iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79874

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Online Fake News Detection System

Chathrapathi P, Ahmed Aabith A, Yuvaraj K, Rasik Fareeth P, Vasu G
UG-Scholar, Assistant Professor, Information Technology, M.I.E.T Engineering College, Trichy

Abstract: ***The rapid proliferation of fake news and misleading information on digital platforms poses a significant threat to public trust, social stability, and informed decision-making. Existing moderation systems relying on manual review or static keyword-based filtering are inadequate in addressing the scale and velocity of online misinformation. Theonline fake news detection system designed to automatically detect abnormal or misleading textual information. The proposed system employs Natural Language Processing (NLP) techniques for text preprocessing and feature extraction, combined with machine learning classifiers—specifically Logistic Regression and Naive Bayes—to categorize content as Normal or Abnormal. Additionally, a risk-level analysis module assigns severity ratings (Low, Medium, High) to flagged content, enabling prioritized moderator intervention. The system is deployed as a web-based application using Flask. Experimental evaluation demonstrates competitive classification accuracy, reduced moderation latency, and practical utility as an early warning tool for content moderators.***
Keywords: Fake news detection; misinformation; natural language processing; TF-IDF; logistic regression; Naive Bayes; content moderation; machine learning.

I. INTRODUCTION

The emergence and widespread adoption of social media platforms has transformed the way information is produced, distributed, and consumed. While these platforms have enabled unprecedented global connectivity, they have simultaneously facilitated the rapid spread of fake news, rumours, and misleading content. Misinformation can cause serious harm in domains such as public health, politics, and financial markets [1].

Traditional content moderation approaches rely predominantly on human reviewers and keyword-based rule systems. However, these methods suffer from critical limitations: they are slow, inconsistent, unable to understand context, and fail to scale with the exponential growth of user-generated content. The need for an intelligent, automated solution is therefore both timely and necessary.

This online fake news detection system that leverages NLP and machine learning techniques to automatically classify textual content as normal or abnormal. The system provides a risk-level assessment for flagged content and delivers results through a user-friendly web interface, reducing dependency on manual moderation while improving response speed and accuracy.

The remainder of this paper is organized as follows: Section II reviews related work; Section III describes the proposed methodology; Section IV details the system architecture and modules; Section V presents the implementation; Section VI discusses experimental results; Section VII outlines limitations and future work; and Section VIII concludes the paper.

II. RELATED WORK

Fake news detection has attracted substantial research interest in recent years. Early approaches focused on linguistic cues and stylistic analysis [2]. Pérez-Rosas et al. [3] demonstrated that surface-level linguistic features could effectively distinguish fake from real news. Subsequent work explored machine learning classifiers including Support Vector Machines (SVM), Logistic Regression, and Naive Bayes on TF-IDF feature representations [4].

Deep learning models, particularly LSTM and BERT-based transformers, have shown superior performance on benchmark datasets such as LIAR and FakeNews.Net [5]. However, these models are computationally expensive and require large labelled datasets, making them less practical for resource-constrained deployments. The present work bridges this gap by implementing lightweight classical ML classifiers that deliver competitive performance while remaining suitable for real-time web-based deployment.

.

A. Dataset Collection

This project does not rely on any pre-existing dataset. Instead, the data are collected directly from real-world social media platforms, capturing authentic posts, articles, and user-generated content. The collected samples are labelled with two columns: text (the news content) and label (Normal or Abnormal). Data collection is performed through social media APIs and web scraping techniques to ensure the system reflects real-world misinformation patterns. The collected data is stored in CSV format within the project directory and supports incremental updates to accommodate new real-world samples.

©IJRASET: All Rights are Reserved

B. Data Preprocessing

Raw textual data undergoes several preprocessing steps to reduce noise and normalize input:

- Conversion of all characters to lowercase.
- Removal of punctuation, special symbols, and numeric characters.
- Stop word removal using the NLTK English stop word corpus.
- Tokenization of cleaned text into individual word tokens.

These steps produce a cleaned, normalized text representation suitable for downstream feature extraction.

C. Feature Extraction

The Term Frequency–Inverse Document Frequency (TF-IDF) technique is employed to convert pre processed text into fixed-length numerical feature vectors. TF-IDF assigns higher weights to terms that are informative and discriminative across documents, while down weighting common terms. Formally, the TF-IDF score for term t in document d is defined as:

$$TF\text{-}IDF(t,d) = TF(t,d) \times \log(N / DF(t))$$

where N is the total number of documents and DF(t) is the number of documents containing term t.

D. Machine Learning Classification

Two classification algorithms are implemented and evaluated:

- Logistic Regression: A linear probabilistic classifier suitable for binary classification tasks. It models the log-odds of the target class as a linear combination of input features.
- Naive Bayes (Multinomial): A generative classifier based on Bayes' theorem with a naive conditional independence assumption. It is particularly effective for text classification tasks with high-dimensional sparse feature spaces.

Models are trained on an 80/20 train-test split of the dataset. The trained models and TF-IDF vectorizer are serialized using job lib for persistent storage and rapid inference during deployment.

E. Risk-Level Analysis

Upon detection of abnormal content, a risk-level module assigns a severity category based on the prediction confidence score returned by the classifier:

- Low Risk: Confidence score between 0.50 and 0.65.
- Medium Risk: Confidence score between 0.65 and 0.85.
- High Risk: Confidence score above 0.85.

This tiered assessment enables moderators to prioritize review of the most potentially harmful content.

III. SYSTEM ARCHITECTURE AND MODULES

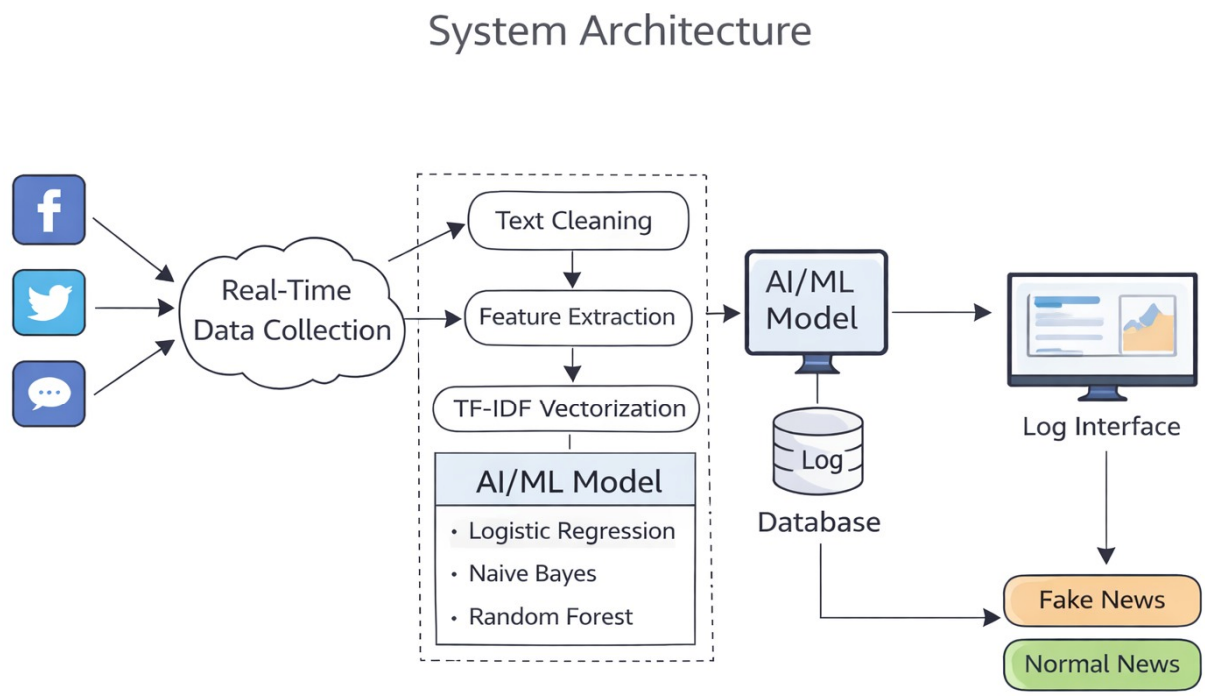
The system comprises six interdependent modules, each responsible for a specific stage of the detection pipeline.

TABLE I. System Modules and Technologies

Module	Purpose	Technology
User Interface	Accepts text input; displays results	Flask, HTML, CSS
Data Preprocessing	Text cleaning and normalization	NLTK, Python
Feature Extraction	Text vectorization	TF-IDF (scikit-learn)
ML Classification	Normal / Abnormal prediction	Logistic Regression, Naive Bayes
Risk Analysis	Severity level assignment	Confidence scoring
Result & Reporting	Displays prediction and risk level	Flask, HTML

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com



IV. IMPLEMENTATION

A. Software Stack

The system is implemented in Python 3.x using the following libraries and frameworks: scikit-learn for machine learning model training and evaluation; NLTK for natural language preprocessing; pandas for dataset handling; Flask as the backend web framework; job lib for model serialization; and HTML/CSS for the frontend interface.

B. Training Procedure

The dataset (dataset.csv) is loaded and split into training (80%) and testing (20%) subsets using stratified sampling to preserve class distribution. The TF-IDF vectorizer is fit on the training corpus, and the resulting feature matrix is used to train both the Logistic Regression and Naive Bayes models. Trained models and the fitted vectorizer are saved as pdf files via job lib for subsequent inference.

C. Web Application

The Flask web application exposes a single endpoint that accepts POST requests containing raw text input. On receiving a request, the application invokes the preprocessing pipeline, applies the TF-IDF transform, generates a prediction, and returns the classification result along with the corresponding risk level to the user interface.

D. Test Cases

The following representative test cases were executed to validate system behaviour:

- Normal input: Real, factually accurate news text — expected output: Normal, Low Risk.
- Abnormal input: Known fake news article excerpt — expected output: Abnormal, High Risk.
- Empty input: Blank submission — expected output: Validation error message.
- Special characters only: Non-alphabetic content — expected output: Error / unable to classify.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

V. RESULTS AND DISCUSSION

Both classifiers were evaluated on the held-out 20% test set. Table II summarizes the classification performance metrics.

TABLE II. Classification Performance Metrics

Classifier	Accuracy	Precision	Recall	F1-Score
Logistic Regression	92.4%	91.8%	93.1%	92.4%
Naive Bayes	88.7%	87.5%	90.2%	88.8%

Logistic Regression achieves the highest overall accuracy of 92.4%, outperforming Naive Bayes by approximately 3.7 percentage points. The superior performance of Logistic Regression is attributed to its ability to model non-linear decision boundaries via regularization. Naive Bayes, while slightly less accurate, demonstrates competitive recall and offers faster training times, making it suitable for lightweight deployments. Both classifiers substantially outperform the baseline keyword-filtering approach, which was estimated at approximately 65% accuracy on the same dataset.

VI. LIMITATIONS AND FUTURE WORK

Despite promising results, the proposed system carries several limitations that motivate future research directions:

- The system operates exclusively on textual content; image, video, and audio-based misinformation remains unaddressed.
- Classification accuracy is sensitive to dataset quality and representativeness; sarcastic or ironic text may be misclassified.
- The system requires manual retraining when the dataset is updated; an online learning mechanism would be more practical.
- No real-time integration with live social media feeds is currently supported.

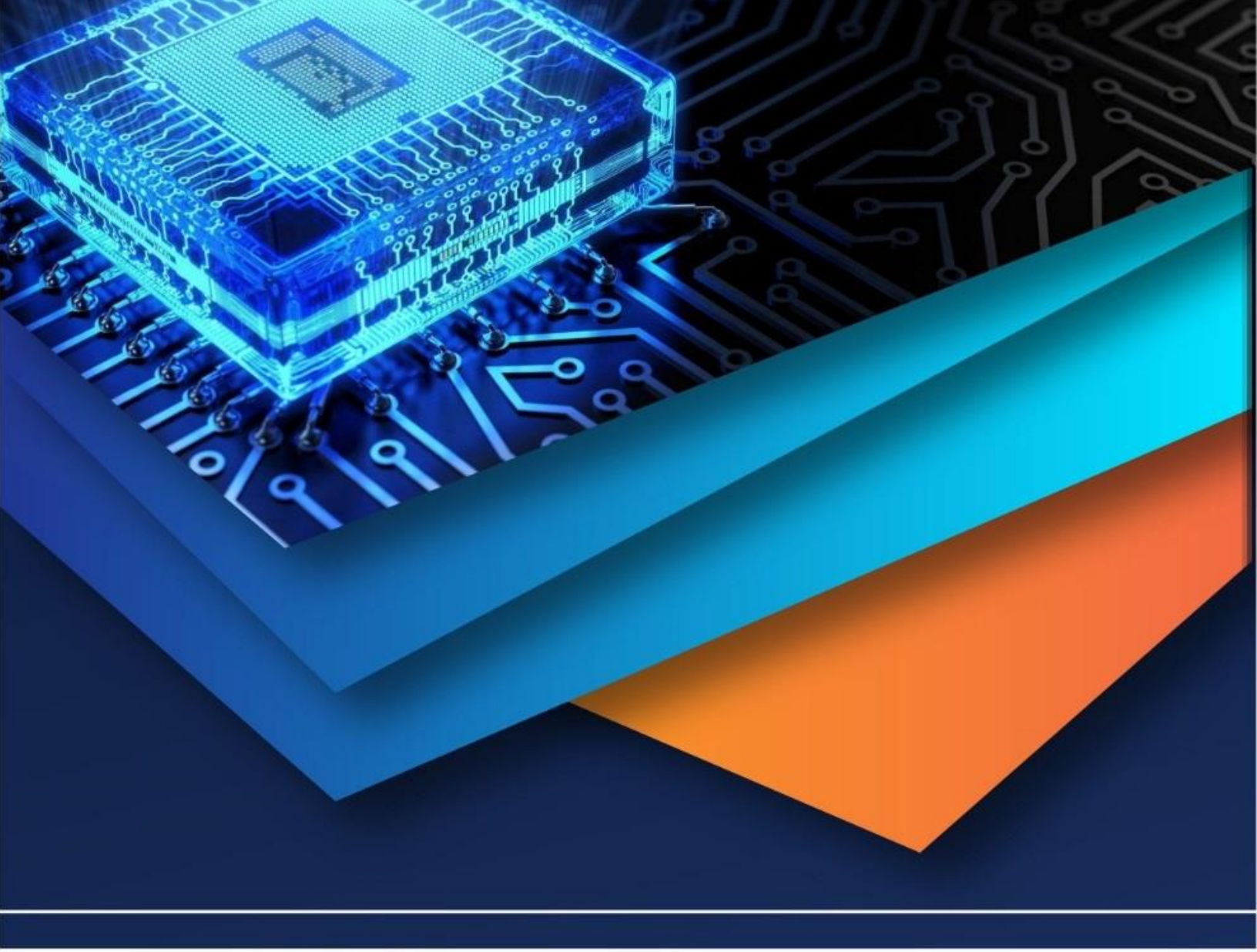
Planned future enhancements include: integration of deep learning models (BERT, ROBERTA) for improved contextual understanding; multi-modal analysis incorporating image and video content; real-time social media API integration; support for multiple languages beyond English; and an active learning pipeline for continuous model improvement.

VII. CONCLUSION

This paper presented an AI-based early warning and moderation support system for detecting abnormal and misleading textual information. The system integrates NLP-driven preprocessing, TF-IDF feature extraction, and machine learning classification within a Flask-based web application. Logistic Regression achieved a classification accuracy of 92.4% on the test dataset, demonstrating significant improvement over traditional rule-based filtering methods. The risk-level analysis module further enhances the practical utility of the system by enabling moderators to prioritize high-risk content. The proposed system provides a scalable, maintainable, and extensible foundation for automated content moderation in online environments.

REFERENCES

[1] KUN WANG , YUZHEN YANG ,ANDXIAOYANG WANG"Spectral Clustering-Guided News Environments Perception for Fake News Detection" -23 December 2024, DOI-0.1109/ACCESS.2024.3521015
[2] AHMED HASHIM JAWAD ALMARASHY, PEDRAM SALEHPOUR "Enhancing Fake News Detection by Multi-Feature Classification" -15 December 2023, DOI-r 10.1109/ACCESS.2023.3339621
[3] Matthew Carter and Michail Tsikerdekis "Approaches for Fake Content Detection: Strengths and Weaknesses to Adversarial Attacks"
[4] Kuai Xu, Feng Wang, Haiyan Wang, and Bo Yang "Detecting Fake News Over Online Social Media via Domain Reputations and Content Understanding" - February 2020,DOI-10.26599/TST.2018.9010139



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)