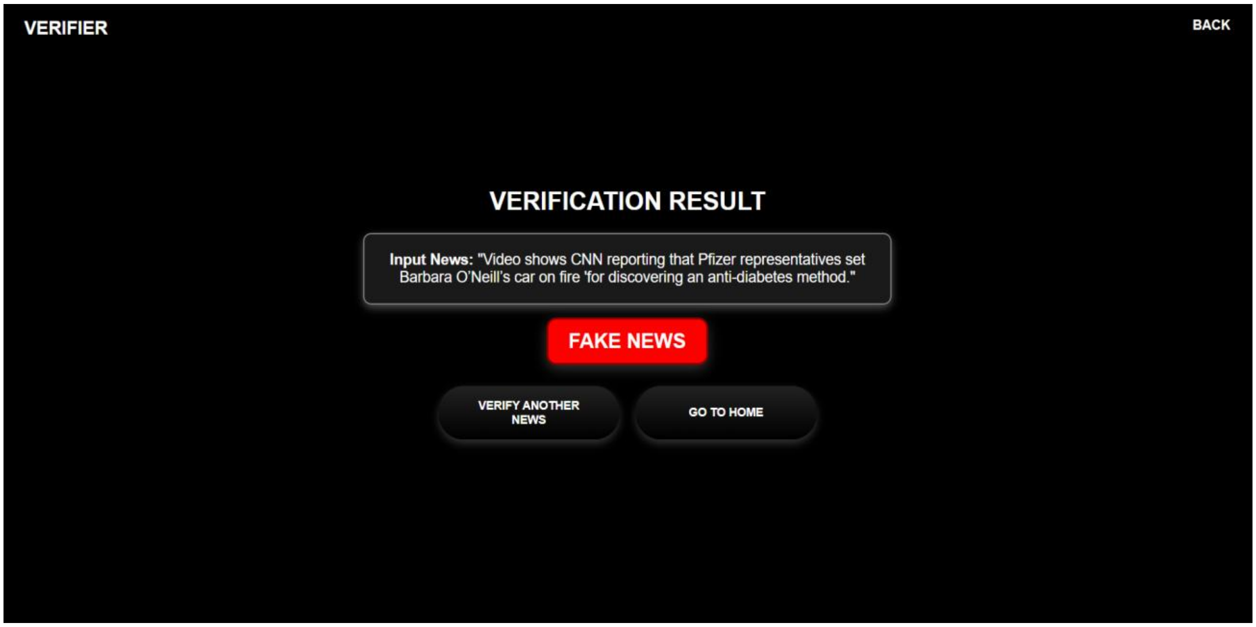


Fig . 5.3. Input Field

5.2 Result



Input News: "Video shows CNN reporting that Pfizer representatives set Barbara O'Neill's car on fire 'for discovering an anti-diabetes method."

Fig . 5.4. Result

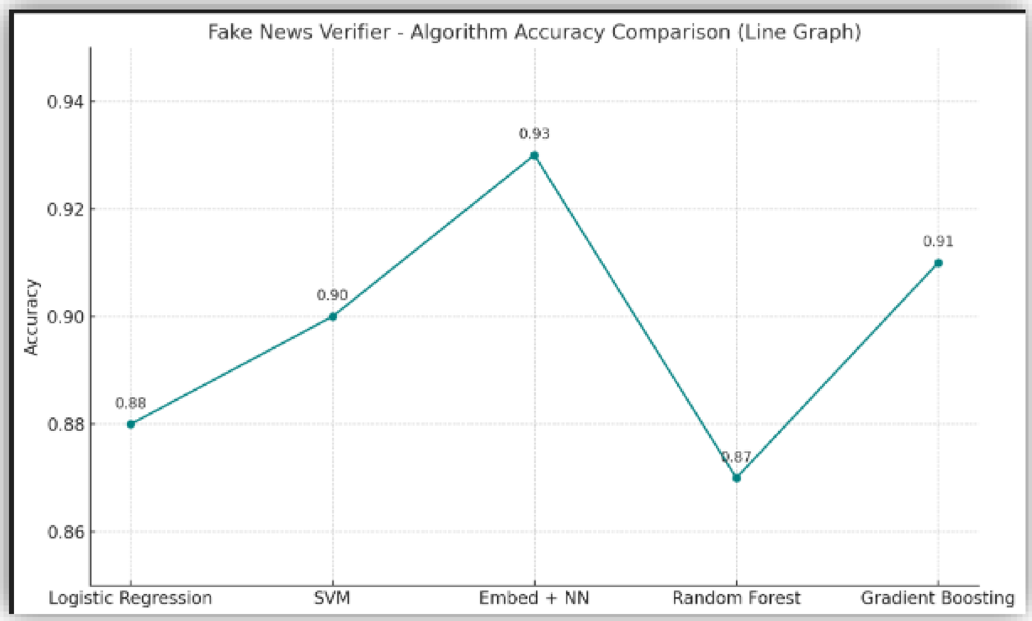


Fig . 5.5. Accuracy Graph Using Multiple Algorithms

CHAPTER 6

CONCLUSION AND FUTURE WORK

The **Fake News Verifier** project effectively addresses the growing concern of misinformation spread through digital platforms. By leveraging Natural Language Processing (NLP) techniques and machine learning classification models, the system provides a reliable and automated solution to identify whether a news item is fake or real. This tool is especially valuable in today's digital age, where the rapid spread of false information can lead to significant social, political, and economic consequences.

Throughout the development phase, the project successfully demonstrated its ability to process user-input news articles, clean and extract meaningful features from the text, and classify the news with high accuracy. The intuitive user interface ensures ease of use, allowing users—even without technical expertise—to quickly verify the authenticity of news articles.

This project stands as an important step toward promoting digital literacy and responsible information consumption. By providing a simple yet powerful tool to verify news authenticity, the Fake News Verifier contributes to a more informed and aware society.

FUTURE WORK

While the current version of the Fake News Verifier meets its core objectives, several enhancements can be explored to improve functionality, scalability, and user experience:

- **Multilingual Support**: Extend the system to handle news in regional languages like Tamil, Hindi, and Telugu, ensuring broader accessibility.
- **Deep Learning Integration**: Use more advanced models such as LSTM, BERT, or transformers to improve accuracy and handle complex sentence structures.
- **News Source Analysis**: Incorporate source credibility analysis to evaluate the trustworthiness of the news publisher or platform.
- **Real-time Fact Checking**: Integrate with APIs or databases of verified news (like Snopes or PolitiFact) to cross-reference and validate content in real time.
- **Mobile Application**: Develop a lightweight mobile app for on-the-go news verification, increasing reach and usability.
- **Cloud Deployment**: Host the application on platforms like AWS, Render, or Heroku for continuous and scalable access.
- **User Feedback System**: Allow users to report incorrect predictions and contribute to

improving the model through user-driven learning.

- **Data Security**: Implement advanced security features like encryption and secure login to protect user inputs and outputs.

By incorporating these improvements, the Fake News Verifier can evolve into a more robust and indispensable tool in combating misinformation, ultimately promoting truth and trust in digital communication.

REFERENCES

[1]. Hashmi E, Yayilgan SY, Yamin MM, Ali S, and Abomhara M (2024). "Advancing Fake News Detection: Hybrid Deep Learning With FastText and Explainable AI." IEEE Access, 12, pp. 44462–44480. https://doi.org/10.1109/ACCESS.2024.3381038.

[2]. Jung D, Kim E, and Cho YS (2023). "Topological and Sequential Neural Network Model for Detecting Fake News." IEEE Access, 11, pp. 143925–143938. https://doi.org/10.1109/ACCESS.2023.3343843.

[3]. Joshi G, Srivastava A, Yagnik B, Hasan M, Saiyed Z, Gabralla LA, Abraham A, Walambe R, and Kotecha K (2023). "Explainable Misinformation Detection Across Multiple Social Media Platforms." IEEE Access, 11, pp. 23634–23647. https://doi.org/10.1109/ACCESS.2023.3251892.

[4]. Jouhar J, Pratap A, Tijo N, and Mony M (2024). "Fake News Detection using Python and Machine Learning." Procedia Computer Science, 233, pp. 763–771. https://doi.org/10.1016/j.procs.2024.03.265.

[5]. Mridha MF, Keya AJ, Hamid MA, Monowar MM, and Rahman MS (2021). "A Comprehensive Review on Fake News Detection With Deep Learning." IEEE Access, 9, pp. 156151–156173. https://doi.org/10.1109/ACCESS.2021.3129329.