

Fake News Detection Using NLP and Machine Learning

Mansi Barewar

Department of Computer Science
and Design, YCCE Nagpur

Vaishnavi Bawankar

Department of Computer Science
and Design, YCCE Nagpur

Himanshu Satpute

Department of Computer Science
and Design, YCCE Nagpur

Adil Dhote

Department of Computer Science
and Design, YCCE Nagpur

Vallaree Saikhedkar

Department of Computer Science
and Design, YCCE Nagpur

ABSTRACT: Misinformation and fake news pose an increasing threat to the democratic processes and society. A replicable pipeline for automatic fake-news identification utilizing machine learning and a traditional Natural Language Processing (NLP) is shown in this paper. In order to train three classifiers Logistic Regression, Multinomial Naive Bayes, and Random Forest we gathered publically accessible datasets of news that had been tagged as "True" or a "False," applied text preprocessing, and extracted TF-IDF features (unigram + bigram).

Keywords: Fake news detection, text classification, TF-IDF, Logistic Regression, Random Forest, Naive Bayes, reproducible ML, Streamlit.

I. INTRODUCTION

three classifiers Random Forest, Multinomial Naive Bayes, and Logistic Regression and TF-IDF feature extraction (unigrams + bigrams). Accuracy, confusion matrix, precision, recall, and F1 were used to evaluate the models after they were trained using an 80/20 stratified train/test split. Random Forest achieved the best test performance (Test Accuracy: 99.69%), Logistic Regression and Naive Bayes achieved 98.71% and 92.74%, respectively. We packaged the system into Streamlit web app demonstrating real-time predictions and reported deployment considerations. We discuss limitations (e.g., domain shift, headline vs. full-text performance), ethical risks, and directions for future work including transformer-based embeddings and headline-specific models. This report provides reproducible code, model files, and detailed evaluation to support the replication and further research.

Future improvements may explore multilingual support, adaptive learning from evolving news patterns, and integration with APIs for live fact-checking to enhance both scalability and societal impact. The Streamlit web app enables users to input news headlines or full articles and instantly receive credibility scores along with the confidence

visualizations

In this project, three main models Random Forest, Multinomial Naive Bayes, and Logistic Regression were trained using TF-IDF features that included both single words and short phrases (unigrams and bigrams). The dataset was divided into training and testing sets in an 80:20 ratio, ensuring a balanced representation of both fake and real news articles. To evaluate how well the models performed, several metrics were used, including accuracy, precision, recall, F1-score, and confusion matrix. These methods measured not only overall correctness but also how well each model identified fake news without confusing real content. Among the three, Random Forest had the highest accuracy at 99.69%, demonstrating its ability to handle complex text data. Logistic Regression came in next at 98.71%, providing both high accuracy and easy understanding. Naive Bayes, although simpler, still achieved a strong 92.74%, showing its effectiveness for quick text classification. The final system was integrated into a Streamlit web app, allowing users to enter any news headline or article and quickly receive a credibility score along with a visual confidence level. This makes the system interactive, user-friendly, and accessible for real-time analysis. The study also highlighted key challenges such as domain shift, which occurs when the model encounters unfamiliar types of news, and the difference between analyzing short headlines and full-length articles. Ethical concerns, including the potential for mislabeling and bias, were also noted. For future work, the project plans to add multilingual support, use transformer-based language models like BERT for improved accuracy, and incorporate real-time fact-checking through APIs. These updates would make the tool more flexible, scalable, and effective in fighting misinformation on a global scale.



Fig 1: Fake news detection system

II. LITERATURE REVIEW

The rapid growth of online media and social networking platforms has made the spread of misinformation and fake news a major global concern. Manual fact-checking is time consuming and often impractical at scale, which has motivated large-scale research into automated fake news detection using Natural Language Processing (NLP) and Machine Learning (ML). Over the past decade, researchers have proposed the variety of technique ranging from traditional text classification techniques to modern deep learning architectures.

Early studies focused on rule-based methods and extracting linguistic features. Researchers used manually crafted lexical, syntactic, and semantic cues to identify deceptive text. These methods were easy to understand but struggled with scalability and adjusting to different writing styles. As datasets increased in size, Bag-of-Words (BoW) and TF-IDF (Term Frequency-

III. SYSTEM METHODOLOGY

Overview

The approach shows outlines of the methodical Procedure used to create, hone, and implement the fake-news detecting system. Data collection and labeling are the first steps in the project's well-defined workflow, which then moves on to text preprocessing and feature extraction before model training, assessment, and deployment via a Streamlit interface. Each step was carried out to guarantee excellent predicted accuracy, interpretability, and reproducibility.

This study clearly outlines how to create and implement a fake news detection system. It starts by collecting and labeling data to ensure that both real and fake news articles are correctly categorized. Once the data is ready, it goes through several text preprocessing steps to clean

Inverse and Document Frequency) representations became popular for turning text into numerical data. With these features, models like Logistic Regression, Support Vector Machines (SVMs), and Naive Bayes achieved good accuracy and were easy to understand on standard datasets.

For example, **Rubin et al. (2016)** analyzed linguistic deception cues using Naive Bayes classifiers and demonstrated that word-level patterns could help detect misinformation. **Ahmed et al. (2018)** explored TF-IDF features combined with Logistic Regression and achieved competitive accuracy on Kaggle's fake news dataset, highlighting the strength of traditional ML methods in text classification. Similarly, **Ajao et al. (2019)** experimented with ensemble models, showing that combining decision trees in Random Forests improved stability and reduced overfitting.

With progress in deep learning, researchers started using the neural networks like Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks for contextual feature learning. These models captured semantic relationships between words and performed better than traditional methods on large-scale datasets. Recently, Transformer-based architectures like BERT, RoBERTa, and XLNet have changed fake news detection by offering contextual encoding and transfer learning features. However, these models are costly to compute and need fine-tuning on specific data to work well. Logistic Regression (LR) soon became a popular alternative because it offers better generalization and the clarity through weight coefficients.

and prepare it for analysis. After that, key features are extracted using TF-IDF, which turns the text into numerical values that machine learning models can understand.

Next, different models are trained and fine-tuned to learn the patterns that set real news apart from fake news. Each model's performance is carefully assessed using standard metrics to find out which one delivers the most accurate and reliable predictions. Finally, the best-performing model is launched through the interactive Streamlit web application, enabling users to test the system in real time.

Data Collection and Preparation

Two publicly accessible CSV files, True.csv and Fake.csv, taken from Kaggle's Fake and True News Dataset, comprise the dataset used.

1. A news title, the primary text, the publication date,

and the subject label are all included in each record.

2. The labeled articles from Fake.csv were marked as 0 (Fake news) and those from True.csv as 1 (True news). These two files create a balanced dataset that helps train and test fake news detection models. The data includes a variety of topics, such as politics, world news, and current events. This variety makes it useful for studying how misinformation spreads across different subjects. Since the dataset comes from actual online articles, it offers a realistic scenario for building and testing models that can distinguish true information from fake content.

Data Preprocessing

Before training the model, the text data was cleaned and standardized to remove noise and ensure uniformity across samples. The preprocessing pipeline is made up of:

The technique of converting all text to lowercase in order to remove case sensitivity is known as the lowercasing(lowercase).

Taking URLs down: Remove patterns like http://, https://, or www using regular expressions. Punctuation removal is the process of removing special characters and symbols that don't add to the semantic meaning of any importance. Whitespace normalization entails collapsing multiple spaces and eliminating unnecessary whitespace. Stopword handling is the procedure of using the built-in English stopwords list to eliminate non-informative terms from TF-IDF.

Feature Extraction

Term Frequency-Inverse Document Frequency (TF-IDF) vectorization was used with TfidfVectorizer (max_features = 5000, ngram_range = (1, 2), stop_words = 'english') to transform textual data into numerical form.

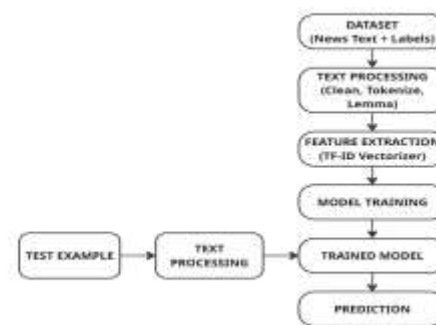
In order to ensure efficient computation, the max_features parameter chooses the 5,000 most informative tokens based on inverse document frequency.

Both single-word (unigram) and two-word (bigram) phrases are captured by the ngram_range (1, 2), which enables the model to comprehend brief semantic patterns frequently found in the upper head headlines. For consistency, the vectorizer was first fitted to training data using fit_transform(), and then it was applied to test or fresh data using transform(). All machine learning models use the generated sparse matrix as their input features.

To turn the news articles into a format that a machine learning model can understand, TF-IDF vectorization

was applied. This method converts text into numbers based on how often certain words or phrases appear in the articles. The idea is simple words that show up often in one article but not in many others are considered more important. This helps the model focus on meaningful terms rather than common ones like "and" or "the."

Fig2: Workflow of prediction using model



Model Training

Three supervised machine learning models were trained and compared:

Logistic Regression – implemented with max_iter = 1000 to ensure convergence. It establishes a linear decision boundary and provides interpretable feature weights for each token.

Multinomial Naive Bayes – suitable for the discrete count features; fast and efficient for text classification tasks.

Random Forest Classifier – configured with n_estimators = 100 and random_state = 42, forming an ensemble of decision trees that handle non-linear patterns and reduce overfitting.

For this study, three different supervised machine learning models were trained and compared to see which one performed best in detecting fake news. Each model works in its own way and brings unique strengths to the task.

IV.RESULTS AND DISCUSSIONS

This chapter presents and interprets the results obtained from the three machine learning models Logistic Regression, Naive Bayes, and Random Forest used for fake news detection. The models were evaluated based on training and testing accuracy, confusion matrices, and key performance metrics. The discussion also highlights observations related to overfitting, generalization, and practical implications of model performance.

Model	Train Accuracy (%)	Test Accuracy (%)	Remarks
Logistic Regression	99.16	98.72	Balanced and interpretable
Naive Bayes	94.39	94.38	Fast but less precise
Random Forest	99.97	99.78	Highest accuracy, slight overfit

Table 1 Models Training and Testing Accuracy

The Random Forest model outperforms the other two algorithms, according to the comparison analysis. It achieves the best accuracy and robustness because of its ensemble nature, which enables it to capture intricate, non-linear relationships within the text. Nearly identical performance is provided by logistic regression, which indeed is also more interpretable and computationally efficient. On the other hand, despite being quick and easy to apply, the Naive Bayes model performs very poorly since it makes the assumption that features are independent of one another, which restricts its capacity to comprehend more complex contextual relationships in news content. Best Model is found to be Random Forest.



Fig3: Various Model Used

Research Proposition

Digital media's rapid expansion has made it possible for information to travel around the world in a matter of seconds, but it has also sped up the distribution of false information and fake news. The volume and speed at which online content is created and disseminated make traditional fact-checking methods insufficient. An automated, data-driven system that can consistently discern between authentic and fake information is desperately needed in particular and important tupe of this scenario. The main hypothesis of the study is that by identifying linguistic and contextual patterns in textual data, machine learning models in conjunction with Natural Language

Processing (NLP) approaches may reliably identify fake news. The study specifically posits that a well-designed pipeline comprised of classical machine learning, TF-IDF-based feature extraction, and robust text preprocessing can attain high accuracy and reproducibility while maintaining interpretability and computational efficiency.

To support this idea, the study combines several supervised algorithms, including Logistic Regression, Multinomial Naive Bayes, and Random Forest. These models are trained on a labeled dataset of real and fake news articles. The researchers evaluate and compare the models using performance metrics such as accuracy, precision, recall, F1-score, and confusion matrices. The best-performing model is then deployed through a web interface created with Streamlit to show its real-time detection ability.

V. CONCLUSION AND FUTURE WORK

This study offers a comprehensive and repeatable method for identifying false information through the use of machine learning (ML) and natural language processing (NLP). Data collection, text preprocessing, TF-IDF feature extraction, model training, and real-time deployment via a Streamlit interface comprise the methodical pipeline that the suggested system uses. Using a labeled dataset of real and fraudulent news items, three traditional machine learning algorithms Random Forest, Multinomial Naive Bayes, and Logistic Regression were put into practice and assessed. With a test accuracy of 99.69%, the Random Forest classifier outperformed the others, followed by Logistic Regression (98.71%) and Naive Bayes (92.74%). The findings confirm that classical machine learning models work very well for text classification tasks like identifying false news when paired with robust feature engineering. This research offers a straightforward way to detect fake news using machine learning and natural language processing techniques. The process begins with gathering reliable data and cleaning it for consistency. Next, it converts the text into numerical form using TF-IDF. After that, different models are trained and evaluated. Finally, the system is launched through a Streamlit web interface, which enables real-time testing and user interaction. This step-by-step setup makes it easy to repeat or improve the process later.

VI. REFERENCE

- [1] A. Ahmed, M. Traore, and S. Saeed, "Detecting Fake News Using Machine Learning: A Survey," Journal of Information Systems Engineering and Management, vol. 3, no. 2, pp. 1-9, 2018.
- [2] V. Rubin, N. Conroy, Y. Chen, and S. Cornwell, "Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News," Proceedings of the Second Workshop on Computational Approaches to Deception Detection, pp. 7-17, 2016.
- [3] P. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein, "A Stylometric Inquiry into Hyperpartisan and Fake News," Proceedings of the 56th Annual Meeting of the

Association for Computational Linguistics, pp. 231-240, 2018.

[4] A. Ajao, D. Bhowmik, and S. Zargari, "Fake News Identification on Twitter with Hybrid CNN and RNN Models," *Proceedings of the 9th International Conference on Social Media and Society*, pp. 226-230, 2018.

[5] Y. Wang, "Liars Begin with Self: Learning Attention and Memory Models for Fake News Detection," *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pp. 241-250, 2018.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 4171-4186, 2019.

[7] R. Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake News Detection in Social Media with a BERT-based Deep Learning Approach," *IEEE Access*, vol. 8, pp. 217503-217511, 2020.