International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 3, May-June-2026, ISSN (Online): 2395-566X

Enhancing Fake News Detection through Optimized Feature Engineering and Supervised Machine Learning

Anuradha Muttamwar[1], Esha Dorkhande[2], Vaibhavi Meshram[3]
[1]Department of Master in Computer Application, GHRCEM, Nagpur, India
Email: anuradhab1992@gmail.com
[2]Department of Master in Computer Application, GHRCEM, Nagpur, India
Email: eshadorkhande@gmail.com
[3]Department of Master in Computer Application, GHRCEM, Nagpur, India
Email: vaibhavimeshram5@gmail.com

Abstract — The exponential proliferation of digital media in the modern era has created an environment where mis- and disinformation as well as "fake news" can spread uncontrollably, leading to challenges to public discourse, political trust and integrity. In this paper we present a detailed research approach toward fake news detection through efficient feature engineering and the use of supervised machine learning. We use a dataset composed of 5,000 current news articles (2,537 real, 2,463 fake news) and conduct an in-depth research regarding the performance of TF-IDF with n-grams. We build and train a Multinomial Naive Bayes model and attain excellent classification accuracy. Furthermore, we investigate the importance of text preprocessing such as stop word removal, stemming and lemmatization. Our model achieves a final accuracy of 93.6%, while also achieving scores for precision, recall and F1 greater than 0.92. When comparing with baseline models, the presented method with enhanced feature engineering shows excellent results. We then developed a web based system with the help of Flask that allows real time fake news detection and confidence. It will establish a reusable, light and scalable pipeline to automate fake news detection in real world applications.

Keywords— fake news detection; natural language processing, TF-IDF, Naive Bayes, supervised machine learning; feature engineering, misinformation, text classification

I. INTRODUCTION

The rapid spread of misinformation via online platforms has emerged as an important problem in the contemporary world. Misinformation, which involves the creation and spreading of false information or fake news as real reporting, can have serious effects on public opinion, elections, and social unrest [1]. Misinformation related to health has been prevalent during the coronavirus disease outbreak and had physical repercussions worldwide.

Efforts to detect fake news traditionally involve manual fact-checking or other editorial practices but are ineffective due to scale and time considerations. With numerous articles appearing on a regular basis via various news portals, blogs, and social media networks, automated fake news detection becomes inevitable. Machine learning (ML) combined with natural language processing (NLP) techniques can solve this problem effectively.

The current study aims to develop an optimized fake news detection system based on supervised machine learning. Feature engineering, utilizing TF-IDF vectorization along with different configurations of n-gram representation of words, will be implemented within the scope of this work. The model of a Multinomial Naïve Bayes classifier trained on a set of 5,000 recent news articles will be developed. Finally, the detection system will be incorporated into the Flask framework to form a real-time news analysis platform.

The main novelties of this paper include: (1) a thorough analysis of the TF-IDF feature engineering algorithm and its impact of the n-gram parameter variation; (2) a detailed investigation of the performance of the Naïve Bayes classification technique on the recent news corpus; (3) development of an optimized deployment architecture for the proposed fake news detection tool; and (4) discussion on text preprocessing and classification quality improvement.

II. LITERATURE REVIEW

In recent years, there has been much research into automated detection of fake news articles. Some early techniques relied on textual characteristics such as language use, sentence structure, and emotional tone to detect fake news versus real news stories[1]. As shown by Potthast et al., stylometric features based on writing style could be used to identify hyperpartisan news websites with reasonable accuracy [2].

With the rise of deep learning, recurrent neural networks (RNNs) and long short-term memory (LSTM) models became common architectures for sequence prediction and analysis. Li et al. introduced the LIAR dataset and benchmarked several machine learning models to find that CNNs worked better than traditional classifiers at statement-level detection of fake news [3]. The downside to deep learning methods is their reliance on computation and large amounts of labeled data.

However, transformer models such as BERT have taken over the current state of natural language processing. BERT fine-tuned on different tasks shows excellent performance on tasks related to fake news detection. These methods perform very well but are still computationally costly and hard to deploy.

Classical machine learning techniques based on TF-IDF and naive Bayes classifiers have proven to be both accurate and computationally less costly alternatives [5]. In fact, Ahmed et al. reported over 90 percent accuracy on the ISOT Fake News Dataset using a passive-aggressive classifier with TF-IDF features [6]. This backs up our approach of focusing on optimizing classical machine learning pipelines.

Metadata, social propagation patterns, and knowledge graph based verification of news claims have also received attention recently [7]. However, content-based approaches still dominate fake news detection research due to lack of social context data.

III. DATASET DESCRIPTION

In this project, the dataset that will be used consists of 5,000 articles recently published online on various news websites that cover topics including but not limited to politics, healthcare, technology, and societal issues. Each article will be labeled with one of the two classes; namely, Real or Fake.

The details of the dataset are highlighted in Table I below. The data is well balanced, having 2,537 real articles (50.74%) and 2,463 fake articles (49.26%). Four variables are included in each record; these are article date, article title, article body, and article classification.

TABLE I. Dataset Statistics

Attribute	Value
Total Articles	5,000
Real News	2,537 (50.74%)
Fake News	2,463 (49.26%)
Average Article Length	~420 words
Features	date, title, text, label
Date Range	2024–2025
Language	English

The fabricated news articles were collected from sources known for distributing misleading information and satire, whereas authentic news articles were collected from reliable sources. There were no duplicate entries detected after data pre-processing. The dataset encompasses several categories, such as artificial intelligence, health, politics, economy, and technology, which makes it highly generalizable.

IV. METHODOLOGY

1. System Architecture

The proposed architecture consists of the following five phases: (1) Data collection & pre-processing phase, (2) Feature extraction using TF-IDF vectorization, (3) Model training using Multinomial Naive Bayes classification algorithm, (4) Evaluation & optimization of model, and (5) Deployment of models using Flask web framework. The architecture is flexible and can be modified by replacing either feature extraction or classification module.

2. Text Preprocessing

Unprocessed text passes through multiple transformation layers prior to feature extraction, including: (i) lowering to make the model insensitive to capitalization, (ii) stripping away any punctuations and numeric entities, (iii) breaking text into tokens based on whitespaces, (iv) eliminating stopwords in English language using NLTK library's standard set, (v) and performing stemming via Porter Stemmer.

Article titles and their body texts are concatenated to increase the discriminative power of per-sample representations.

3. Feature Engineering: TF-IDF with N-Grams

Term Frequency-Inverse Document Frequency (TF-IDF) serves as the primary approach to features generation in this study. Specifically, the approach combines term's occurrence within a particular document with its inverse occurrence frequency in the entire collection, thus emphasizing discriminative words and discarding non-informative ones.

The TF-IDF vectorizer uses unigrams and bigrams (1~2), limits vocabulary to 5000 features with sublinear TF scaling. Bigrams enable capturing multi-word expressions that convey stronger classification signals compared to single terms, e.g., 'fake

claim' or 'verified source'. Thus, the generated TF-IDF matrix has 5000 x 5000 dimensions.

According to Equation (1) below, the formula for computing TF-IDF score for term t in document d is tf(t, d) x log(N/df(t)), where N is the number of documents and df(t) is the number of documents in which term t occurs.

$$tfidf(t, d) = tf(t, d) \times \log(N / df(t)) \quad ...(1)$$

4. Classification: Multinomial Naive Bayes

Multinomial Naive Bayes (MNB) was selected because of its established efficiency for text classification problems with feature frequencies modeled by multinomial distribution (which is true for TF-IDF weights). According to Bayes' theorem and MNB conditional independence assumption (2),

$$P(y \mid x_1,...,x_n) \propto P(y) \prod P(x_i \mid y) \quad ...(2)$$

where y stands for the class label (True/False), and x_i is the ith feature value. Furthermore, Laplace smoothing (alpha=1.0) will be used to solve for zero-probability terms.

5. Model Training and Evaluation

The dataset is then divided into 80% training dataset and 20% testing dataset where the former includes 4,000 samples while the later includes 1,000 samples. Training the model is done through processing the TF-IDF vector with regard to its accuracy, precision, recall and F1-score. Along with that, five-fold cross-validation is performed on the dataset.

Tuning the hyperparameters is done using the Grid Search method with respect to the size of the vocabulary, ranging from 1,000 to 10,000; n-grams ((1,1) and (1,2)); and the value of alpha ranging from 0.1 to 2.0.

V. EXPERIMENTAL RESULTS

1. Classification Performance

The classification results of the proposed approach are illustrated in Table II for the testing data set. With the optimization of the TF-IDF + MNB model, the classification accuracy can reach up to 93.6%.

TABLE II. Classification Performance Metrics

Class	Prec.	Recall	F1	Support
Real	0.94	0.93	0.935	507
Fake	0.93	0.94	0.935	493
Weighted Avg	0.936	0.936	0.936	1000
Accuracy	94%	94%	93.6%	1000

2. Effect of N-Gram Configuration

The findings achieved using different combinations of n-grams are presented in Table III below. The implementation of bigrams leads to an increase in the F1-score by 2.1% as opposed to unigrams, and therefore, multi-word expressions offer useful discrimination for classification.

TABLE III. N-Gram Configuration Comparison

N-Gram Range	Accuracy (%)	F1-Score
(1,1) Unigrams	91.4	0.914
(1,2) Unigram + Bigram	93.6	0.936
(2,2) Bigrams Only	87.2	0.871
(1,3) Up to Trigram	93.1	0.930

3. Cross-Validation Accuracy

A mean accuracy of 92.8% ± 1.2% is obtained via five-fold cross-validation, showing that the model's performance is reliable and no overfitting is present. The small standard deviation proves that the results are consistent between folds.

4. Comparison to Other Models

In Table IV, we compare our new classifier to other classifiers trained on the same set of TF-IDF features. Although Logistic Regression and SVM models produce slightly better accuracies, Multinomial Naïve Bayes is far more computationally efficient.

TABLE 4. Classifier Comparison (TF-IDF Features)

Classifier	Accuracy	F1-Score	Inference (ms)
Naive Bayes (Proposed)	93.6%	0.936	~2
Logistic Regression	94.8%	0.948	~5
SVM (Linear Kernel)	95.1%	0.951	~18

Decision Tree	88.3%	0.882	~3
Random Forest	91.7%	0.916	~45

VI. SYSTEM DEPLOYMENT

The neural network model and the TF-IDF vectorizer have been serialized using Python's pickle library and incorporated into a Flask web application. The deployment infrastructure facilitates the real-time classification of news articles, with each request being processed within less than 10 milliseconds. The web application incorporates user authentication with hashing of passwords (using Werkzeug library), secure routes via sessions, and a user prediction history page with information stored in an SQLite database. Users provide input for the news text in an HTML5 frontend designed using CSS3 glassmorphics theme, getting immediate output including a confidence score percentage.

It can be accessed at http://localhost:5000, using a RESTful API framework design, which allows for future integration of this web application with browser or mobile apps. Cross-Origin Resource Sharing security is implemented, ensuring that there will be no SQL injection.

VII. DISCUSSION

The findings confirm the applicability of the optimal TF-IDF feature engineering coupled with the Multinomial Naive Bayes algorithm in fake news classification tasks. The model accuracy rate of 93.6% demonstrates better performance than many more sophisticated methods proposed in related studies, with the added advantage of easy implementation and deployment.

The crucial role played by the use of bigrams (2.2% higher accuracy than unigrams) emphasizes the relevance of semantic phrases in the task at hand. In particular, fake news articles use distinctive language structures containing inflammatory elements that would go unnoticed when using a unigram model only.

One of the primary limitations associated with the presented approach is the absence of any other type of features aside from those extracted from the textual content of news articles. Social media context, including author or website credibility, the process of propagation within a network, and temporal factors, is disregarded.

Finally, it should be noted that the Naive Bayes model makes the simplifying assumption about the independence of textual features, which causes a degree of bias. Transformer-based contextual embeddings could serve as an alternative representation layer for the model input.

VIII. CONCLUSION

This research proposed a complete system for detecting fake news by applying optimized TF-IDF features and Multinomial Naive Bayes classifier. An experiment using a balanced data set consisting of 5,000 recent news articles proved that the performance of the system is highly efficient, reaching 93.6% accuracy, with both precision and recall higher than 0.93 at all times. The addition of bigrams as a part of n-grams is highlighted as one of the factors contributing to the increase of the model performance.

A Flask-based web application for news article classification was created based on the model used in this study. The lightweight nature of such applications makes it easily scalable and applicable in any situation.

Acknowledgment
The authors gratefully acknowledge the Department of Computer Applications for the support rendered in this work. A special acknowledgment goes to our professors who provided us with important inputs while developing the system.

REFERENCES

1. H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," Journal of Economic Perspectives, vol. 31, no. 2, pp. 211–236, 2017.
2. M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein, "A stylometric inquiry into hyperpartisan and fake news," in Proc. 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, 2018, pp. 231–240.
3. W. Y. Wang, "'Liar, liar pants on fire': A new benchmark dataset for fake news detection," in Proc. 55th Annual Meeting of the ACL, Vancouver, 2017, pp. 422–426.
4. J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep
5. K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," ACM SIGKDD Explorations Newsletter, vol. 19, no. 1, pp. 22–36, 2017.
6. H. Ahmed, I. Traore, and S. Saad, "Detection of online fake news using N-gram analysis and machine learning

techniques," in Proc. International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, 2017, pp. 127–138.
7. N. Ruchansky, S. Seo, and Y. Liu, "CSI: A hybrid deep model for fake news detection," in Proc. ACM International Conference on Information and Knowledge Management, 2017, pp. 797–806.