Fake News Detection and Analysis using Machine Learning Algorithms

1st Akshith Madugula
Computer Science
Cal Poly Pomona
Pomona, USA
armadugula@cpp.edu

2nd Dhruv Patel
Computer Science
Cal Poly Pomona
Pomona, USA
dhruvpatel1@cpp.edu

3rd Lahari Sandepudi
Computer Science
Cal Poly Pomona
Pomona, USA
lsandepudi@cpp.edu

4th Ruchitha Gande
Computer Science
Cal Poly Pomona
Pomona, USA
rgande@cpp.edu

***Abstract*—The prevalence of fake news poses a significant threat to the integrity of information in the digital age. This research project introduces a comprehensive approach for detecting fake news, utilizing Python and an array of machine learning algorithms. The system incorporates meticulous text preprocessing, robust feature engineering, and thorough model construction. By employing algorithms such as k-nearest neighbor (k-NN), random forest (RF), Decision Tree, Gradient Boosting, Passive Aggressive., we seek to identify the most effective model in accurately discriminating between authentic and fabricated news articles. This project addresses the urgent need for reliable source verification and demonstrates the power of machine learning in enhancing detection accuracy.**

***Index Terms*—fake news, detection, machine learning, text preprocessing, feature engineering**

I. Introduction

The prevalence of fake news in today's digital age poses a significant threat to the integrity of information and public discourse. With the rapid advancement of online platforms and social media, the dissemination of misleading and fabricated information has become increasingly pervasive. This phenomenon not only undermines trust in traditional news sources but also challenges the very fabric of democratic societies by influencing public opinion and decision-making processes.This research project introduces a comprehensive approach for detecting fake news using machine learning and deep learning algorithms. The study utilizes Python for implementation and explores a range of sophisticated techniques, including meticulous text preprocessing, robust feature engineering, and model construction. By leveraging algorithms such as k-nearest neighbor (k-NN), random forest (RF), Decision Tree, Gradient Boosting, and Passive Aggressive, the aim is to identify the most effective model for accurately discriminating between authentic and fabricated news articles.The motivation for this research is driven by the urgent need for reliable source verification in the face of widespread misinformation. This project seeks to contribute to the development of robust detection mechanisms that can enhance media literacy and empower individuals to critically evaluate news sources. By automating the fake news detection process, this study aims to safeguard the integrity of online information and promote a more informed digital landscape.

The methodology employed in this study is structured into three key stages. Firstly, data preprocessing techniques are applied to clean and filter the dataset, including stopword removal, special character replacement, and number normalization. Secondly, advanced feature extraction techniques such as TF-IDF vectorization are used to transform textual data into numerical feature vectors that capture the importance of terms within each document. Lastly, a variety of machine learning and deep learning classifiers are implemented and evaluated to classify news articles as either genuine or fake. This research utilizes a dataset sourced from Kaggle, which contains a balanced distribution of real and fake news articles.The dataset is related to past elections in 2018, ensuring relevance to contemporary societal issues and political contexts. Initial results demonstrate promising accuracy rates across various classifiers, with models such as Gradient Boosting and Passive Aggressive showing superior performance in distinguishing between authentic and fabricated news.

This research project contributes to the growing field of fake news detection by providing a systematic approach that integrates machine learning and NLP techniques. By developing effective detection models, this study aims to mitigate the harmful effects of misinformation and uphold the credibility of online information sources.

II. DataSet

A. Links

https://www.kaggle.com/datasets/hassanamin/textdb3

B. Observations in Data

- This dataset contains 6335 news article samples with no missing values, split into 4 columns.
- The data set used is past elections articles data.
- The target column, "label," designates articles as "REAL" (3171 samples) or "FAKE" (3164 samples), offering a balanced distribution for classification.
- Additionally, the dataset includes "title" and "text" columns containing textual content of news articles. Lastly, an unnamed numeric column exists, for ID
- The textual features will likely necessitate preprocessing and feature engineering techniques for effective use in machine learning models.

III. Related Work

Recent advancements in fake news detection have leveraged a variety of machine learning and deep learning techniques to improve accuracy and robustness. Suryawanshi et al. (2021) conducted a comprehensive study using machine learning algorithms such as Support Vector Machines (SVM) and Random Forest, achieving accuracies of up to 92%, demonstrating significant improvements through feature engineering and optimization. Kaliyar et al. (2023) emphasized the effectiveness of deep learning models, particularly Long Short-Term Memory (LSTM) networks, which excel at capturing temporal dependencies in textual data and outperform traditional methods. Additionally, the work by Granik and Mesyura (2017) utilized linguistic analysis and machine learning, achieving an accuracy of 89%, showcasing the importance of textual features in differentiating fake news from real news. These studies collectively highlight the potential of integrating machine learning and deep learning approaches to address the multifaceted nature of fake news, though challenges remain in achieving scalability and maintaining robustness across diverse datasets and rapidly evolving deceptive strategies.

IV. Methodology

The methodology employed in this study was divided into three distinct stages. Initially, in the pre-processing stage, various data filtering and cleaning techniques were applied to extract semantic features from the raw dataset. These techniques included implementing a stopword filter, replacing special characters, normalizing numbers, and decontracting words. Furthermore, HTML tags were removed using advanced filtering technologies to eliminate impurities that could hinder effective classification.In the second stage, numerical features were extracted from the semantic features to create feature vectors. Finally, in the third stage, machine learning and deep learning classifiers were applied to group items in the dataset. Each of these methods was separately applied to the same dataset to determine the most effective algorithm with high accuracy for classifying news articles as genuine or fabricated. Details of these three stages are elaborated in Figure 1.

A. Data Preprocessing

In the data preprocessing step, the text was cleaned and prepared for Natural Language Processing (NLP). This involved several tasks such as text normalization, stop words removal, tokenization, and lemmatization. The unstructured data contained numerous impurities, including HTML tags, special characters, and punctuation, which made the process of transforming textual data into natural language very challenging. Various techniques were employed to convert this unstructured data into a structured format that a machine can process.Stopword removal is a common technique used in data filtering, information retrieval, and text classification. It eliminates certain unwanted words (e.g., ”the,” ”in,” ”a,” ”an,” ”with,” etc.) that do not contribute to the categorization process. To clean the dataset, the Python Standard Library was utilized to remove HTML tags, employing the BeautifulSoup function. Additionally, the compile function was used to eliminate punctuation. Finally, the preprocess text function was applied to remove un-necessary characters.

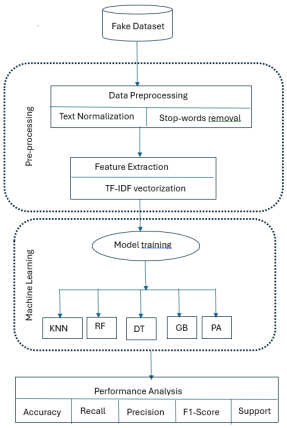

Fig. 1.

B. Features Extraction

Before training the ML models, we need to extract the important words that give the best performance. This step is also known as features extraction because words are the features to be used to train the model. Extracting features is a method in which text data are converted into Vectors 0 and 1, and new vectors were created from the sample text file. Three features extraction techniques which is used in this study is Term Frequency-Inverse Document Frequency (TF-IDF) Vectorizer. After cleaning the data, each features extraction technique was performed on the data. Term Frequency-Inverse Document Frequency (TF-IDF vectorizer) One of the most widely used feature extraction techniques is the term frequency-inverse document frequency (TF-IDF) . This technique is divided into two stages, in which the term frequency (TF) is calculated first, and the inverse document frequency (IDF) is calculated in the second stage. This is done by finding the term frequency (TF) then multiplying it by the inverse document frequency (IDF). This technique shows the most related words to each document by assigning them high values, while the words that are repeated in all the documents will have low values even if they were frequently used.

$$\text{TF-IDF}_t = \text{TF}_t \times \text{IDF}_t = \frac{n}{k} \times \log \frac{D}{D_t}$$

where:

- TF_t is the number of appearances of term t in the document n divided by the total number k of terms in the document, keeping the multiplicity of each term.
- IDF_t is the total number of documents D divided by the number D_t of documents citing this term.

C. Classifiers stage

The following classifiers were used in this study: k-nearest neighbor (k-NN), random forest (RF),Decision Tree, Gradient Boosting, Passive Aggressive.

- **K-nearest neighbor (K-NN):** This is an algorithm characterized by simplicity and ease of implementation, derived from supervised machine learning algorithms, which were first used in the early 1970s to solve classification and regression problems. This algorithm is based on the principle of similarity between neighbors. Since the principle of similarity depends on the value of K, cases are classified by the majority of the votes of their neighbors. This occurs because similar cases are close to each other.
- **Random forest (RF):** This is considered one of the key classifiers in supervised machine learning, as it consists of a group of decision tree classifiers, whereby samples are taken randomly for each decision tree, and each decision tree votes for the most popular category of input vector classification.
- **Decision Tree (DT):** A Decision Tree is a popular supervised learning method used for both classification and regression tasks. It works by recursively partitioning the data into subsets based on the values of input features. Each subset is split further until a stopping criterion is met, such as a maximum tree depth or no further improvement in purity. Decision Trees are intuitive and easy to interpret, capable of handling both numerical and categorical data. However, they can be prone to overfitting, especially when the trees become overly complex.
- **Gradient Boosting (GB):** Gradient Boosting is an ensemble learning method that sequentially combines the predictions of several base estimators, typically decision trees. It builds the model in a stage-wise fashion, where each new tree helps correct errors made by the previously trained trees. This method generalizes by optimizing an arbitrary differentiable loss function. Gradient Boosting is renowned for its high predictive accuracy and is less prone to overfitting compared to single Decision Trees.
- **Passive Aggressive (PA):** Passive Aggressive (PA) is a family of online learning algorithms used primarily for classification and regression tasks. These algorithms are particularly useful when the amount of data is large and it is not feasible to keep the entire dataset in memory. The name ”Passive Aggressive” reflects its approach: it remains passive when predictions are correct and becomes aggressive (i.e., updates the model) when predictions are incorrect. PA algorithms are efficient and can adapt to changing data streams, making them suitable for scenarios where data arrives sequentially or when memory resources are limited.

V. Results

While analyzing the data, we found that our target class, which indicates the type of news, is well-balanced. The distribution shows that REAL news accounts for **50.1%** and FAKE news makes up **49.9%**, as illustrated in Fig. 2. Therefore, the dataset has an almost equal number of data points for each class. For consistency in our analysis, we converted the labels by changing REAL to ’1’ and FAKE to ’0’.

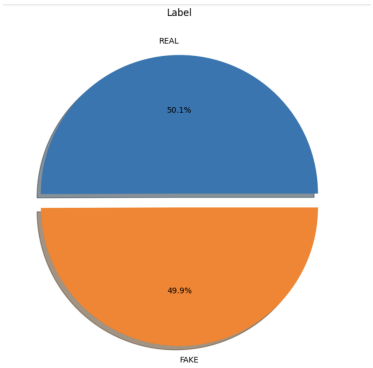

Fig. 2. Class Distribution ”Label”

For the column initially labeled ”unnamed: 0,” which contains unique identifier values, we observed a pattern: where identifiers less than 5500 correspond to REAL news, while identifiers greater than 5500 correspond to FAKE news. As part of the preprocessing steps, we renamed ”unnamed: 0” to ”ID” to enhance clarity in our analysis.

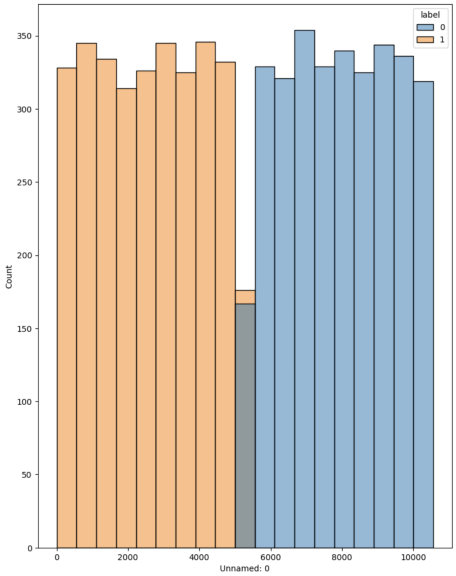

Fig. 3. Observation on Unnamed Column

For the title column, which consists of character strings, we found that FAKE news titles typically range from 50 to 100 characters, while REAL news titles are strictly between 40 and 80 characters as in Fig. 4. Titles exceeding 110 characters are classified as FAKE news. In the preprocessing step, we removed duplicate titles to ensure data integrity.

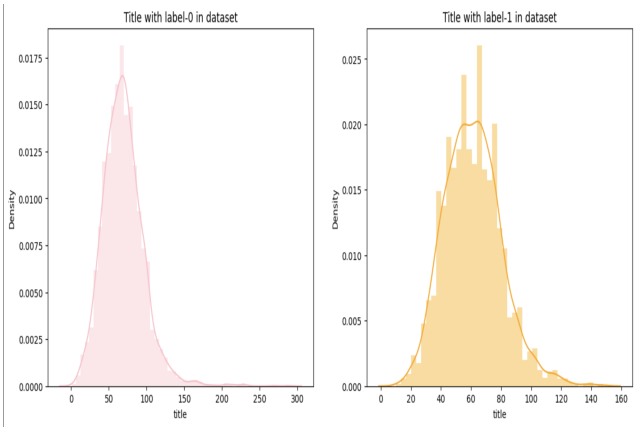

Fig. 4. Title class observation

For the column named "text," we preprocessed the data by converting symbols and signs into words, such as changing '%' to 'percent'. From this, we observed that most REAL news articles have a character length between 0 to 10,000 and should be classified as true. In contrast, longer articles are typically classified as FAKE, as shown in Fig 5.

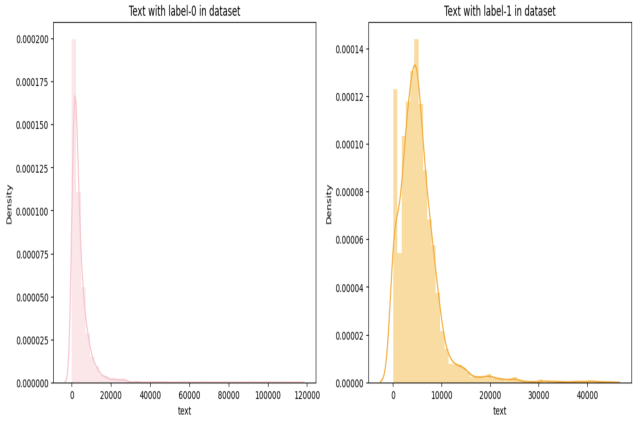

Fig. 5. Text class analysis

we applied the K-Nearest Neighbour algorithm (with K=1), which resulted in a slightly higher accuracy of 83.03%, though it had a much longer runtime of 764.65 seconds (approximately 12.5 minutes). For faster results and higher accuracy, we used the Random Forest algorithm, achieving an accuracy of 91.08% with a significantly reduced training time of 15.72 seconds. Additionally, we explored Gradient Boosting and Passive Aggressive algorithms, achieving accuracies of 93% and 94.47%, respectively, both of which outperformed the other models.. For further details, you can refer to the project's codebase via this **[code link]** and the LaTeX document via this**[LaTeX link]**

VI. Conclusion

In conclusion, the algorithms applied to our dataset achieved an accuracy of 94.47% using the Passive Aggressive algorithm. The Gradient Boosting algorithm also performed similarly, with an accuracy of 93%, making both models result in better accuracy among others. Looking forward, we plan to implement and train models on a newer dataset, as our current dataset, sourced from Kaggle, dates to 2018.

VII. Future Work

Future work in fake news detection holds significant potential for advancing the field's effectiveness, particularly in anticipation of upcoming election cycles. Adapting existing algorithms such as Passive Aggressive to the unique dynamics of forthcoming elections is paramount. This adaptation involves not only refining the model's feature set but also integrating election-specific factors such as candidate mentions, sentiment analysis of political discourse, and real-time monitoring of emerging narratives. Additionally, transitioning from word-level to character-level vectorization techniques represents a promising avenue for future research. By analyzing text at the character level rather than the word level, models can capture finer linguistic nuances, handle misspellings and slang more effectively, and improve robustness to out-of-vocabulary words. This shift towards character-level vectorization has the potential to refine fake news detection systems, making them more adept at identifying deceptive content across diverse linguistic contexts and evolving language usage patterns, particularly during election periods.

References

[1] Meital Balmas. When fake news becomes real: Combined exposure to multiple news sources and political attitudes of inefficacy, alienation, and cynicism. Communication Research, 41(3):430–454, 2014.

[2] Conroy, N. J., Rubin, V. L., & Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. Proceedings of the Association for Information Science and Technology, 52(1), 1-4.

[3] Karimi, F., & Wang, J. (2018). Fake News Detection Using Data Mining Techniques: A Literature Review. 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 423-426.

[4] Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2019). Beyond News Contents: The Role of Social Context for Fake News Detection. IEEE Transactions on Computational Social Systems, 6(3), 470-480. DOI: 10.1109/TCSS.2019.2907483

[5] Pérez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2018). Automatic Detection of Fake News. Proceedings of the 27th International Conference on Computational Linguistics, 3391-3401. DOI: 10.18653/v1/N18-1170

[6] Yang, K., Chen, B., & Zhu, J. J. H. (2020). Fighting the COVID-19 Infodemic: Modeling the Perspective of Journalists, Fact-Checkers, Social Media Platforms, Policy Makers, and the Society. Human Behavior and Emerging Technologies, 2(3), 251-261. DOI: 10.1002/hbe2.204

[7] Ruchansky, N., Seo, S., & Liu, Y. (2017). CSI: A Hybrid Deep Model for Fake News Detection. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 797-806. DOI: 10.1145/3132847.3132967

[8] Horne, B. D., & Adali, S. (2017). 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 2017 ACM on Web Science Conference, 91-99. DOI: 10.1145/3091478.3091480

[9] Wang, W. Y. (2017). "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, 422-426. DOI: 10.18653/v1/P17-1044

[10] Castillo, C., Mendoza, M., & Poblete, B. (2011). Information Credibility on Twitter. Proceedings of the 20th International Conference on World Wide Web, 675-684. DOI: 10.1145/1963405.1963500

[11] Vlachos, A., & Riedel, S. (2014). Fact checking: Task definition and dataset construction. Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science, 18-22. DOI: 10.3115/v1/W14-1803
[12] Wang, W. Y. (2018). "Evaluating the Credibility of Fake News Articles by Focusing on the Content and Structure". International Conference on Web and Social Media (ICWSM), 280-289.
[13] Castillo, C., Mendoza, M., & Poblete, B. (2013). Predicting Information Credibility in Time-Sensitive Social Media. Internet Research, 23(5), 560-588. DOI: 10.1108/IntR-06-2013-0115
[14] Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., & Stein, B. (2017). A Stylometric Inquiry into Hyperpartisan and Fake News. Proceedings of the 2017 ACM on Web Science Conference, 89-90. DOI: 10.1145/3091478.3091500
[15] MidouAZERTY. Detecting Fake News Step by Step. Kaggle. Retrieved from https://www.kaggle.com/code/midouazerty/detecting-fake-news-step-by-step/notebook
[16] Yuushii. LSTM-AA. Kaggle. Retrieved from https://www.kaggle.com/code/yuushii/lstm-aa
[17] Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2021). Fake News Detection Using Machine Learning and Deep Learning Algorithms. Retrieved from https://www.researchgate.net/publication/352021652_Fake_News_Detection_Using_Machine_Learning_and_Deep_Learning_Algorithms

VIII. Supplementary

Code link : https://colab.research.google.com/drive/1oBKNKwxymxdC8NCoJPJi_S-rkXvl84n_?usp=sharing

Latex file link :https://www.overleaf.com/read/yxqdvddxhbxt#565259