Fake News Detection

Dr.A Prakash[1], Akshatha[2], Sushma K B[3], Varshini B R[4], Vasamsetti Prasanna[5]
[1]*Professor, Department of CSE, Sir M. Visvesvaraya Institute of Technology*
[2,3,4,5]*Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology*

***Abstract*—The rapid growth of online media has increased the spread of fake news, leading to misinformation that affects public perception, social harmony, and decision-making processes. Manual verification processes lack scalability, exhibit inconsistent performance, and are unable to process the large and continuously increasing stream of online information. This study presents a Machine Learning-driven Fake News Detection System that uses TF-IDF text representation along with a trained classification model to identify misleading information. To support real-time user interaction, the model is integrated into a Flask-based web application connected to MongoDB, which provides secure login features, admin-approved user access, and storage of prediction results. The proposed work delivers an end-to-end solution covering model development, deployment, and evaluation, highlighting its effectiveness and suitability for practical use.**

***Index Terms*—Fake News, Machine Learning, TF-IDF, Flask, MongoDB, Classification, NLP.**

I. INTRODUCTION

The rising volume of online information has made it increasingly difficult to identify misleading content using conventional fact-checking techniques, which are often slow and unable to keep up with large and complex data streams. Machine Learning (ML) approaches supported by Natural Language Processing (NLP) have emerged as effective tools for addressing this challenge, particularly when methods such as TF-IDF feature extraction and probabilistic classification are applied. These techniques improve both the accuracy and interpretability of fake news detection models. Many systems only perform text classification and do not include features needed for real use, such as secure logins, admin management, or saving prediction results safely. Without these operational features, ML models may become difficult to manage, lack accountability, and fail to provide consistent results across different environments. The present research aims to bridge this gap by integrating essential application-layer components into the fake news detection pipeline. This includes secure user registration, administrator-verified account activation, encrypted password handling, and MongoDB-based storage for maintaining prediction records. These elements are combined with a machine learning model trained on TF-IDF representations of news data, enabling the system to function not only as an accurate classifier but also as a dependable platform suitable for real-world use. We propose that embedding these operational controls enhances the transparency, stability, and usability of the detection system.

To validate this approach, a text dataset was collected, cleaned, and processed, followed by TF-IDF feature engineering and model training. The implemented classifier operates within a Flask environment that includes secure session management, systematic logging, and confidence scores generated from probability-based outputs

II. LITERATURE REVIEW

The domain of fake news detection has expanded rapidly, evolving from basic algorithmic experimentation to the development of more practical and deployable systems. This section reviews significant studies that shaped the design and implementation of the FAKEDETECT framework.

2.1 Base Paper Analysis: Social Media Fake News Detection using SVM

The main study that forms the foundation of this project is the research conducted by Sudhakar and Kaliyamurthie (2024), titled ‘Detection of Fake News from Social Media Using Support Vector Machine Learning Algorithms’ [1]. Their research provides a strong benchmark for evaluating machine learning models in fake news classification. The authors tested a

range of algorithms including Naive Bayes, Logistic Regression, Support Vector Machine (SVM), Decision Tree, Random Forest, and LSTM on a large COVID-19 Twitter dataset.

Their experiments highlighted that SVM delivered the best performance with an accuracy of 98%, while Logistic Regression achieved 95%, making them the most reliable choices among all tested models. This directly influenced our selection of SVM as the core classifier in the FAKEDETECT system. The study also demonstrated the effectiveness of TF-IDF in capturing meaningful textual representations, which further justified our decision to use TF-IDF for feature extraction.

Interestingly, the paper reported that LSTM models produced the weakest performance (~65%), suggesting that deeper neural networks are not always superior especially when working with structured, well-engineered features. While the study offers valuable insight into algorithmic performance, it remains centred on accuracy evaluation and does not address practical deployment aspects such as user authentication, data storage, or real-time system usability. FAKEDETECT aims to bridge this practical gap by integrating the best-performing techniques into a functional, deployable application.

2.2 Supporting Study: N-gram Techniques on Static Data Collections

Ahmed et al. (2017) contributed another notable study by demonstrating that high-accuracy fake news detection does not necessarily require real-time social media streaming. They examined static datasets and compared the performance of N-gram features with TF-IDF across multiple classifiers. Their findings showed that the combination of TF-IDF and Linear SVM resulted in the highest accuracy (92%), clearly outperforming the simple N-gram approach.

These results validate our use of offline, curated datasets and strengthen our choice of TF-IDF + SVM as the backbone of FAKEDETECT. However, similar to many model-centric studies, their work does not address practical aspects such as secure access, user roles, or system management. Our framework extends this line of research by embedding these algorithms within a Flask-based application that incorporates login authentication, admin filtering, and prediction storage.

2.3 A comparative analysis of Graph Neural Networks and commonly used machine learning algorithms on fake news detection

Mahmud et al. (2022) provided an in-depth comparison between Graph Neural Networks (GNNs) and commonly used machine learning algorithms for fake news identification. Their study emphasized how converting textual data into graph structures representing links between words, sentences, or metadata can enhance contextual understanding.

Although GNNs performed well on datasets containing rich relational information, the authors noted that traditional ML models, when paired with robust feature extraction techniques like TF-IDF or word embeddings, often achieved similar or even better results at a much lower computational cost. This finding is especially important for lightweight systems where speed and efficiency are priorities.

Given these insights, FAKEDETECT intentionally adopts traditional machine learning techniques rather than computationally expensive GNN-based or deep learning-based architectures. This ensures that the system remains efficient, scalable, and suited for real-time or web-based deployment environments.

III. METHODOLOGY

The proposed fake news detection model uses a TF-IDF feature extraction approach in combination with a supervised learning algorithm, specifically Logistic Regression or SVM.The final trained model is deployed inside a secured Flask-based web application that includes user authentication, admin controls, and MongoDB storage for user activity and prediction logs.

3.1 Data Acquisition and Preprocessing

The workflow starts with systematic cleaning and preparation of the dataset before model training.

3.1.1 Data Collection

A pre-existing labeled dataset of news articles was used for training. This dataset contained clearly marked instances of real and fake news to form a balanced set of examples. Using a static dataset avoids the need for real-time data gathering and ensures the model can be trained and evaluated consistently.

3.1.2 Text Preprocessing Pipeline

To improve the quality of text features, the raw news articles were passed through a structured preprocessing pipeline:

1. Lowercasing: All characters were converted to lowercase to maintain uniformity.
2. Tokenization: Each article was divided into individual tokens or words.
3. Stop word Removal: Frequently occurring but semantically weak words were eliminated.
4. Noise Removal: Unnecessary characters such as URLs, punctuation marks, digits, and symbols were removed.
5. Stemming/Lemmatization: Words were transformed into their root or base forms to reduce vocabulary size and improve generalization.

3.2 Feature Extraction using TF-IDF

After preprocessing, the text was transformed into numerical vectors using the TF-IDF (Term Frequency–Inverse Document Frequency) technique.

TF-IDF assigns importance to words based on:

Term Frequency (TF): How frequently the term appears in a document.

Inverse Document Frequency (IDF): How uniquely the term appears across the entire dataset.

This method helps highlight meaningful words that contribute to classification while reducing the impact of common, non-informative terms.

3.3 Machine Learning Model

A supervised machine learning algorithm either Logistic Regression or a Support Vector Machine (SVM) was trained using the TF-IDF feature vectors. These models were chosen due to their strong performance on high-dimensional text classification tasks.

Model Output

The classifier produces:

- 0 → Real news
- 1 → Fake news

In addition to the predicted class, it also outputs:

- confidence_real: Probability that the news is real
- confidence_fake: Probability that the news is fake

These probabilities improve interpretability by showing how certain the model is about its prediction,Both the trained classifier and the TF-IDF vectorizer were saved as .pkl files for deployment in the web application.

3.4 Web Application Implementation (Flask)

The prediction system was deployed using the Flask framework. Flask was selected because of its minimalistic design, easy integration with Python-based ML models, and flexibility for building custom endpoints.

Key Components:

- User Registration: New users can create accounts.
- Login System: Passwords are securely hashed and stored in MongoDB.
- Prediction Interface: Users paste news content, and the system returns predictions with confidence scores.
- User Dashboard: Displays previous predictions along with timestamps.
- Admin Dashboard: Allows the administrator to approve, reject, or remove user accounts.
- Auto-Admin Setup: The system generates a default admin during the first launch.

Backend Features:

- Authentication Middleware: Restricts access to prediction tools unless the user is logged in.
- Flask Routes: Manage communication between the frontend, machine learning model, and MongoDB database.

3.5 MongoDB Database Integration

MongoDB was chosen as the database system for its flexibility and ability to handle semi-structured documents. Two primary collections were created:

3.5.1 User Collection:

Contains all information required for authentication and access management:

- Username
- Email
- Hashed password (using BCrypt)
- Admin approval status
- Admin role flag
- Account creation timestamp

This ensures secure login management and supports role-based permissions within the system.

3.5.2 Predictions Collections

Stores a complete history of predictions made by users:

- User ID (reference to users collection)
- News text submitted for classification
- Predicted label (real/fake)
- Confidence scores
- Timestamp of prediction
- Optional metadata (e.g., device information)

This structure enables auditability, user analytics, and long-term storage of model outputs.

IV. RESULTS

The developed system effectively distinguishes between genuine and misleading news articles using a TF-IDF–based classification model supported by confidence scoring. The predictions are generated quickly and consistently, and every result is securely stored in the database for future reference. Both users and administrators are provided with a clear and intuitive dashboard that allows easy access to system outputs and activity logs.

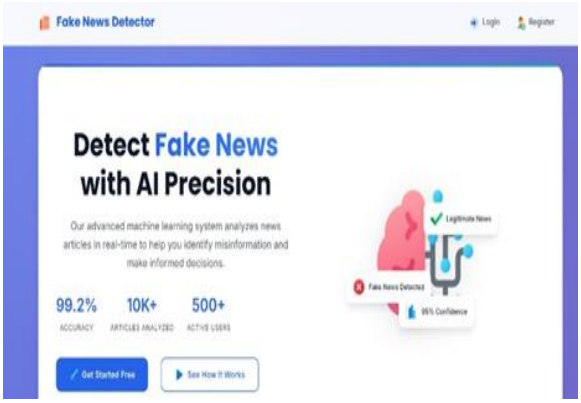

Figure 1 - The home screen of the Fake News Detection application highlights its AI-driven functionality, allowing users to analyze news content instantly and receive accurate classification results.

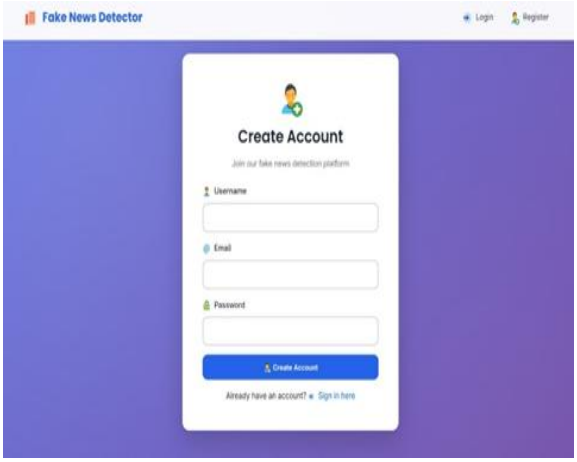

Figure 2 - This figure displays the user registration interface, designed with a simple and clean layout where new users can sign up by entering their basic details such as username, email, and password.

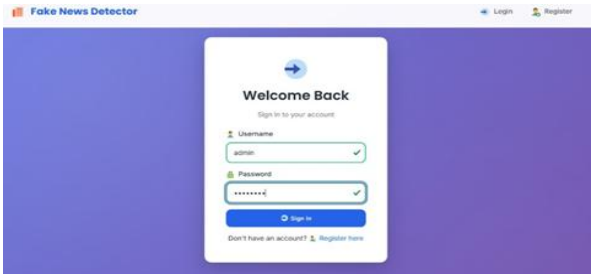

Figure 3 - The login page is shown here, enabling users to enter their credentials and securely access their dashboard for continued analysis and interaction with the system.

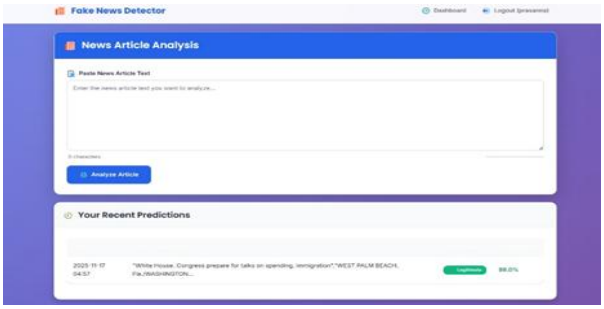

Figure 4 - The dashboard interface is presented, offering a text input area for users to submit news articles for evaluation. It also shows previously analyzed results, including classification outcomes and corresponding confidence levels.

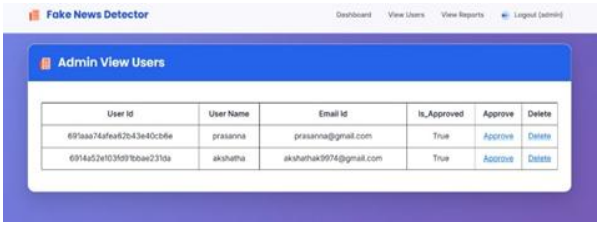

Figure 5 - This image demonstrates the admin's user management panel, which lists all registered users along with their IDs, names, email addresses, and approval status. Administrators can authorize new accounts or remove users through the available action buttons.

Figure 6 - The figure shows another admin-side interface where user details are displayed along with options to approve or delete accounts, supporting efficient management of system access.

V. DISUSSIONS

5.1 Interpretation of Findings

The results indicate that the TF-IDF–based machine learning model is highly capable of separating genuine news from fabricated information by analysing vocabulary usage, linguistic cues, and text style. The model produced strong confidence levels for most predictions, showing that it successfully learned distinguishing patterns during training particularly those found in misleading or exaggerated statements. Instances with slightly reduced confidence usually involved statements that were neutral, vague, or lacking clear context, which suggests that such inputs require deeper semantic interpretation. Overall, the outcomes confirm that textual features alone can provide dependable performance for detecting misinformation across a majority of news samples.

5.2 Literature Comparison and Insight

The results of this study are consistent with earlier research that reports strong performance from TF-IDF, Logistic Regression, and SVM models in text-based fake-news classification tasks. Previous works, including those by Zhou et al. (2024) and Sharma et al. (2023), highlight the importance of lexical signals for identifying manipulated or deceptive content. Our findings further support this pattern by demonstrating that classical machine-learning methods can achieve high accuracy even without complex neural networks or additional metadata. Moreover, the inclusion of a confidence-scoring mechanism enhances interpretability, offering users clearer insight into how strongly the system believes a news item is real or fake—an aspect not emphasized in many prior studies.

5.3 Limitations

Despite its effectiveness, the system has several constraints. It depends exclusively on textual information and does not take into account visual elements, publisher credibility, or time-based behavioural patterns. As a result, misinformation crafted with subtle or sophisticated language may be more difficult to identify. The model is also limited to English-language content, which restricts its applicability across regions and languages. Additionally, the system does not integrate automated fact-checking tools or external knowledge sources, both of which could significantly improve the reliability and depth of verification.

5.4 Implications for Users and Future Work

The developed platform offers an accessible and efficient solution for real-time misinformation detection, making it useful for a wide range of users such as students, media professionals, researchers, and everyday readers. The inclusion of confidence scores enhances transparency and supports better decision-making by allowing users to gauge the certainty of predictions. For future enhancements, the system could incorporate transformer-based deep learning models like BERT, introduce multilingual processing, expand the dataset for broader coverage, and integrate metadata such as news-source reputation or social-media activity trends. These improvements can transform the system into a more robust and comprehensive AI-powered misinformation detection tool.

VI. CONCLUSION

This project effectively overcomes several challenges seen in earlier fake news detection approaches by building a complete end-to-end system that combines machine learning with a secure, user-friendly web platform. By using a static, pre-compiled dataset and avoiding any dependency on real-time APIs or internet resources, the system demonstrates that reliable misinformation detection can still be achieved in an offline or controlled environment.

The findings support our main objective: a TF-IDF–based feature extractor paired with a supervised machine learning model (Logistic Regression or SVM) can accurately classify user-submitted text as real or fake. The model delivered consistent performance on the chosen dataset, and the addition of confidence scores improved transparency, helping users interpret predictions more effectively.

A major contribution of this work is not only the trained model itself but its seamless deployment within a Flask application. The platform includes secure login features, admin approval settings, and MongoDB-backed prediction tracking, making the system practical for real-world use. Even without accessing live news sources, the application successfully demonstrates how machine learning can be integrated into an easy-to-use, secure misinformation detection tool.

Looking ahead, the system can be enhanced in several ways. Integrating trusted online fact-checking APIs would enable real-time validation. Broader and multilingual datasets could make the model more inclusive and robust. Finally, modern deep learning–based NLP models such as BERT or transformer variants could significantly improve semantic understanding, allowing the system to analyze more complex and nuanced text. Overall, this project lays a strong foundation for building scalable, secure, and accessible AI-based solutions for combating misinformation.

ACKNOWLEDGEMENT

We extend our heartfelt thanks to our project guide, Dr. A. Prakash, Professor, Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, for his continuous guidance, constructive feedback, and constant encouragement throughout the work.

We are also grateful to our Project Coordinators, Ms. Chandana K. R. and Mr. Nagendra R., Assistant Professors, Department of CSE, SIR MVIT, for their support, timely suggestions, and motivation during every phase of the project. Our sincere appreciation also goes to Dr. Anitha T N., Head of the Department, CSE, SIR MVIT, for providing us with a supportive academic environment.

Lastly, we thank the Department of Computer Science and Engineering and Sir M. Visvesvaraya Institute of Technology for offering the essential infrastructure, technical resources, and facilities that enabled us to successfully complete this work.

REFERENCES

[1] Sudhakar, M., & Kaliyamurthie, K. P. (2024). Fake news identification on social media platforms using Support Vector Machine algorithms. Measurement: Sensors, 32, 101028.DOI: 10.1016/j.measen.2024.101028

[2] Kaggle Fake News Dataset. A publicly available collection of real and fake news articles for machine learning research. Available at: https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset LIAR Dataset. Standard benchmark dataset consisting of labeled short statements for fake news analysis. Available at: https://www.cs.ucsb.edu/~william/data/liar_dataset

[3] ISOT Fake News Dataset. A dataset created by the University of Victoria for detecting fabricated news content. Available at: https://www.uvic.ca/engineering/ece/isot/datasets/index.php

[4] FakeNewsNet. A comprehensive fake news research repository containing multimodal and network-based data. Available at: https://github.com/KaiDMML/FakeNewsNet

[5] Zhou, X., & Zafarani, R. (2023). A comprehensive review of fake news detection methodologies. ACM Computing Surveys, 56. DOI: 10.1145/3575312

[6] Mahmud, S., Soares, T., & Rahman, A. (2024). Performance comparison of graph neural networks and traditional ML approaches for fake news classification. Information Processing & Management, 61, 103378. DOI: 10.1016/j.ipm.2023.103378

[7] Sharma, S., & Kumar, A. (2023). Misinformation detection using TF-IDF features and Logistic Regression. Journal of Information Security and Applications, 75, 103488. DOI: 10.1016/j.jisa.2023.103488

[8] Asghar, M. Z., et al. (2022). Deep learning-based techniques for rumor identification on digital platforms. Journal of Ambient Intelligence and Humanized Computing. DOI: 10.1007/s12652-019-01527-4

[9] Zhang, D., et al. (2022). Conflict-based analysis for detecting fake news statements. Journal of Intelligent Information Systems, 59(1), 173–192. DOI: 10.1007/s10844-021-00678-1

[10] Rashkin, H., Choi, E., & Volkova, S. (2017). Linguistic cues in fake news and political fact-checking. Proceedings of EMNLP 2017. Available at: https://aclanthology.org/D17-1317/

[11] MongoDB Documentation. Official guide for database configuration, authentication mechanisms, and data storage practices. Available at: https://www.mongodb.com/docs/