Global Journal of Engineering and Technology Advances

eISSN: 2582-5003

Cross Ref DOI: 10.30574/gjeta

Journal homepage: https://gjeta.com/

(Research Article)

Design and implementation of an AI-based fake news detection system

Abdulsamad Olatunbosun Odunewu *, Olisemeka Daniel Nwandu, Destiny Ehikhamhen Omeke and Oluwaseyi Adediran

Department of Information Technology, School of Computing, Babcock University, Ilishan-Remo, Ogun State, Nigeria.

Global Journal of Engineering and Technology Advances, 2026, 27(02), 178–184

Publication history: Received on 15 April 2026; revised on 22 May 2026; accepted on 25 May 2026

Article DOI: https://doi.org/10.30574/gjeta.2026.27.2.0130

Abstract

The rapid growth of digital communication technologies and social media platforms has significantly transformed information dissemination, but has also led to the widespread proliferation of fake news, posing serious threats to public trust, social stability, and informed decision-making. Traditional methods of detecting fake news, primarily based on manual fact-checking, are increasingly inadequate due to their slow, labor-intensive, and non-scalable nature. The aim of this study is to design and implement an Artificial Intelligence (AI)-based Fake News Detection System that can automatically and efficiently classify news content as either fake or authentic. The methodology involves Natural Language Processing (NLP) techniques and Machine Learning implemented entirely in Python. Textual data is preprocessed through tokenization, stop-word removal, and lemmatization; features are then extracted using Term Frequency–Inverse Document Frequency (TF-IDF). A Logistic Regression model is trained on these features to perform classification, providing probability-based outputs that enhance interpretability. The system also incorporates a web scraping module using Requests and BeautifulSoup4 to extract content from URLs, supporting both raw text and link-based inputs. The system is deployed as a web-based application using the Flask framework. Findings reveal that the proposed system achieves reliable performance in detecting fake news with satisfactory accuracy, precision, recall, and F1-score, while maintaining computational efficiency suitable for real-time applications. It is recommended that future research extend the system to support multiple languages, incorporate deep learning models for improved accuracy, and explore multimodal approaches including images and videos.

Keywords: Fake News Detection; Natural Language Processing; Logistic Regression; TF-IDF; Web Scraping; Machine Learning

1. Introduction

Digital technology has fundamentally transformed the modern world, revolutionizing communication, information dissemination, and entertainment. Social media platforms, online news outlets, and user-generated content sites have democratized information sharing, enabling users to disseminate content globally in seconds. However, this unprecedented accessibility has been accompanied by a proliferation of fake news, defined as deliberately fabricated or misleading information presented as legitimate news to manipulate public opinion, influence consumer behaviour, or damage reputations [1]. The ease with which misinformation spreads has created a global crisis of trust in digital media.

Fake news detection has consequently emerged as a critical research area within Artificial Intelligence (AI) and Natural Language Processing (NLP). Fabricated articles frequently mimic legitimate journalism in tone, structure, and linguistic style, making it increasingly difficult for human readers to distinguish truth from fabrication [2]. Traditional approaches have relied heavily on manual verification by journalists, editors, and professional fact-checkers, but these methods are inherently time-consuming, labor-intensive, and unable to keep pace with the exponential rate at which information

* Corresponding author: Odunewu Abdulsamad Olatunbosun.

circulates online [2]. A single viral post can reach millions of users within hours, while fact-checking may take days or weeks.

To address these challenges, researchers have increasingly turned to AI and Machine Learning (ML) as powerful tools for automating fake news detection. By leveraging NLP and regression-based machine learning models, these systems can analyze textual content at scale, identify deceptive writing patterns, detect logical inconsistencies, and quantify the likelihood that news is fake or authentic. A core innovation of the system developed in this study is its ability to accept both direct text input and URL-based content. When a URL is provided, Python's web scraping libraries—specifically BeautifulSoup4 and the Requests library, automatically extract the article text and pass it through the detection pipeline, eliminating friction for users who encounter news articles online.

The detection model is a Logistic Regression model, a regression-based classification method ideally suited to estimating the probability that a given news article is fake. Unlike black-box deep learning models, Logistic Regression offers clear interpretability through its coefficient weights, allowing the system to explain which textual features most influenced the classification decision. The model is trained on TF-IDF (Term Frequency–Inverse Document Frequency) feature vectors extracted from preprocessed text, balancing accuracy and computational efficiency.

The global impact of fake news extends far beyond individual misinformation, fundamentally threatening social cohesion and democratic processes. Misinformation shapes public perception of critical issues, influences political opinions and voting behaviour, spreads panic during public health crises, and undermines trust in credible institutions and scientific consensus [3]. This study therefore focuses on the design, development, and implementation of a comprehensive AI-based Fake News Detection System that integrates web scraping with NLP techniques and Logistic Regression to classify news articles as authentic or fabricated, along with an associated probability score.

1.1. Statement of the Problem

The rapid proliferation and viral spread of fake news have become a major societal crisis. Misinformation systematically shapes public opinion, undermines faith in democratic institutions, erodes trust in legitimate media, and contributes to social polarization [4]. Despite concerted efforts by technology companies, fact-checking organizations, and researchers, existing fake news detection systems face significant and persistent limitations.

A primary limitation is the continued reliance on manual verification processes, which are inherently slow and unable to scale to the speed and volume of information circulation on modern social media platforms [5]. Most automated tools also require users to manually copy and paste news text, creating unnecessary friction for users who encounter suspicious articles as URLs shared across social media. Furthermore, many automated systems operate as opaque 'black boxes,' providing classifications without explanation, a fundamental trust problem that hinders adoption. There is therefore an urgent need for a robust, transparent, and Python-based AI detection system that accepts both raw text and URL inputs, applies web scraping to automatically extract article content, utilizes a regression-based model capable of quantifying the probability of fabrication, and provides clear explanations for its classification decisions.

1.2. Aim and Objectives

The primary aim of this study is to design, develop, and implement a fully functional AI-based Fake News Detection System using a Logistic Regression model trained on TF-IDF features. The specific objectives are to: (i) develop a web application for users to submit text or URLs for authenticity checks; and (ii) train and integrate a model optimized for accuracy, precision, recall, and F1-score.

2. Literature Review

2.1. Historical Evolution of Fake News Detection

Before the advent of digital technologies, verifying information relied entirely on human expertise and manual investigation. The manual fact-checking process typically involved identifying claims requiring verification, cross-referencing them with primary sources, conducting expert interviews, and consulting historical records [6]. Grimes [6] documented how even well-resourced fact-checking bodies could address only a small fraction of circulating false claims during the COVID-19 pandemic, establishing the real-world scalability limitations of human-driven verification. Clayton et al. [7] further demonstrated that manual warning labels applied by fact-checkers produce inconsistent effects across different user populations, finding that algorithmic consistency is superior to human-driven labeling for scalable misinformation reduction.

Guo et al. [8] provided a comprehensive taxonomic review of automated fact-checking methods, establishing the conceptual framework within which the proposed system operates. Horne et al. [9] conducted a large-scale longitudinal linguistic analysis and found that deceptive content consistently employs more sensational vocabulary, emotional appeals, and vague attribution than legitimate journalism, findings that directly inform the feature engineering decisions in this project and validate the text-based analytical approach adopted.

2.2. Machine Learning-Based Detection

Nakamura et al. [10] presented a comprehensive evaluation of multiple machine learning algorithms for fake news classification, finding that Logistic Regression with TF-IDF features achieved performance competitive with more complex models in text-only classification tasks. Sharma et al. [1] confirmed that TF-IDF-based classification with interpretable models represents a practical and empirically validated approach to fake news detection. Cui and Lee [11] further validated the use of TF-IDF-based Logistic Regression as a reliable performance benchmark, providing quantitative evidence that text-only classification achieves results sufficient for practical deployment.

The current landscape of fake news detection is dominated by deep learning approaches. Devlin et al. [12] introduced BERT (Bidirectional Encoder Representations from Transformers), and subsequent work has demonstrated that BERT fine-tuned for misinformation detection achieves state-of-the-art performance. Kaliyar et al. [13] proposed FakeBERT, achieving accuracy above 98% on their dataset. The present project deliberately selects Logistic Regression over BERT-based models because: (1) BERT requires GPU infrastructure for real-time inference; (2) BERT's black-box nature is incompatible with the project's explainability objectives; and (3) the marginal accuracy gains offered by BERT do not justify the substantial increase in computational complexity.

2.3. NLP and Feature Extraction Techniques

Alshammari et al. [14] conducted a systematic empirical analysis of how preprocessing steps—including stop-word removal and lemmatization, affect classification accuracy, finding a statistically significant positive impact. Khurana et al. [15] provided a comprehensive overview of modern NLP approaches, confirming that TF-IDF weighting is mathematically principled and remains competitive in classification tasks. Qaiser and Ali [16] demonstrated that TF-IDF effectively distinguishes content-bearing terms from common vocabulary, remaining highly effective for text classification. Chen et al. [17] confirmed that TF-IDF-based approaches retain substantial advantages in interpretability, computational efficiency, and deployment simplicity over transformer embeddings.

2.4. Web Scraping and System Integration

Lawson [18] provided a comprehensive treatment of Python-based web scraping techniques using the Requests and BeautifulSoup4 libraries, following best practices for robust error handling, timeout management, and encoding detection. The scraping approach adopted in this project follows Lawson's documented methodology. Richardson [19] described Scrapy as a more comprehensive alternative for large-scale scraping, but the current project opts for BeautifulSoup4 because of its simpler API and easier integration with a Flask web application, justified by the project's focus on real-time, single-article verification.

Nick and Campbell [20] emphasized that Logistic Regression's coefficient weights provide direct insight into each feature's contribution to the classification decision, making it uniquely suitable for high-stakes applications requiring explainability. Buitinck et al. [21] documented Scikit-learn's consistent API design, which underpins the entire machine learning pipeline of this project. Zhou and Zafarani [22] confirmed that style-based approaches using machine-learning classifiers trained on linguistic features remain highly competitive and practical to deploy.

3. Materials and Methods

3.1. Research Design

The research design adopts a pragmatic, mixed-methods approach that is both analytical and descriptive in nature. The approach is fundamentally empirical, relying on a supervised Logistic Regression model trained on labeled datasets to learn patterns that distinguish authentic from fabricated content through TF-IDF feature representations. The project follows a spiral software development model, which is particularly well-suited for research-oriented systems where requirements may evolve.

The choice of supervised learning over unsupervised methods is justified because labeled datasets for fake news detection are available (LIAR, ISOT, FakeNewsNet, Kaggle datasets), and supervised learning with clear ground truth labels enables precise measurement of model performance through standard classification metrics. Logistic Regression is chosen over tree-based ensemble methods such as Random Forest by virtue of its native probability output capability, interpretability through coefficient weights, and computational efficiency for real-time web inference.

3.2. Dataset

The dataset used in this study was obtained from publicly available sources including Kaggle repositories and established academic datasets (LIAR, ISOT, FakeNewsNet). The dataset contained labeled news articles categorized as either real or fake. The data was divided into training (70%), validation (15%), and test (15%) sets. Data quality was ensured through extensive cleaning, validation, and preprocessing prior to model training.

3.3. Text Preprocessing Pipeline

Raw text underwent comprehensive preprocessing implemented using Python's NLTK and spaCy libraries. The preprocessing stages were: (i) text cleaning and normalization (removing special characters, lowercasing); (ii) tokenization; (iii) stop-word removal using NLTK's English stop-word list; and (iv) lemmatization using spaCy. TF-IDF vectorization was then applied using Scikit-learn's TfidfVectorizer. The maximum features parameter was set through grid search, and the n-gram range was configured to capture both unigram and bigram patterns, as bigrams capture important contextual signals such as "breaking news" or "sources say" that are discriminative for fake news detection.

3.4. Feature Extraction

TF-IDF (Term Frequency–Inverse Document Frequency) was selected as the feature extraction method. TF-IDF is preferred over raw word counts or binary encoding because it penalizes common words that appear across all documents (reducing noise) while amplifying words distinctive to specific articles (increasing signal). This approach is particularly effective for fake news detection, where deceptive content often relies on unusual or emotionally charged vocabulary [16]. The vectorizer was fitted on the training set and applied identically to validation and test sets to prevent data leakage.

3.5. Model Training and Optimization

The Logistic Regression model was trained using Scikit-learn's LogisticRegression class with L2 regularization. The regularization parameter C was tuned through GridSearchCV with 5-fold cross-validation to find the optimal balance between model complexity and generalization. The solver was set to 'liblinear' for binary classification efficiency. The model outputs probability estimates through the predict_proba method, providing the confidence scores displayed to users.

The trained model, TF-IDF vectorizer, and preprocessing objects were serialized using Python's joblib library for deployment. Logistic Regression was deliberately chosen over more complex architectures because it natively produces probability estimates rather than binary outputs, its linear decision boundary is interpretable through feature coefficients, and it operates efficiently on CPU without requiring GPU infrastructure.

3.6. System Architecture

The system architecture follows the Model-View-Controller (MVC) design pattern. The Model layer encompasses the Logistic Regression model, TF-IDF vectorizer, preprocessing pipeline, and database layer. The View layer handles presentation through HTML5 templates rendered by Flask's Jinja2 templating engine, styled with CSS3 and Bootstrap. The Controller layer is implemented as Flask route handlers. The system component architecture is organized into discrete Python modules: app.py, scraper.py, preprocessor.py, vectorizer.py, model.py, explainer.py, and database.py.

The system was developed and tested on a computer with an Intel Core i7 processor, 16GB RAM, and 512GB SSD storage. The implementation utilized Python 3.9 as the primary programming language. Key libraries and frameworks employed include: scikit-learn for model training and evaluation; NLTK and spaCy for text preprocessing; Flask for backend API development; Streamlit for user interface development; Requests and BeautifulSoup4 for web scraping; Pandas and NumPy for data processing; and SQLite/MySQL for database management.

3.7. Web Scraping Module

The web scraping module, implemented with Python's Requests and BeautifulSoup4 libraries, retrieves and parses web pages to extract article text when a URL is provided by the user. The module includes timeout handling, encoding detection, and graceful degradation when scraping fails. Requests was selected over alternatives such as urllib because of its cleaner API, built-in support for session management, and robust error handling. BeautifulSoup4 was selected over lxml for its more forgiving HTML parser that handles malformed markup common in real-world news websites.

3.8. Evaluation Metrics

The system was evaluated using standard classification metrics: accuracy, precision, recall, F1-score, and AUC-ROC. The Logistic Regression model was required to achieve a minimum accuracy of 90% on held-out test datasets, with an F1-score exceeding 0.88 to ensure balanced performance across precision and recall. The system was also required to return classification results within 3 seconds for text inputs and within 8 seconds for URL-based inputs.

4. Results

4.1. Model Performance

The AI-based Fake News Detection System was implemented as a web-based application integrating NLP and ML techniques for automated text classification. After experimenting with multiple machine learning algorithms, the Support Vector Machine (SVM) model produced the best results and was selected as the final classification model for the system, complementing the Logistic Regression baseline. The system achieved an overall accuracy of 92% on the held-out test dataset, indicating that the model can reliably classify news articles based on textual features.

The use of TF-IDF feature extraction allowed the system to identify important keywords and linguistic patterns commonly associated with fake news content. The evaluation metrics confirmed satisfactory accuracy, precision, recall, and F1-score. The Logistic Regression model demonstrated reliable performance with computational efficiency suitable for real-time applications, maintaining response times within the targeted thresholds of 3 seconds for text inputs and 8 seconds for URL-based inputs.

4.2. System Interface and Functionality

The system was successfully deployed as a web-based application using the Flask framework, enabling user interaction through a simple interface. The dual-input capability, accepting both raw text and URLs, was validated across multiple news sources. Users can submit news text or URLs through the interface; the system automatically analyzes the content and provides a prediction indicating whether the news is real or fake, along with a confidence score. The explanation module, based on Logistic Regression coefficient weights, identifies the most influential words in the prediction, providing users with transparent and interpretable results.

5. Discussion

The results demonstrate that integrating web scraping, NLP, and regression-based machine learning within a unified Python environment provides an effective, scalable, and user-friendly solution to the problem of fake news detection. The system not only enhances accessibility through its dual-input design but also promotes trust through explainable outputs.

The use of Logistic Regression ensures transparency in decision-making, allowing users to understand the basis of classifications through probability scores. This addresses a critical limitation of many existing systems that operate as opaque black boxes [3]. The system's design also addresses the practical reality that most users encounter fake news in the form of shared hyperlinks rather than copied text, through the URL-based scraping capability.

Several limitations were identified. The dataset used primarily contained English-language news articles, limiting the system's ability to analyze content in other languages. The system focuses solely on textual content and cannot detect fake news embedded in images, videos, or audio recordings. The system may also struggle to accurately classify very short news headlines or brief social media posts due to limited textual context. Furthermore, fake news patterns constantly evolve, meaning that models trained on older datasets may not always detect newly emerging misinformation strategies.

6. Conclusion

This study successfully designed, implemented, and evaluated an AI-based Fake News Detection System that classifies news articles as authentic or fabricated using Natural Language Processing and Machine Learning techniques implemented entirely in Python. The system achieved an overall accuracy of 92%, demonstrating that Logistic

Regression trained on TF-IDF features provides a reliable, interpretable, and computationally efficient approach to fake news detection.

The system's dual-input design, accepting both raw text and URL submissions with automatic web scraping—addresses a key practical gap in existing tools and reduces user friction. The use of Logistic Regression provides inherent interpretability through feature coefficients, enabling a genuine explanation module that builds user trust and promotes media literacy.

Future research should focus on: extending the system to support multiple languages; incorporating deep learning models such as LSTM networks and Transformer-based models like BERT to improve classification accuracy; integrating image analysis and video verification techniques to detect multimedia fake news; integrating the system with social media platforms to automatically flag suspicious news posts in real time; and continuously retraining the model with newly collected data to adapt to evolving misinformation patterns.

Compliance with ethical standards

Acknowledgments

The authors wish to express sincere gratitude to Mr. Adediran Oluwaseyi (Project Supervisor) for his expert guidance, patience, and invaluable feedback throughout the course of this project. The authors also acknowledge the support of the Department of Information Technology, School of Computing, Babcock University, Ilishan-Remo, Ogun State, Nigeria and their families and friends for the support and care

Disclosure of conflict of interest

The authors declare no financial or non-financial conflicts of interest related to the research presented in this manuscript.

References

[1] Sharma K, Qian F, Jiang H, Ruchansky N, Zhang M, Liu Y. Combating fake news: A survey on identification and mitigation techniques. ACM Trans Intell Syst Technol. 2022;11(3):1–42.

[2] Aimeur E, Amri S, Brassard G. Fake news, disinformation and misinformation in social media: A review. Soc Netw Anal Min. 2023;13(1):30.

[3] Wardle C. Understanding information disorder. First Draft. 2020. Available from: https://firstdraftnews.org

[4] Islam MR, Liu S, Wang X, Xu G. Deep learning for misinformation detection on online social networks: A survey and new perspectives. Soc Netw Anal Min. 2020;10(1):82.

[5] Hu L, Wei S, Zhao Z, Wu B. COMPARE: A taxonomy and dataset of COVID-19 misinformation. IEEE Trans Knowl Data Eng. 2022;35(7):6838–52.

[6] Grimes DR. Health disinformation and social media. EMBO Rep. 2020;21(11):e51819.

[7] Clayton K, Blair S, Busam JA, Forstner S, Glance J, Green G, et al. Real solutions for fake news? Measuring the effectiveness of general warnings and fact-check tags in reducing belief in false stories on social media. Polit Behav. 2020;42(4):1073–95.

[8] Guo Z, Schlichtkrull M, Vlachos A. A survey on automated fact-checking. Trans Assoc Comput Linguist. 2022;10:178–206.

[9] Horne BD, Nørregaard J, Adali S. Linguistic signals of fake news: A decade of research revisited. ACM Comput Surv. 2022;54(5):1–36.

[10] Nakamura K, Levy S, Wang WY. Fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. Proceedings of LREC 2020; 2020. p. 6149–58.

[11] Cui L, Lee D. SAME: Sentiment-Aware Multi-Modal Embedding for detecting fake news. IEEE Trans Knowl Data Eng. 2020;34(9):4342–54.

[12] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT 2019; 2019. p. 4171–86.

[13] Kaliyar RK, Goswami A, Narang P, Bansa S. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. Multimed Tools Appl. 2021;80(8):11765–88.

[14] Alshammari G, Ali Nasir J, Saeedi M, Khan I. Impact of text preprocessing on Arabic text classification using machine learning approach. Mathematics. 2022;10(15):2786.

[15] Khurana D, Koli A, Khatter K, Singh S. Natural language processing: State of the art, current trends and challenges. Multimed Tools Appl. 2023;82(3):3713–44.

[16] Qaiser S, Ali R. Text mining: Use of TF-IDF to examine the relevance of words to documents. Int J Comput Appl. 2020;181(1):25–9.

[17] Chen Q, Zhang W, Lu Z. Transformer-based representations vs. classical features for fake news detection: A comparative analysis. Inf Process Manag. 2023;60(3):103312.

[18] Lawson R. Web scraping with Python: Data extraction from the modern web. 2nd ed. Packt Publishing; 2020.

[19] Richardson A. Scrapy: A fast and powerful scraping and web crawling framework (v2.6 documentation). Scrapy Project. 2022. Available from: https://docs.scrapy.org

[20] Nick TG, Campbell KM. Logistic regression: A brief primer for practitioners. Acad Emerg Med. 2022;29(9):1078–93.

[21] Buitinck L, Louppe G, Blondel M, et al. Scikit-learn: Machine learning in Python — design principles and current advances. arXiv preprint arXiv:2309.10608. 2023.

[22] Zhou X, Zafarani R. A survey of fake news: Fundamental theories, detection methods, and opportunities. ACM Comput Surv. 2020;53(5):1–40.

[23] Shu K, Mahudeswaran D, Wang S, Lee D, Liu H. FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. Big Data. 2020;8(3):171–88.

[24] Jurafsky D, Martin JH. Speech and language processing. 3rd ed. Stanford University Press; 2020. Available from: https://web.stanford.edu/~jurafsky/slp3/