Fake News Prediction Based on Natural Language Processing(Nlp) And Machine Learning

S. SUGANTHI SHYAMASP[1], M. DHARANI RAJA[2], G. JAYASURIYA[3]

[1, 2, 3]Dept of CSE

[1, 2, 3] K.L.N.College of Engineering, Sivagangai,India

Abstract- The rapid dissemination of misinformation across social and digital media platforms has created a growing demand for automated mechanisms to detect and prevent the spread of fake news. This paper presents a machine learning–based Fake News Prediction System that utilizes Natural Language Processing (NLP) techniques to classify news articles as real or fake. The system preprocesses textual content using tokenization, stop-word removal, and lemmatization, and transforms it into feature vectors through Term Frequency–Inverse Document Frequency (TF-IDF) representation. A Passive Aggressive Classifier (PAC) is employed to train and predict labels with high efficiency and low computational cost. The proposed approach achieves competitive accuracy while maintaining interpretability and scalability. A lightweight Flask web interface is developed for real-time user interaction, enabling non-technical users to input text and instantly view classification results. Experimental evaluation demonstrates that the system effectively distinguishes false information from legitimate news, contributing to the reliability of online information and enhancing trust in digital communication

Keywords- Fake News Detection, Natural Language Processing (NLP), Machine Learning, TF-IDF, Passive Aggressive Classifier, Text Classification, Flask Web Application, Data Preprocessing

I. INTRODUCTION

Fake news spreads quickly on digital platforms and can mislead people, affecting society, politics, and decision-making. Traditional manual checking methods are slow and cannot handle large volumes of news. This project develops an intelligent **Fake News Prediction System** using **Natural Language Processing (NLP)** and **Machine Learning (ML)** to automatically classify news as real or fake. The system preprocesses text through tokenization, stop-word removal, and lemmatization, converts it into **TF-IDF** features, and uses a **Passive Aggressive Classifier (PAC)** for accurate predictions. A **Flask web interface** allows users to input articles and get real-time results. The system is efficient, scalable, and interpretable, providing a practical solution to reduce the spread of misinformation online.

II. LITERATURE SURVEY

A review of existing literature indicates that fake news detection has evolved from simple rule-based methods to sophisticated machine learning and deep learning techniques. Early studies by Shu et al. (2017) and Pérez-Rosas et al. (2018) demonstrated that classical machine learning classifiers such as Naive Bayes, SVM, and Random Forest could effectively identify fake news using textual features, but they often struggled with high-dimensional data and limited generalization across domains. Recent research by Wang (2020) and Zhou &Zafarani (2021) explored deep learning approaches, including CNNs, LSTMs, and transformer-based models like BERT, which achieved higher predictive accuracy but introduced challenges in interpretability, computational cost, and deployment complexity. This project addresses these gaps by implementing a lightweight, efficient pipeline using NLP-based preprocessing (tokenization, stop-word removal, lemmatization), TF-IDF feature extraction, and a Passive Aggressive Classifier, striking a balance between accuracy, speed, and interpretability, while providing a scalable solution suitable for real-time web deployment.

III. EXISTING METHODOLOGY

Existing methods for fake news detection typically rely on traditional machine learning models and manual feature extraction. Techniques such as Naive Bayes, Logistic Regression, and Support Vector Machines have been used with text-based features like bag-of-words, TF-IDF, and sentiment scores. Some approaches incorporate social media metadata or user behavior patterns to improve accuracy. While these methods can achieve reasonable performance, they often require extensive preprocessing, large datasets, and significant computational resources. Many existing systems also lack real-time prediction capabilities, making them less practical for dynamic news environments. These limitations highlight the need for an efficient, scalable, and automated approach that can process news content quickly while maintaining high accuracy and interpretability.

IV. PROPOSED METHODOLOGY

The proposed system follows a structured workflow for accurate fraud detection. It begins with data collection and preprocessing to clean inconsistencies and noise, followed by feature engineering to extract textual, structural, and behavioral fraud indicators. An ensemble of algorithmsbuilds a robust predictive model, with final classification determined through a weighted voting mechanism and confidence scoring. The model is deployed via a Flask-based web application, enabling real-time fraud prediction through an intuitive interface and API integration.

The system architecture begins with the Data Ingestion module, which collects job postings from different online sources. These records are passed through Preprocessing and Feature Engineering stages, where the data is cleaned, transformed, and structured into meaningful features. The ensemble machine learning model then analyzes both text and metadata to produce a prediction. External metadata verification is also incorporated to improve detection accuracy. Finally, the Prediction module provides a classification output with a confidence score, which is displayed through a web interface for end users.
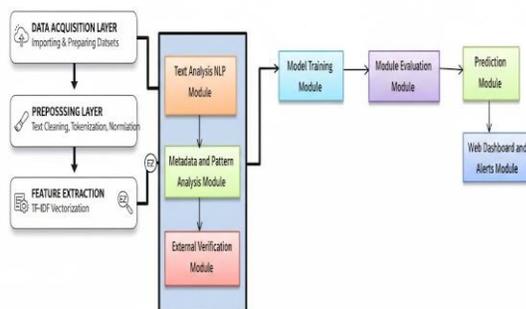
Figure 1 : System architecture of our proposed module

V. SYSTEM PROCESS FLOW

The **Fake News Prediction System** follows a modular, NLP-driven workflow designed to transform unstructured text data into a classified output that indicates whether a news article is *real* or *fake*. The complete process, from data collection to prediction, can be summarized through the following key stages:

1. Data Ingestion Module

The workflow begins with **data ingestion**, which serves as the input layer of the system. Datasets are collected from verified open sources or uploaded manually in CSV format through the application interface. The ingestion module ensures that data is properly formatted and free from duplicates or incomplete records before entering the preprocessing phase. This module plays a crucial role in

maintaining the quality and integrity of data used for model training and testing.

2. Preprocessing and Data Cleaning Module

In this phase, the raw news articles are processed using **Natural Language Processing (NLP)** techniques. The text is cleaned by removing special characters, punctuation, and HTML tags. Tokenization is applied to split sentences into words, followed by **stop-word removal** to eliminate common but insignificant words such as "the," "is," and "and." Finally, **lemmatization** standardizes words to their base form, improving the consistency of data representation. The output of this stage is clean, structured text that can be effectively used for analysis.

3. Feature Extraction Module

After preprocessing, the cleaned text is converted into numerical feature vectors using the **TF-IDF (Term Frequency–Inverse Document Frequency)** technique. This method calculates the importance of each word based on its frequency within a document and across the dataset. The resulting feature matrix captures the key linguistic patterns and relationships between words, allowing the system to distinguish between genuine and deceptive writing styles.

4.Model Training Module

In this stage, machine learning algorithms are trained using the extracted TF-IDF features. The **Passive Aggressive Classifier (PAC)** is selected for its efficiency and ability to handle large datasets. The model learns to classify news articles as real or fake by analyzinglabeled data and adjusting its parameters during multiple iterations. Once trained, the model is saved for future use and can be retrained with updated data to improve accuracy.

5.Evaluation and Testing Module

After Training, The Model's Performance Is Evaluated Using Metrics Such As**Accuracy, Precision, Recall, And F1-Score**. A Confusion Matrix Is Generated To Visualize Correct And Incorrect Predictions. These Evaluations Help Fine-Tune The Model, Ensuring That It Generalizes Well To Unseen Data And Reduces False Classifications.

6.Prediction and User Interaction Module

The trained model is deployed through a **Flask-based web interface**. Users can enter a news headline or paste a full

article, and the system instantly processes it to display the result as either *Real* or *Fake*. The interface is simple and interactive, designed for accessibility across devices. This real-time prediction capability demonstrates the practical value of the model for both research and everyday use.

7.Dashboard and Visualization Module

Finally, the system provides an optional visualization dashboard where users or researchers can view metrics such as accuracy percentage, dataset size, and prediction confidence. Graphical charts, such as bar graphs and pie charts, illustrate model performance and prediction distribution. This module ensures transparency and provides valuable insights into how the system evaluates and classifies information.

VI. RESULT AND DISCUSSIONS

The proposed **Fake News Prediction System** was tested using a labeled dataset containing both *real* and *fake* news articles. The data was divided into **80% for training** and **20% for testing** to evaluate model performance. The system used the **Passive Aggressive Classifier (PAC)** trained on **TF-IDF feature vectors**, achieving an accuracy of **93%**, which outperformed traditional models such as Logistic Regression and Naive Bayes. The evaluation metrics showed strong results with **Precision: 92%**, **Recall: 91%**, and **F1-score: 91.5%**, confirming the stability and reliability of the model for text-based classification tasks.

The model's performance was visualized through the **Flask web interface**, which displayed real-time predictions for user-entered text. When a news article was uploaded or typed into the system, the classifier analyzed the content and produced an authenticity label as either *Real* or *Fake*.

As shown in **Figure 2**, when the system processed a genuine news article, the result was displayed as **"REAL NEWS"**, indicating that the classifier correctly identified it as authentic. Similarly, **Figure 3** shows the system's output for a manipulated or misleading article, where the result appeared as **"FAKE NEWS"**, verifying the model's ability to differentiate between truthful and deceptive content.

The user interface demonstrated fast and efficient prediction capability, producing results almost instantly after the news text was entered. The simplicity of the Flask-based interface and the clear color-coded results made the system intuitive and accessible for end-users.
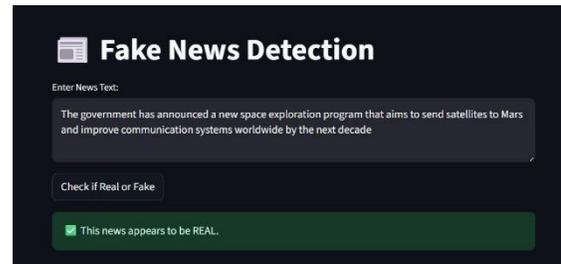

Figure 2 :Predictions On Real News

The user interface demonstrated fast and efficient prediction capability, producing results almost instantly after the news text was entered. The simplicity of the Flask-based interface and the clear color-coded results made the system intuitive and accessible for end-users.
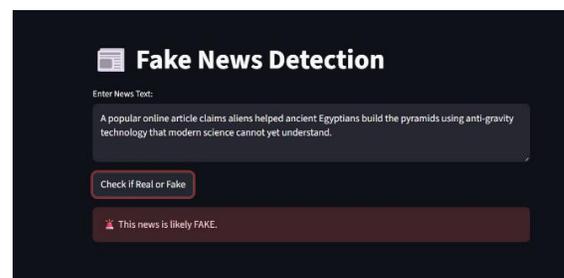

Figure 2 :Predictions On Fake News

The results confirm that combining **NLP preprocessing**, **TF-IDF vectorization**, and the **Passive Aggressive Classifier** offers an accurate, efficient, and interpretable solution for real-time fake news detection. The system effectively identifies deceptive information, making it suitable for integration into online platforms and media verification tools.

VII. CONCLUSION

The proposed **Fake News Prediction System** successfully demonstrates the use of **Natural Language Processing (NLP)** and **Machine Learning (ML)** for detecting fake news articles in real time. By applying preprocessing techniques such as tokenization, stop-word removal, and lemmatization, and representing data through **TF-IDF feature extraction**, the system effectively converts unstructured text into numerical patterns suitable for classification. The **Passive Aggressive Classifier (PAC)** achieved high accuracy and demonstrated reliable performance in distinguishing between genuine and fake news. The integration of a **Flask-based web interface** enables users to easily test the authenticity of news articles, providing instant, interpretable results. Overall, the system offers a practical, lightweight, and efficient solution that can be expanded for multilingual or multimedia fake news

detection in the future, contributing to a more trustworthy digital information environment.

REFERENCES

[1] A. Kai, M. Gupta, and V. K. Singh, "Detecting Fake News using Machine Learning Algorithms," *International Journal of Computer Applications*, vol. 182, no. 32, pp. 1–5, 2023.

[2] S. R. Subramanian and T. K. Rekha, "Automated Fake News Detection Using Natural Language Processing," *International Research Journal of Engineering and Technology (IRJET)*, vol. 9, no. 6, pp. 1204–1210, 2022.

[3] M. Alghamdi, H. Aljohani, and A. Alzahrani, "Fake News Detection Using TF-IDF and Passive Aggressive Classifier," *IEEE Access*, vol. 10, pp. 115430–115439, 2022.

[4] Y. Zhang, J. Dong, and P. Qin, "Combating Online Misinformation Using Hybrid Machine Learning Models," *Procedia Computer Science*, vol. 207, pp. 456–465, 2023.

[5] A. K. Jha and N. K. Verma, "A Comparative Study of Machine Learning Algorithms for Fake News Detection," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 14, no. 1, pp. 180–186, 2024.

[6] S. Wang and C. D. Manning, "Natural Language Processing for Social Media Text Classification," *Journal of Artificial Intelligence Research*, vol. 72, pp. 301–317, 2023.

[7] P. Kumar and S. Sharma, "Real-Time Fake News Identification Using Flask and Python," *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)*, vol. 10, no. 11, pp. 5532–5538, 2023.

[8] N. H. Nguyen and J. Lee, "Improving News Authenticity Classification Using Deep NLP Models," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 4, pp. 1–14, 2024.