

# Fake News Detection Using Machine Learning and Natural Language Processing

<sup>1</sup>Prashant Banjare, <sup>2</sup>Pawan Kumar Jaiswal

<sup>1</sup>Student, <sup>2</sup>Assistant Professor

<sup>1,2</sup>AMITY UNIVERSITY, CHHATTISGARH

<sup>1</sup>Prashantbanjare1939gmail.com, <sup>2</sup>pkumar@rpr.amity.edu

## Abstract

The rapid growth of digital communication platforms has dramatically transformed the global flow of information. Social media, online journalism, and independent publishing platforms have enabled information to spread instantly across geographical boundaries. However, the same technological advancements have also accelerated the spread of misinformation and fabricated news content, commonly referred to as fake news. Fake news poses a serious threat to public trust, democratic institutions, economic systems, and social stability. Traditional manual fact-checking systems are insufficient to handle the enormous volume of content generated daily across digital platforms.

This research presents a full-stack Fake News Detection System utilizing Natural Language Processing (NLP) and Machine Learning (ML) techniques. The proposed system combines TF-IDF feature extraction with Logistic Regression classification to identify deceptive textual structures in online news articles. The architecture follows the Model-View-Controller (MVC) paradigm and integrates a Flask-based backend, SQLite database, and responsive frontend developed using HTML, CSS, and JavaScript.

The model was trained on a hybrid dataset exceeding 7,000 rows consisting of global political news, synthetic Indian contexts, and real-world news datasets. Experimental evaluation demonstrated strong classification performance with 91.24% overall accuracy, high precision, and strong recall metrics. Furthermore, the application integrates live RSS news feeds from BBC and CNN APIs, enabling real-time evaluation of dynamic internet news streams. This research also investigates Out-of-Distribution vulnerabilities and discusses future advancements using semantic transformer architectures such as BERT and RoBERTa.

## Keywords

Fake News Detection, Machine Learning, Natural Language Processing, Logistic Regression, TF-IDF, Flask, SQLite, NLP, Misinformation Detection

## 1. Introduction

### 1.1 Defining Fake News

Fake news refers to fabricated or misleading information intentionally designed to imitate authentic journalism. Unlike accidental misinformation, fake news is deliberately engineered to manipulate public perception, generate emotional reactions, or spread political propaganda. In

the modern digital ecosystem, fake news spreads rapidly through social media platforms, blogs, online forums, and video-sharing websites.

### 1.2 Growth of Digital Misinformation

The rise of social media has transformed ordinary users into active publishers of information. Platforms such as Facebook, X (Twitter), Instagram, Reddit, and YouTube enable content to spread globally within seconds. Unfortunately, this rapid dissemination makes misinformation extremely difficult to control. During major events such as elections, pandemics, and geopolitical crises, fake news often spreads faster than verified journalism.

### 1.3 Importance of Automated Detection

Manual fact-checking systems require significant human effort and cannot keep pace with the enormous amount of digital information generated every second. Automated machine learning systems therefore become essential for detecting suspicious textual patterns, sensationalized narratives, and deceptive language structures in real time.

### 1.4 Research Objectives

The primary objectives of this research are:

- To design a machine learning pipeline capable of detecting fake news.
- To implement a deployable full-stack web application.
- To integrate live news APIs for real-time testing.
- To evaluate Out-of-Distribution vulnerabilities.
- To maintain prediction history using relational databases.

### 1.5 Scope of the Research

The scope of this project includes supervised machine learning classification of textual news data using statistical NLP techniques. The research focuses primarily on political and socio-economic news domains while acknowledging limitations when processing highly scientific or technical content.

## 2. Literature Review

The field of fake news detection has evolved significantly over the past decade. Early approaches relied heavily on rule-based filtering systems and manually engineered keyword detection algorithms. These systems lacked scalability and adaptability because they could not generalize to unseen narratives or evolving misinformation patterns.

### 2.1 Statistical NLP Approaches

The introduction of statistical NLP techniques transformed textual classification systems. Bag-of-Words models represented text numerically by counting word frequencies; however, these approaches suffered from sparse feature spaces and poor contextual understanding.

TF-IDF emerged as a more effective alternative by assigning higher importance to rare but semantically meaningful words while suppressing generic filler terms.

## 2.2 Machine Learning Algorithms

Researchers experimented with multiple classification algorithms including:

- Naive Bayes
- Support Vector Machines
- Decision Trees
- Random Forests
- Logistic Regression

Among these, Logistic Regression gained popularity due to its interpretability, computational efficiency, and strong binary classification performance.

## 2.3 Deep Learning Advancements

Modern research increasingly focuses on transformer-based architectures such as:

- BERT
- RoBERTa
- GPT
- XLNet

These models utilize attention mechanisms and contextual embeddings to understand semantic relationships between words. Although highly accurate, transformer systems require large datasets and expensive computational infrastructure.

## 2.4 Research Gap

Despite significant advancements, many fake news systems remain restricted to isolated notebook environments without real-world deployment capability. This research addresses the gap by integrating machine learning with scalable web technologies and real-time API processing.

## 3. System Architecture

The proposed Fake News Detection System follows the Model-View-Controller (MVC) architecture, separating presentation, logic, and data layers to improve scalability and maintainability.

### 3.1 Frontend Layer

The frontend was developed using:

- HTML5
- CSS3
- JavaScript

The interface employs Glassmorphism design principles and asynchronous Fetch APIs to ensure smooth user interaction without full-page reloads.

### 3.2 Backend Layer

The backend utilizes Flask, a lightweight Python micro-framework, responsible for:

- Managing RESTful APIs
- Processing prediction requests
- Handling database communication
- Fetching live RSS feeds
- Managing asynchronous operations

### 3.3 Machine Learning Layer

The ML layer consists of:

- TF-IDF Vectorizer
- Logistic Regression Model
- Serialized Joblib resources

### 3.4 Database Layer

SQLite was selected due to its lightweight architecture and ease of integration. The database stores:

- User queries
- Predictions
- Confidence scores
- Timestamps

This enables historical auditing and performance tracking.

## 4. Methodology

### 4.1 Data Collection

The dataset combines open-source fake news datasets with synthetic Indian and global political contexts. The hybrid approach improves vocabulary diversity and regional robustness.

### 4.2 Text Preprocessing

The preprocessing pipeline includes:

- Lowercasing
- URL removal
- Regex sanitization
- Stop-word elimination
- Punctuation stripping
- Whitespace normalization

#### 4.3 Feature Extraction

Machine learning models cannot process raw strings directly. TF-IDF converts textual content into weighted numerical vectors representing term importance.

#### 4.4 Classification Model

The Logistic Regression classifier computes the probability that an article belongs to the FAKE or REAL category. Logistic Regression was selected because:

- It is computationally lightweight.
- It provides high interpretability.
- It performs well for binary classification.
- It supports rapid inference in web environments.

#### 4.5 Real-Time RSS Integration

The application integrates RSS feeds from BBC and CNN APIs. This feature allows the system to analyze live internet news streams dynamically.

### **5. Dataset Engineering and Training**

Dataset quality directly impacts machine learning performance. To improve generalization, this research implemented a hybrid data synthesis strategy.

#### 5.1 Synthetic Data Generation

Synthetic Indian and global political contexts were generated programmatically using:

- Politician names
- City names
- Political issues
- Geopolitical entities

#### 5.2 Dataset Composition

The final dataset exceeded 7,000 rows and included:

- Real news articles
- Fake news narratives
- Indian political contexts
- Global political contexts

### 5.3 Training Pipeline

The training process involved:

- Train-test splitting
- TF-IDF vectorization
- Logistic Regression training
- Model serialization using Joblib

Serialized resources:

- model.pkl
- vectorizer.pkl

This serialization mechanism allows the Flask application to load pre-trained models directly into active memory.

## 6. Implementation

Implementation bridges the gap between theoretical machine learning and deployable software engineering.

### 6.1 Backend Development

Flask APIs were implemented for:

- Prediction requests
- History retrieval
- RSS feed processing

### 6.2 Frontend Realization

The frontend dynamically communicates with the backend using asynchronous Fetch APIs. JavaScript updates the interface in real time without requiring page reloads.

### 6.3 Database Integration

SQLite was integrated through Flask-SQLAlchemy ORM. Every prediction is stored securely with:

- Text snippets
- Predictions
- Confidence scores
- Timestamps

#### 6.4 User Experience Design

The application follows a modern Glassmorphism design system featuring:

- Dynamic cards
- Responsive layouts
- Loading animations
- Smooth transitions

These features improve user engagement and accessibility.

### **7. Experimental Results and Evaluation**

The proposed system was rigorously evaluated using unseen testing datasets.

#### 7.1 Performance Metrics

The model achieved:

- Accuracy: 91.24%
- Precision: 89%
- Recall: 93%

The strong recall value demonstrates the system's effectiveness in identifying deceptive content while minimizing false negatives.

#### 7.2 Integration Testing

Integration testing confirmed successful communication between:

- Frontend interface
- Flask backend
- SQLite database

#### 7.3 Real-Time Testing

Live RSS feed integration allowed the model to process unpredictable real-world news streams.

#### 7.4 Out-of-Distribution Analysis

A critical contribution of this research involves analyzing model failures under scientific and technical domains. Because TF-IDF relies heavily on known vocabulary, unseen terminology frequently resulted in near-zero feature vectors, causing inaccurate predictions.

## 8. Advantages and Limitations

### 8.1 Advantages

The proposed architecture offers:

- Lightweight deployment
- Fast prediction speed
- High interpretability
- Low operational cost
- Real-time API integration
- Persistent logging
- Scalable architecture

### 8.2 Limitations

Despite strong performance, the system exhibits:

- Vocabulary dependency
- Reduced semantic understanding
- OOD vulnerability
- Limited contextual reasoning
- Binary classification restrictions

These limitations motivate future transformer-based improvements.

## 9. Future Scope

Future enhancements can significantly improve the system's capabilities.

Potential future developments include:

- Transformer integration using BERT and RoBERTa
- Multilingual fake news detection
- Social media API integration
- Docker and Kubernetes deployment
- PostgreSQL database migration
- Explainable AI visualizations
- Mobile application development
- User authentication systems

Future semantic transformer models will improve contextual understanding and cross-domain adaptability.

## 10. Conclusion

This research successfully demonstrates the practical integration of Natural Language Processing and Machine Learning within a deployable full-stack web application. The combination of TF-IDF feature extraction and Logistic Regression classification achieved strong performance while maintaining computational efficiency and interpretability. The integration of Flask APIs, SQLite persistence, and real-time RSS feed processing transformed the system into a practical misinformation analysis platform. Furthermore, the study highlighted critical vulnerabilities associated with statistical NLP systems under Out-of-Distribution conditions. These findings emphasize the importance of future contextual semantic architectures.

Overall, the project proves that lightweight machine learning systems remain highly effective for scalable fake news detection applications and provide a strong foundation for future transformer-based advancements.

## References

- [1] L. Hamel, "Fake News Dataset," GitHub Repository, 2020.
- [2] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, 2011.
- [3] Flask Documentation, Pallets Projects, 2024.
- [4] Mozilla Developer Network, "Fetch API Documentation," 2025.
- [5] M. Grinberg et al., "Fake News on Twitter During the 2016 U.S. Presidential Election," Science, 2019.
- [6] Python Software Foundation, "Python Documentation," 2025.
- [7] Scikit-Learn Official Documentation, 2025.
- [8] TensorFlow Research Papers on NLP and Text Classification, 2024.
- [9] Research Papers on Transformer Architectures and NLP Systems, 2025.