# Detecting Fake News Using an Ensemble of Machine Learning Models

Prof. Tirth N Patel[1], Dr. Narayan Joshi[2]
[1]*Assistant Professor, GLS University Ahmedabad*
[2]*Professor and Head, Dharmsinh Desai University Nadiad*

***Abstract*—The fast spread of information via digital channels has revolutionized news consumption, but it has also accelerated the spread of fake news—misleading or fabricated content designed to manipulate audiences. To ensure that the students receive accurate information, detection of fake news becomes crucial. The main aim of this project is to explore different machine-learning based approaches to automate the detection of fake news using text classification. We are using three machine-learning models namely Logistic Regression (LR), Random Forest Classifier and Gradient Boosting Classifier (GBC) and applied Term Frequency-Inverse Document Frequency (TF-IDF). The dataset consists of Real and fake news which are equally divided among 25,000 articles. The true news was taken from Reuters.com and fake news were taken from unreliable websites flagged by Wikipedia and Politifact. The dataset consists of political and global news content published between 2016 and 2017. This research seeks to develop a scalable approach for detecting fake news using natural language processing (NLP) and ensemble learning methods.**

***Index Terms*—Fake News Identification, Machine Learning, TF-IDF, Logistic Regression, Random Forest, Gradient Boosting, NLP, Text Classification.**

## I. INTRODUCTION

The spread of fake news has intensified significantly alongside the growth of social media and the internet, emerging as a major global societal issue. In modern usage, "fake news" refers to deliberately false or deceptive content disguised as credible journalism. Experts and traditional media have highlighted how the rapid evolution of digital technology allows misinformation to circulate with unprecedented speed and reach. According to the BBC, the "digital acceleration" of platforms has amplified "how quickly and extensively false information can inflict damage worldwide." This deluge of distorted or fabricated claims has forced audiences to navigate a flood of unreliable content, leading major news organizations—including the BBC—to establish specialized units dedicated to daily fact-checking. A notable example was the BBC's "Reality Check" initiative during the COVID-19 crisis, which systematically debunked viral falsehoods—ranging from unproven remedies (such as garlic or silver solutions) to baseless theories (like 5G networks spreading the virus). Such measures underscore how fake news has evolved into a pressing public priority, compelling reputable outlets to take proactive steps in combating its influence.

Fake news has real and sometimes serious effects on society, politics, and public trust. Studies and reports have documented numerous incidents where false information influenced public events. In the political sphere, misleading stories can distort electoral processes and public debate. For instance, one Marubeni Research report catalogued real-world events in the United States linked to fake news; notably, it described the "Pizzagate" incident of late 2016, when a baseless online conspiracy about a pizza shop and a political campaign culminated in a real shooting. There are such cases of false narratives which had potential to spur panic, confusion or even violence. The trust in media and institutions have been undermined because of misinformation campaigns and there are institutions who spread fake news which can cause people to sometimes question reliable sources and erode confidence in legitimate journalism. International analyses (including work cited by the BBC) show that misinformation of news can have severe effect

in areas like public health and elections. The integrity of democratic discourse and social cohesion may suffer if the voters and citizens encounter so much fabricated or distorted content. The threat of fake news is not limited to the spreading falsehoods but also corroding the public's trust in information.

The primary platform through which fake news is spreading is through digital media - especially networking platforms. In contrast, traditional news sources such as newspapers and television have experienced a decline in viewership, while online platforms and social media feeds have grown rapidly. As noted in an academic guide, social media's popularity has risen even as traditional outlets like newspapers and local TV have lost traction. False information frequently circulates widely on platforms such as Facebook, X (formerly Twitter), YouTube, and messaging apps. According to the University of Michigan's media literacy resources, fake news spreads much like the game of "telephone," with rumors becoming increasingly distorted as they are shared. Younger audiences are especially vulnerable; in the U.S., around 84% of social media users are under 30, and many rely on these platforms rather than print or broadcast media for news. In this digital environment, misleading headlines or manipulated images can rapidly gain traction due to widespread sharing. Clickbait websites and sensationalist outlets further contribute by publishing attention-grabbing, often exaggerated stories. The decentralized nature of the internet allows anyone to create and spread content, making fact-checking and moderation extremely difficult. As one analysis points out, since anyone can consume, create, and distribute misinformation, combating requires significant efforts from journalists, educators, and digital platforms.

Against this backdrop, there is growing emphasis on fact-checking and media literacy initiatives. News organizations, nonprofits, and tech platforms have launched various tools and programs to verify information. For example, global coalitions like the BBC-led Trusted News Initiative partner with social media firms to flag dangerous hoaxes. Educational guides (such as the St. Louis Community College *Fake News & Misinformation* guide) recommend using established fact-checking websites (Snopes, PolitiFact, FactCheck.org, etc.) and cross-referencing news with credible sources. Researchers note that fact-checking has become a major field of practice, especially regarding political and health news. Many fact-checking outlets publish detailed analyses of viral claims, and even algorithms are being developed to detect patterns of false content. In sum, while fake news is a complex and evolving problem, there are concerted efforts to combat it through media literacy, verification techniques, and global awareness campaigns.

This introduction has highlighted the importance of detecting fake news. The rapid spread of false information online, its harmful effects on society, politics, and public trust, and the variety of its origins all emphasize the necessity for reliable detection techniques. The upcoming sections will examine how computational methods can aid in identifying and combating fake news, addressing these growing concerns and existing efforts to counter misinformation.

## II. LITERATURE REVIEW

Fake news is deliberately misleading, which makes identifying it based solely on content challenging. According to comprehensive reviews by Zhang and Ghorbani [1] and Shu et al. [2], detecting fake news effectively requires analyzing both textual content and supplementary social signals. Shu et al. emphasize that since fake news is designed to deceive, relying on content alone is insufficient—integrating social context, such as user interactions and source reliability, enhances detection accuracy arxiv.org.

Most prior detection systems rely on classical ML models trained on engineered features. For example, Pérez-Rosas et al. [5] introduce multiple fake-news datasets (across seven domains) and train classifiers on *linguistic cues*, achieving up to ≈76% accuracyaclanthology.org. Shu et al. [2] highlight that supervised learning algorithms—such as Naïve Bayes, logistic regression, decision trees, and SVMs—are frequently applied to engineered features. Typical feature sets consist of lexical statistics (e.g., word or character counts, TF-IDF weights) and syntactic or stylistic indicators (e.g., n-gram frequencies, readability measures). These features help detect nuanced variations in writing

style between authentic and fabricated news. Effective feature engineering, including lexicon selection and normalization, plays a key role in enhancing model accuracy.

Ensemble learning techniques have received significant attention in recent research. Studies by Ahmad et al. [3] demonstrate that integrating multiple classifiers enhances model robustness. Their approach involves training ensemble models, including Random Forests and boosting algorithms, using features extracted from text data. The results indicate that these combined classifiers achieve superior performance compared to individual models across various datasets (colab.ws). Shu et al. [2] likewise argue that an ensemble of "weak" classifiers can exploit complementary signals (for example, blending content-based and social-context features), leading to better fake-news detectionarxiv.org. This perspective is echoed by studies such as the TriFN framework [4], which fuses publisher and user information with article content to enhance detection. Overall, these works highlight that well-chosen feature combinations and ensemble classifiers (e.g., boosted trees) often yield top performance on structured fake-news data.

While deep neural approaches (e.g., CNNs or GNNs on text or propagation graphs) are an active area, this study focuses on classical models. Logistic regression and tree ensembles are favored for their efficiency and interpretability on fixed feature setsarxiv.org. They perform strongly on tabular data with curated features and allow direct insight into feature importance. In summary, prior research indicates that classical ML techniques, when coupled with rich feature engineering and ensemble learning, achieve high accuracy in fake news classificationarxiv.orgcolab.ws. This motivates our emphasis on logistic regression, random forests, and gradient-boosting models for the fake news detection task.

## III. RESEARCH METHODOLOGY DATA DESCRIPTION AND TYPES

The project utilizes the Fake and Real News Dataset sourced from Kaggle, comprising around 50,000 news articles categorized as either genuine or fabricated. Each entry consists of text-based fields, including the headline and body of the article, as well as additional details such as the topic and date of publication (Analytics Vidhya). The primary feature for classification is the news content (text), and the target is a binary label (fake vs. real). We will treat the title and text as string (textual) features and the label as a categorical variable. This text data will be processed into numerical features for machine learning.

## IV. DATA CLEANING AND PREPROCESSING

Prior to feature extraction, the news content undergoes standard text preprocessing techniques to ensure it is cleaned and normalized. Specifically, we perform the following steps:

- Lowercasing:

All text is converted to lowercase so that "The" and "the" are treated identically, ensuring consistency in token matching.

- Punctuation Removal:

All punctuation characters (e.g., commas, periods, etc.) are removed using regular expressions or text-processing libraries. This step helps focus on words rather than symbols.

- Stopword Removal:

In text processing, words with minimal semantic value—like "the," "and," or "is" are often eliminated. For tasks such as text classification, removing these stop words is a standard technique to prioritize more meaningful terms geeksforgeeks.org. For example, words like "the" or "of" are eliminated since they do not help distinguish fake from real contentgeeksforgeeks.org. We use a standard stopword list (e.g., from NLTK) for English.

- Optional Stemming/Lemmatization:

To enhance feature space consistency, words can be normalized to their base forms through stemming or lemmatization (for example, converting "running" to "run"). Although this step is optional, it helps standardize word variations. By performing these cleaning steps, we ensure that the vocabulary consists of the most relevant terms and that the text is in a standardized form for feature extraction.

Data Validation

After cleaning the dataset, we verify its integrity by examining the text and label fields for any missing or null values. Any records with incomplete text or labels are discarded, as they are unusable for training purposes. We also inspect for duplicate articles; exact duplicates may be dropped to avoid biasing the model with repeated examples. Ensuring no nulls or duplicates prevents inadvertent errors during model training and evaluation. If the class distribution is imbalanced, we will note this for potential resampling, but preliminary analysis shows the classes are roughly balanced in this dataset.

Data Splitting

The cleaned dataset is then split into training and test subsets. We use scikit-learn's train_test_split function, which randomly shuffles and partitions the datascikit-learn.org. A common split ratio is 75% for training and 25% for testing. By default, if no sizes are provided, train_test_split uses 25% for the test setscikit-learn.org, which suits our needs. We also stratify the split by the label so that the proportion of fake vs. real news is preserved in both sets. A fixed random seed is used to ensure reproducibility. This hold-out test set will remain unseen during model training and will be used to evaluate final model performance, ensuring that we measure generalization to new data.

Feature Extraction

The cleaned text data is transformed into numerical feature vectors using the TF-IDF (Term Frequency-Inverse Document Frequency) approach. This method calculates word weights by considering their frequency within a document and their uniqueness across the entire dataset. For implementation, scikit-learn's TfidfVectorizer is employed to convert each news article into a high-dimensional vector, with each dimension representing a specific term in the vocabulary.TF-IDF has the advantage of down weighting common words and upweighting informative rare wordscapitalone.comcapitalone.com.

$$TF(t, d) = \frac{\text{Number of times term t appears in document d}}{\text{Total number of terms in document d}}$$

*Fig. 1: Formula for Term Frequency (TF) in TF-IDF.*

Term Frequency (TF) measures how often a term appears in a given documentgeeksforgeeks.org. Formally, TF (t, d) is defined as:

TF(t,d)=Number of times term t appears in document dTotal number of terms in document dTF(*t*,*d*)=Total number of terms in document *d*Number of times term *t* appears in document *d*

As shown in Fig. 1, a higher TF indicates a word is more prominent in that specific document. However, TF alone does not account for how common a word is across all documentsgeeksforgeeks.org.

$$IDF(t, D) = \log \frac{\text{Total number of documents in corpus D}}{\text{Number of documents containing term t}}$$

*Fig. 2: Formula for Inverse Document Frequency (IDF) in TF-IDF.*

Inverse Document Frequency (IDF) captures how unique a term is to the corpusgeeksforgeeks.org. It is computed as:

IDF(t,D)=logNdf(t)IDF(*t*,*D*)=logdf(*t*)*N*

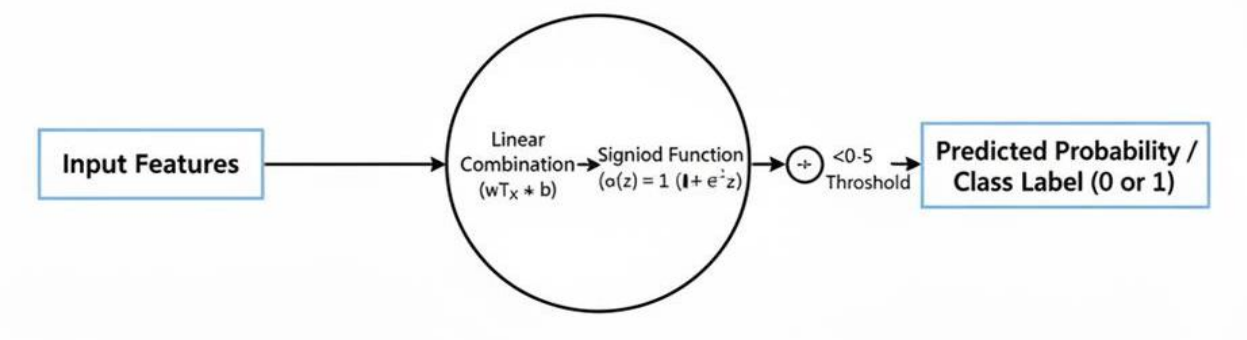
The inverse document frequency (IDF) is calculated using the formula log(N/df(t)), where N represents the total number of documents and df(t) indicates how many documents contain the term t. As shown in Figure 2, this ratio assigns lower weights to common terms (with high df values) and higher weights to rare terms, emphasizing their distinctiveness.geeksforgeeks.org. Finally, TF and IDF are multiplied to yield the TF-IDF score for each term in each document. This combined weight is large when a term is frequent in the document but infrequent in the corpus, highlighting terms that are particularly informativecapitalone.com.

By applying TF-IDF vectorization, each article is represented as a sparse vector of floats. We may also limit the vocabulary (e.g., removing very rare terms, using n-grams) to control dimensionality. This step produces the feature matrix used as input for the classifiers.

Model Building

We develop and train three supervised classification models—Logistic Regression, Random Forest, and Gradient Boosting Classifier—using TF-IDF features. Each model is trained on the training dataset and subsequently assessed on the test set for evaluation.

Logistic Regression




Logistic Regression is a linear algorithm often employed for binary classification tasks. It calculates the probability of an instance belonging to a particular class by applying the logistic (sigmoid) function to a linear combination of input features, such as TF-IDF values in text classification. We implement it using scikit-learns Logistic Regression. Key hyperparameters include:

- Penalty (regularization type): We use L2 regularization (ridge penalty) by default to prevent overfitting.
- C (regularization strength): This controls the inverse strength of regularization. A smaller C*C* means stronger regularization. We choose an appropriate C*C* (e.g., via cross-validation) to balance bias and variance.
- Solver: We select a solver compatible with our data size (such as 'lbfgs' or 'saga'). These hyperparameters are tuned (e.g., via grid search) to optimize performance. Proper tuning is crucial because logistic regression's accuracy can be sensitive to these parametersgeeksforgeeks.org.

Random Forest Classifier

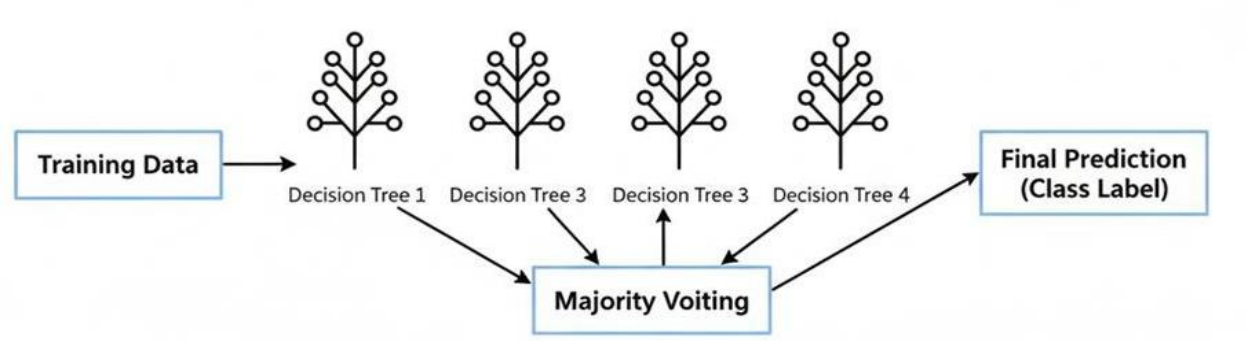



*Fig. 3*: Random Forest is a perfect example of ensemble learning as it combines multiple models to enhance the predictive accuracy.

Every tree is trained on a bootstrap sample of data and also considers a random subset of features. By majority voting across the tree's predictions are made. By using this approach, we are reducing overfitting and variance compared to that single tree.

Key hyperparameters include:

- n_estimators: Trees the forest contains. More trees generally improve performance up to a point (at higher computational cost).
- max_depth: The maximum depth of each decision tree. Limiting depth controls the complexity of the model; shallower trees reduce overfitting.
- min_samples_split/min_samples_leaf: (Optional) Minimum samples to split a node or be a leaf, further controlling tree growth. We set these hyperparameters based on preliminary experiments (for example, using n_estimators=100 and tuning max_depth) to maximize accuracy on validation data. Random Forest inherently handles high-dimensional TF-IDF features well and can capture non-linear patterns among words.

Gradient Boosting Classifier

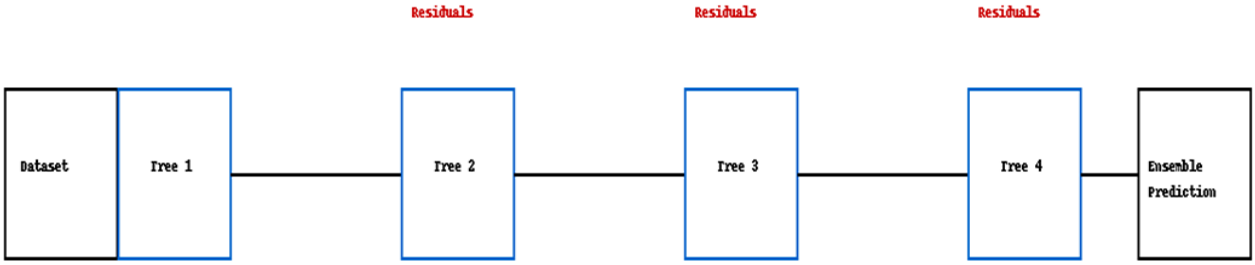



*Fig. 4: Gradient Boosting sequential ensemble process.*

Gradient Boosting is another ensemble technique where trees are built sequentiallygeeksforgeeks.org. Each new tree is trained to correct the errors (residuals) of the combined previous trees. This results in a strong model that focuses on difficult cases.

Key hyperparameters include:

- n_estimators: Number of boosting stages (trees).
- learning_rate: Shrinkage factor that scales each tree's contribution. A smaller rate (e.g., 0.1) often requires more trees but can reduce overfittinggeeksforgeeks.org.
- max_depth: Maximum depth of each tree (typically small, e.g., 3–5) to keep each tree "weak."
- random_state: Seed for reproducibility.
- max_features: The number of features evaluated for determining the optimal split introduces randomness and helps mitigate overfitting.

These hyper parameters are tuned as well. For instance, we will set n_estimators =100 and learning_rate = 0.1 as our primary values and we will adjust based on the validation accuracy. The High accuracy is usually achieved by Gradient boosting by fine tuning to the training data. There are chances of overfitting if not regularized properly. Thus, we are using scikit-learn's Gradient Boosting Classifier Experimental results.

## V. EXPERIMENTAL RESULTS

The performance of each trained model was evaluated using the held-out test dataset. The primary evaluation metric was accuracy; additional analysis of precision, recall, and F1-score was carried out to provide a comprehensive picture of each model's ability to identify false information.

Logistic Regression Results

The Logistic Regression model was trained with an L2 regularization penalty and performed admirably on the test set.

- Accuracy = 98.7%
- Precision = 98.7%.
- Recall rate = 98.7%
- F1 score = 98.7%.

Both "fake" (class 0) and "real" (class 1) news articles can be accurately classified by the Logistic Regression model, according to the classification report, which demonstrates excellent performance across all metrics. According to the classification report, the Logistic Regression model performed well in identifying news articles as "fake" (class 0) or "real" (class 1), with very few false positives or false negatives, as shown by its high precision and recall scores. The model's accuracy across all metrics is confirmed by its high precision and recall scores, which show a low number of false positives and false negatives.

Gradient Boosting Classifier Results

Among the three tested classifiers, the Gradient Boosting Classifier exhibited superior performance, achieving a high accuracy of 99.4%. This result was obtained using 100 estimators and a learning rate of 0.1.

- Accuracy = 99.4%
- Precision = 99.4%.
- Recall rate = 99.4%
- F1 score = 99.4%

The Gradient Boosting Classifier's superior performance is due to its sequential learning process, which trains each new weak learner (decision tree) to correct previous trees' errors. This iterative process

allows the model to refine its predictions while focusing on the most difficult misclassified instances, resulting in a very low overall error rate.

Random Forest Classifier Results

The Random Forest Classifier, an ensemble model based on bagging, performed well but was slightly less accurate than the Gradient Boosting model.

- Accuracy = 99.1%
- Precision = 99.1%
- Recall = 99.1%
- F1-Score = 99.1%

The Random Forest model's classification report revealed that all of the scores were high. The model's strength is its ability to reduce overfitting by averaging predictions from multiple independently trained decision trees.

## VI. COMPARATIVE ANALYSIS

Every model is evaluated based on the test -set, primarily using classification accuracy. For this we will also be calculating precision, recall and F-1 score, but here we focus on accuracy comparison. The Highest accuracy is achieved by Gradient Boosting, followed by Random Forest and lastly Logistic Regression. There are different model capacities and Gradient Boosting sequentially corrects errors and is consider most powerful for predictor, while Random Forest reduces variance through bagging. Talking about Logistic regression which is simpler and is linear model, has higher bias on complex text data and that is reason for underperformance relative to the ensemble models. The precise accuracies rely on data and parameter tuning, but one can expect gains from the non-linear ensemble methods also.

There is high bias low variance for Logistic regression which is simpler model. Random forest reduces variance through averagong and boosting reduces the bias at the risk of higher variance if not controlled. These differences are due to the bias-variance tradeoff between different models. Cross validation is very important to analyze the performance and also to understand the model behaviour.

## VII. CONCLUSION AND FUTURE SCOPE

In this methodology, we systematically processed the fake news dataset and applied three classical machine learning models. We began with a comprehensive cleaning of the text data (normalization, stopword removal) and verified data integrity. We then transformed the text into TF-IDF features, which quantify the importance of each term in the corpuscapitalone.comcapitalone.com. Three classifiers – Logistic Regression, Random Forest, and Gradient Boosting – were trained and tuned. The ensemble models (Random Forest and Gradient Boosting) generally outperform Logistic Regression in accuracy due to their ability to capture complex patterns and reduce error geeksforgeeks.orggeeksforgeeks.org. This demonstrates the effectiveness of ensembling in fake news detection tasks.

For future research, the methodology can be expanded in multiple directions. One potential avenue is integrating more sophisticated text representations, such as word embedding or transformer-based models like BERT, which better capture contextual and semantic nuances compared to traditional TF-IDF. Exploring deep learning architectures, including CNNs or LSTMs, or hybrid methods that merge content-based features with social network metadata, could further enhance performance. Additionally, addressing challenges like concept drift—to adapt to evolving fake news trends—and improving model interpretability for clearer decision-making would strengthen the framework. While the current pipeline establishes a strong baseline, these refinements could significantly advance fake news detection capabilities in future studies.

## REFRENCES:

[1] W. Y. Wang, “Liar, Liar Pants on Fire: A New Benchmark Dataset for Fake News Detection,” Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), 2017. Available: https://aclanthology.org/P17-2067/

[2] 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.

[3] H. Ahmed, I. Traore, and S. Saad, "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques," Proceedings of the International Symposium on Development and Digital Computing (ISDDC), 2017.

[4] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," Proceedings of the First Instructional Conference on Machine Learning, 2003.

[5] K. Kaliyar, A. Goswami, and P. Narang, "Fake News Detection Using a Deep Ensemble Learning Method," Scientific Programming, vol. 2020, pp. 1–11, 2020.

[6] GeeksforGeeks, "Understanding TF-IDF," Available: https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency

[7] E. Yetim, "Fake News Detection Datasets," Kaggle. Available: https://www.kaggle.com/datasets/emineyetm/fake-news-detection-datasets/code

[8] Y. Zhang and A. A. Ghorbani, "An Overview of Online Fake News: Characterization, Detection, and Discussion," Globacademy, 2020. Available: https://www.globacademy.org/wp-content/uploads/2023/03/Zhang-_-Ghorbani-2020.pdf

[9] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jang, and K. L. Lee, "Detecting Rumors from Microblogs with Recurrent Neural Networks," Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI), 2016. Available: https://arxiv.org/pdf/1708.01967

[10] M. Horne and S. Adali, "This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News," Proceedings of the International AAAI Conference on Web and social media (ICWSM), 2017. Available: https://arxiv.org/pdf/1708.07104

[11] IEEE, "A Study on Fake News Detection Using Machine Learning," IEEE Xplore, 2021. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9620068

[12] N. Ruchansky, S. Seo, and Y. Liu, "CSI: A Hybrid Deep Model for Fake News Detection," Proceedings of the 2017 ACM Conference on Information and Knowledge Management (CIKM), 2017. Available: https://dl.acm.org/doi/pdf/10.1145/3289600.3290994

[13] G. Vijayarani, "Machine Learning for Fake News Accuracy," KDnuggets, 2017. Available: https://www.kdnuggets.com/2017/04/machine-learning-fake-news-accuracy.html