

# A Lightweight Framework for Fake News Classification Using TF-IDF and SVM

<sup>1</sup>Mr.A. SrinivasaRao, <sup>2</sup>Anantha Lakshmi Madduri, <sup>3</sup>Siva NagaLakshmi Lakkimsetty, <sup>4</sup>Siva Sankar Achi, <sup>5</sup>Asif Shaik, <sup>6</sup>Chandrika Vattikunta

*1Assistant Professor, Department Of CSE(AI&ML), Tirumala Engineering College , AP, 2, 3, 4, 5, 6UG Student, Department Of CSE(AI&ML), Tirumala Engineering College , AP*

*[1srinu.veeru@gmail.com](mailto:srinu.veeru@gmail.com), [2ananthumadduri143@gmail.com](mailto:ananthumadduri143@gmail.com), [3sivanagalakshmi126@gmail.com](mailto:sivanagalakshmi126@gmail.com), [4shivashankaracchi@gmail.com](mailto:shivashankaracchi@gmail.com), [5shaikasif200427@gmail.com](mailto:shaikasif200427@gmail.com), [6vattikuntachandrika0708@gmail.com](mailto:vattikuntachandrika0708@gmail.com)*

## Abstract

*The Fake News Detection System is a machine learning-based web application developed in Python to classify news as real or fake. It uses NLP techniques for preprocessing and TF-IDF for feature extraction. Supervised models like Linear SVM and LSTM are trained on labeled datasets for accurate classification. The system provides real-time analysis to help prevent the spread of misinformation online. Overall, it improves news authenticity verification using data analysis and machine learning techniques.*

*Index Terms—Machine Learning, Fake News Detection, Multimodal Classification, TF-IDF, Linear SVM, Large Language Model, Natural Language Processing, Mis-information, Web Application., Explainable AI.*

## Introduction

In the modern digital age, the internet has become the primary source of information for millions of people around the world. Social media platforms, online news portals, blogs, and forums allow users to access and share information instantly. While this has significantly improved communication and awareness, it has also led to the rapid spread of fake news, which has become a serious global issue.

Fake news refers to false or misleading information that is presented as legitimate news. It is often created intentionally to deceive readers, manipulate public opinion, or generate financial or political gain. With the increasing popularity of social media platforms such as Facebook, Twitter, and WhatsApp, fake news spreads much faster than traditional news, making it difficult to control or verify its authenticity.

After preprocessing, the system uses TF-IDF (Term Frequency-Inverse Document Frequency) for feature extraction. TF-IDF converts textual data into numerical form by assigning weights to words based on their importance. Words that appear frequently in a document but are rare across the genuine news articles. This dataset typically includes headlines, full text, and corresponding dataset are given higher importance. This technique helps the model understand the significance of different words in determining whether a piece of news is real or fake.

The classification of news articles is carried out using machine learning algorithms such as Logistic Regression, Naive Bayes, or Support Vector Machines (SVM). These algorithms are

trained on a dataset containing labeled examples of real and fake news. During training, the model learns to distinguish between the two categories by analyzing patterns in the data. Once trained, the model can predict the authenticity of new, unseen news articles.

## **I. METHODOLOGY:**

- The proposed fake news detection system follows a structured workflow consisting of data collection, preprocessing, feature extraction, model training, evaluation, and deployment.

- **A. Data Collection**

The initial phase involves collecting a labelled dataset containing both fake and labels. Data is gathered from multiple sources to ensure diversity in writing styles, tone, and content. A well-balanced and high-quality dataset is essential for building a model that performs reliably on unseen data.

- **B. Preprocessing**

Raw textual data often contains noise and inconsistencies that can affect model performance. Therefore, preprocessing is applied to clean and standardize the data. This includes removing punctuation, special characters, numbers, and irrelevant elements. All text is converted to lowercase to maintain uniformity.

Stop words are eliminated to focus on meaningful content, and words are normalized using lemmatization or stemming. Missing or incomplete data entries are also handled to ensure dataset consistency.

- **C. Feature Extraction**

To enable machine learning algorithms to process text, it must be converted into numerical form. Feature extraction techniques such as word frequency representation and TF-IDF are used to capture the importance of words. Additionally, contextual relationships between words may be considered to better understand the semantics of the text. These features serve as the input for training the classification model.

- **D. Model Selection and Training**

In this stage, an appropriate machine learning algorithm is selected and trained using the extracted features. Different models offer varying advantages, such as interpretability or the ability to capture complex patterns. Ensemble techniques may also be used to improve performance. The model is trained by adjusting its parameters to minimize classification errors and improve prediction accuracy.

- **E. Model Evaluation**

The trained model is evaluated using performance metrics such as accuracy, precision, recall, and F1-score. These metrics provide a comprehensive understanding of the model's effectiveness. A confusion matrix is also used to analyse correct and incorrect predictions. Testing the model on unseen data ensures robustness and helps avoid overfitting.

## **F. Deployment**

After successful evaluation, the model is integrated into a real-time system, such as a web or mobile application. Users can input news content and receive instant predictions indicating whether the news is fake or real. This stage ensures that the system is practical and accessible for real-world use.

## **G. Continuous Improvement**

Since fake news patterns evolve over time, the system requires regular updates. Periodic retraining with new data helps maintain accuracy and adaptability. Continuous refinement enables the model to handle emerging misinformation trends effectively.

## **II. DESIGN:**

The fake news detection system is designed using a modular architecture composed of three key components: the web interface, the core processing unit, and the classification module. These components work together to process user input and generate accurate predictions.

### **A. Web Interface Component**

The web interface acts as the front-end layer that enables user interaction with the system. It serves as the entry point where users can input news content, such as headlines or full articles, for verification. The entered text is transmitted to the backend through a REST API for further processing. Once the analysis is completed, the interface displays the results, indicating whether the news is fake or real along with a confidence score. Additionally, the interface may provide explanatory details, such as highlighting suspicious phrases or misleading patterns, helping users better understand the classification outcome.

### **B. Core Processing Unit**

The core processing unit is responsible for executing all machine learning operations. When the input text is received, it undergoes preprocessing steps such as cleaning, lowercasing, stopword removal, and lemmatization to ensure consistency. The processed text is then converted into numerical form using feature extraction techniques like TF-IDF. These features are passed to a trained machine learning model, which analyses linguistic patterns and determines whether the news content is authentic or misleading.

### **C. Classification Module**

The classification module interprets the output generated by the machine learning model and produces the final result. It assigns a label (fake or real) along with a probability score that reflects the confidence of the prediction. In addition, the module may retrieve supporting information from a database, such as known fake news patterns or similar examples, to provide better explanation. The final results, along with any additional insights, are sent back to the web interface for display.

## D. System Architecture

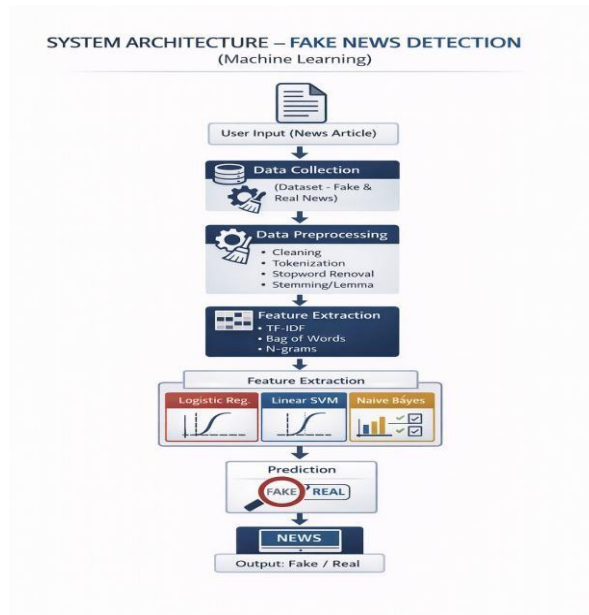


FIG 1: SYSTEM ARCHITECHTURE

## III. IMPLEMENTATION

### A. Passive-Aggressive Classifier

The Passive-Aggressive Classifier is used for fake news detection due to its fast learning and efficiency. It updates the model only when a prediction is incorrect, allowing quick adaptation to new data and improving classification accuracy.

- 1) **Model Definition:** The model is implemented using the Scikit-learn library. Parameters such as iteration count and regularization are set to balance accuracy and complexity.
- 2) **TF-IDF Vectorization:** Text data is converted into numerical form using TF-IDF, which highlights important words while reducing the impact of common terms.
- 3) **Training:** The model is trained on processed data and learns patterns by adjusting weights based on errors, enabling accurate classification of fake and real news.
- 4) **Evaluation:** Performance is measured using accuracy, precision, recall, and F1-score. A confusion matrix is also used to analyse results.

### B. Data Preprocessing

- 1) **Lemmatization:** Words are reduced to their base form to ensure consistency and reduce redundancy.
- 2) **Stopword Removal:** Common words with little meaning are removed to reduce noise and improve model efficiency.

#### IV. Experimental Results and Analysis

**A. User Interface and Deployment:** The front- end interface displays the classification results in a clear and user-friendly manner, indicating whether the news is real or fake along with a probability score. It also provides explanations for the decision, such as identifying sensational language or absence of reliable sources.

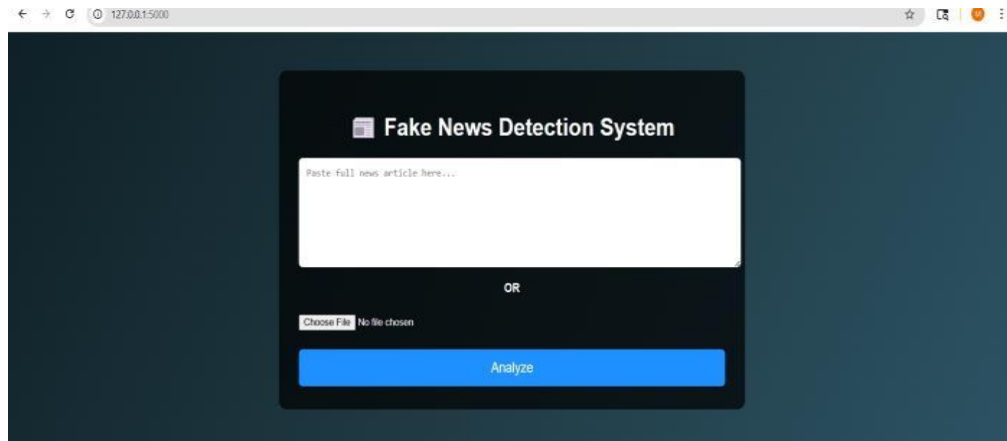
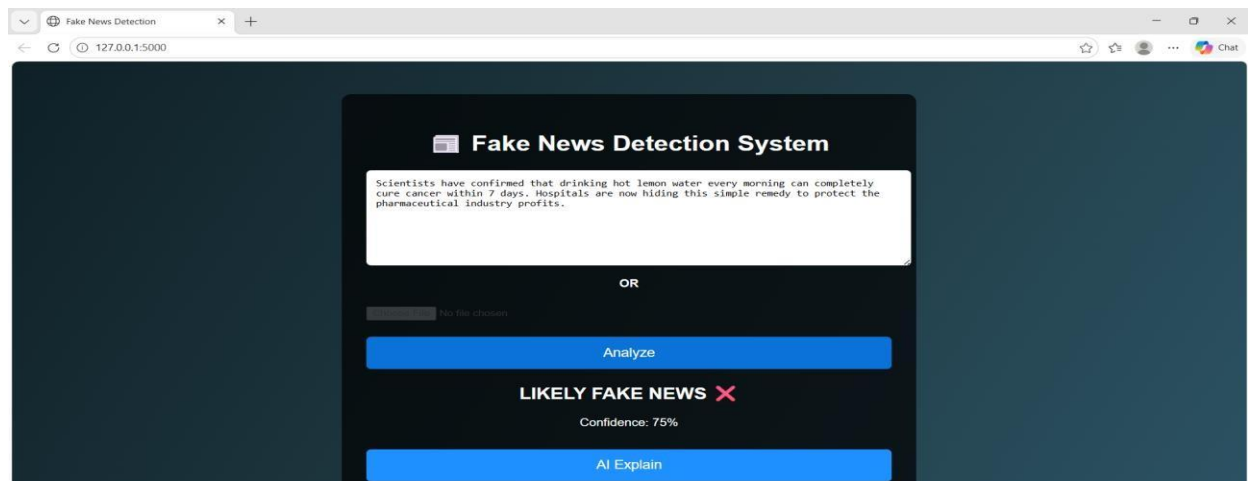
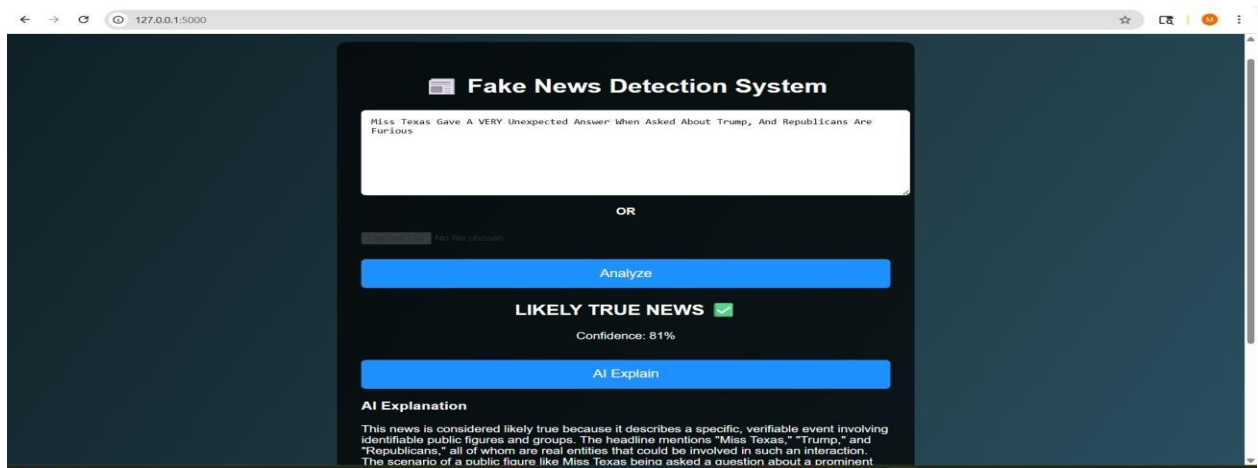


Fig 2.1: Home Screen for Fake News Detection



**B. Performance Metrics and Comparison** The model’s performance was evaluated by comparing it with standard baseline classifiers. An 80:20 train test split was used to ensure fair and unbiased evaluation results.

**Table 1: Performance Comparison of Models**

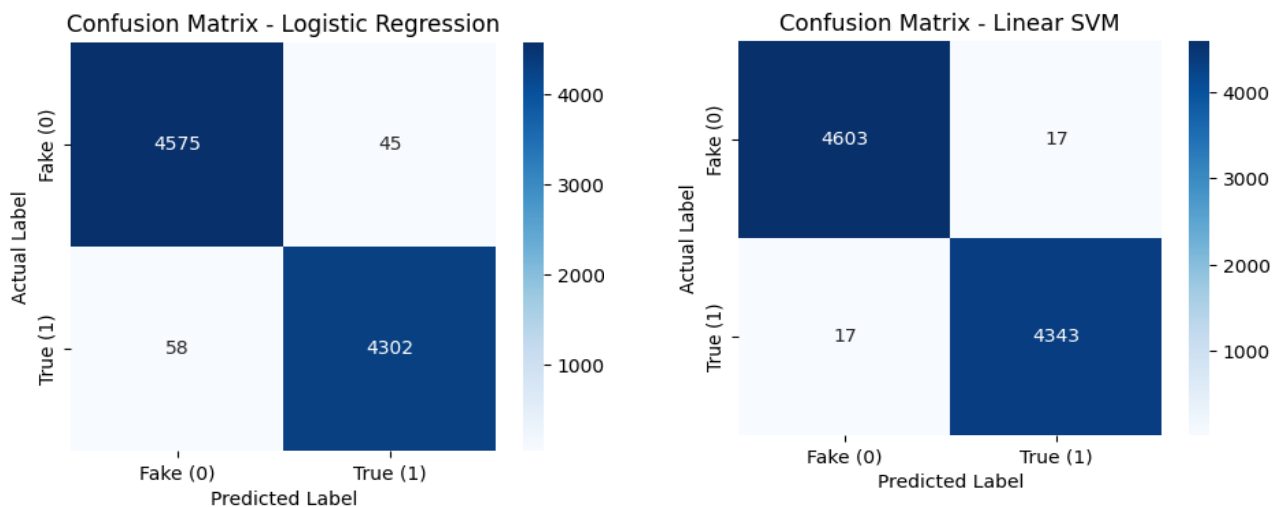
	Algorithm Name	Accuracy	Precisio n	Recall	FSCOR E
1	Logistic Regression	98.853	99.00	99.00	99.00
2	Naive Bayes	95.434	95.00	95.00	95.00
3	Linear SVM	99.621	100.0	100.0	100.0
4	Random Forest	99.098	99.00	99.00	99.00

**C. Feature Correlation and Transparency:**

Feature correlation analysis helps identify relationships between input features and their impact



on predictions. It improves transparency by showing how different features contribute to the model’s decision-making process.



*Fig 3.1: Confusion Matrix for Logistic Regression*

## **V. Conclusion**

The proposed fake news detection framework successfully combines text preprocessing, feature extraction, and classification techniques to deliver accurate predictions. The results indicate that machine learning-based approaches can significantly improve the identification of misleading information. Future enhancements can further strengthen the system's robustness and applicability.

## **References**

- 1] S. Nakamoto, "*Bitcoin: A Peer-to-Peer Electronic Cash System*," 2008.  
Available: <https://bitcoin.org/bitcoin.pdf>
- [2] D. Chaum, "*Secret-Ballot Receipts: True Voter- Verifiable Elections*," IEEE Security & Privacy, 2004.  
Available: <https://www.pivotal.io/sites/default/files/Chaum-Vote.pdf>
- [3] A. K. Jain, A. Ross, and S. Prabhakar, "*An Introduction to Biometric Recognition*," IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 1, pp. 4–20, 2004.
- [4] D. E. King, "*Dlib-ML: A Machine Learning Toolkit*," Journal of Machine Learning Research, 2009.  
Available: <http://dlib.net>