A Comparative Analysis of Text Embeddings (TF-IDF, Word2Vec, Fast Text) for Machine Learning-Based Fake News Detection

Lakshmi Kalyani, Mary Jacintha,[3] Debjit Kar, Notan Roy, V.K. Sharma

[1]Scientist F,[2]Scientist F,[3]Scientist C, Scientist E, Scientist G

[1]Education & Training,

[1]Centre For Development of Advanced Computing, Noida, India

Abstract - Robust and dependable detection techniques are necessary to stop the spread of fraudulent information on internet platforms. This study compares and examines the efficacy of various embedding algorithms (TF-IDF, Word2Vec, and FastText) in terms of misleading information. We utilize these inputs to train various machine learning models, such as Random Forest, Decision Tree, Support Vector Machine (SVM), Logistic Regression, and XGBoost, and assess how well they distinguish between genuine and misleading information. Our findings demonstrate the great accuracy achieved by TF-IDF integration. The accuracies of the machine learning models are as follows: Random Forest - 99.16 percent; Decision Trees - 99.04 percent; Word2Vec, although it catches word connections in the text better, has an accuracy of 94.01 percent; and finally, FastText - 94.88 percent. Clearly, the identification of the advantages and disadvantages of each integration technique is crucial, meaning that looking at the confusion matrix is mandatory. With this, pertinent assistance is observable for both academics and developers working on false information detection systems.

Index Terms – Term Frequency-Inverse Document Frequency (TF-IDF), Word2Vec, Fast Text, Machine Learning Models, Support Vector Machine (SVM), False Information Detection

I. INTRODUCTION

An era of extraordinary access to information has been brought about by the digital revolution, with which a wealth of information on almost every subject imaginable is available at our fingers. Others may now study and communicate with others throughout the world more easily thanks to the democratization of knowledge, which promotes cross-cultural cooperation. However, this abundance also brings with it a big problem: fake news—a term used to describe the widespread dissemination of false information.

The harm posed by fake news to society is complex to say the least, due to the fact that it restricts productive public conversation. This has the potential to undermine confidence in reputable news organizations, which are essential to a functioning democracy. Furthermore, the widespread prevalence of fake news has the ability to influence public opinion on important problems, which could have far-reaching effects on decision-making processes. For instance, fake news about political candidates has the power to affect election results. Another example is the false information about healthcare procedures that might cause people to make decisions risking their health.

Fake news detection has become more urgent as individuals become more conscious of the possibility of manipulation in the online information environment, and algorithms for machine learning (ML) have become a viable weapon in this conflict. All these techniques are advanced computer programmes trained on large-scale labeled text datasets, which can now recognize the patterns and characteristics that set authentic news stories apart from fake ones due to this training.

Converting textual data into a format that computers can comprehend is a key step in determining whether such algorithms are effective. Here's where using embedding techniques becomes useful! Consider these methods as bridges between machine cognition and human language, essentially, as translators. They convert individual letters and sentences into numerical vectors. These vectors capture the meaning of the words as well as the nuances and small-scale semantic correlations found in the text. Consequently, embedding techniques enable machine learning algorithms to analyze large amounts of textual data and detect the subtle linguistic patterns that distinguish between authentic reporting and fake information. This is done by capturing the substance of the language in a format that computers can easily interpret.

This study examines the effectiveness of three well-known embedding methods for false news detection: Word2Vec, FastText, and TF-IDF and seeks to offer a thorough and exacting comparison of their performance. The research aims to shed light on each technique's practical use in the battle against disinformation by comparing its capacity to differentiate between authentic and fraudulent news pieces as well as its limits. The ultimate objective is to further the current conversation in the domains of computational linguistics and false news identification. This research aims to influence future developments in ML/DL-based systems by giving empirical data on the effects of various embedding methodologies, which will pave the way for more effective solutions to stop the spread of fake news. This, in turn, will provide a more trustworthy online information landscape.

II. LITERATURE SURVEY

The recent research in natural language processing (NLP) explores various techniques to improve text classification tasks in a copious number of areas. One such area is focused on word embeddings representing words as vectors to capture semantic meaning. For example, a study by (Pritom Mojumder, 2020) investigates FastText word embeddings for Bangla document classification, achieving high accuracy without complex pre-processing steps. Additionally, (Khasanah, 2021) conducted a study exploring the use of FastText embeddings in sentiment analysis, demonstrating competitive performance with simpler deep learning models like CNNs compared to more complex architectures. Another area of research examines different feature weighting methods for text classification. A study by (Mamata Das, 2020) investigates TF-IDF (Term Frequency-Inverse Document Frequency) compared to N-grams for sentiment analysis; their findings suggest that TF-IDF weighting leads to superior feature extraction, particularly when combined with the Random Forest classifier. Finally, studies like (Derry Jatnika, 2019) explore the effectiveness of word embedding models like Word2Vec in capturing semantic similarities, which highlights the strengths and weaknesses of various techniques depending on the specific NLP task.

III. METHODOLOGY

Our research utilizes a balanced dataset of real and fake news articles. After text pre-processing (missing value handling, tokenization, stop-word removal, stemming), we explore three feature engineering techniques: content creation, TF-IDF vectorization, and word embeddings (Word2Vec & FastText). Following that, we evaluate several machine learning models (Random Forest, Decision Tree, SVM, Logistic Regression, XGBoost) trained with each embedding technique to determine the most effective approach for fake news detection.

Dataset Acquisition and Pre-processing: The dataset encompasses articles categorized as either real or fake news, sourced from diverse origins. Articles considered authentic were gathered by scraping Reuters.com, a reputable news website, whereas articles considered fake were sourced by PolitiFact and Wikipedia. Covering a range of topics, the majority of them are centered around political and global news, as shown in table 1. Comprising two CSV files, "True.csv" contains over 12,600 articles sourced from Reuter.com, while "Fake.csv" includes an equivalent number from various fake news outlets. For each article, details such as title, text, type, and publication date were provided. To align with data from kaggle.com, emphasis was placed on collecting articles primarily from 2016 to 2017. While the dataset underwent cleaning and processing, the original text from fake news articles retained punctuation and errors.

News	Size (Number of Articles)	Subjects	
Real-News	21,417	**Type**	**Article Size**
		World-News	10145
		Political-News	11272
Fake-News	23,481	**Type**	**Article Size**
		Government News	1570
		Middle East	778
		US News	783
		Left News	4459
		Politics	6841
		General News	9050

Table1 Datasets of real & fake news

Data Pre-processing: Prior to analysis, the dataset underwent preprocessing steps to handle missing values and clean the text data, as mentioned previously. Missing values were replaced with empty strings; and text data was processed to remove non-alphabetic characters, converted text to lowercase, tokenized sentences, removed stop-words, and performed stemming using the Porter stemming algorithm.

Feature Engineering: The title and author information from each news article were combined to create a unified content feature, representing the textual information used for classification.

Word Embeddings: This research investigates three methods for converting text data into numerical vectors for machine learning models: TF-IDF Vectorization, Word2Vec Embeddings, and FastText Embeddings. TF-IDF (Term Frequency-Inverse Document Frequency) considers both a word's frequency within a document and its rarity across the dataset, implemented using scikit-learn's Tf-Idf Vectorizer with optimized features via grid search. In addition, Word2Vec creates dense vector representations for words in a continuous space, trained on the pre-processed data with a vector size of 300 dimensions and specific window and minimum word count parameters. Finally, FastText Embeddings, an extension of Word2Vec, tackles limitations with rare words and sub-word information, employing similar training parameters to Word2Vec for a direct comparison of their effectiveness in fake news detection.

Model Training and Evaluation: In order to investigate the most effective embedding technique for fake news detection, this research will employ several machine learning models, including Random Forest, Decision Tree, Support Vector Machine (SVM), Logistic Regression, and XGBoost. Each model will be trained on datasets utilizing three different embedding techniques: TF-IDF vectors, Word2Vec embeddings, and FastText embeddings. To ensure optimal performance, grid search with cross-validation will be used to fine-tune the models' hyperparameters. Finally, the effectiveness of each combination of model and embedding technique will be assessed based on accuracy scores calculated using a dedicated test dataset.

Performance Evaluation: To evaluate the effectiveness of each embedding technique (Word2Vec, FastText, and TF-IDF) in detecting fake news, the research will compare the accuracy scores achieved by machine learning models trained with each technique. Also, confusion matrices will be generated for each model to provide a more detailed picture of their performance. Essentially, confusion matrices categorize the model's predictions into true positives (correctly identified real news), false positives (mistakenly classified fake

news as real), true negatives (correctly identified fake news), and false negatives (mistakenly classified real news as fake). This comprehensive analysis will shed light on the strengths and weaknesses of each embedding approach in the context of fake news detection

IV. RESULTS

This section explores the performance of various machine learning models trained with different text embedding techniques for fake news detection. We analyse the effectiveness of TF-IDF, Word2Vec, and FastText embeddings, comparing their impact on model accuracy and efficiency. The results shed light on the strengths and weaknesses of each approach, along with insights into the characteristics of fake news that different embedding techniques are best suited to capture.

Embedding Technique Comparison: **Model Performance Analysis**

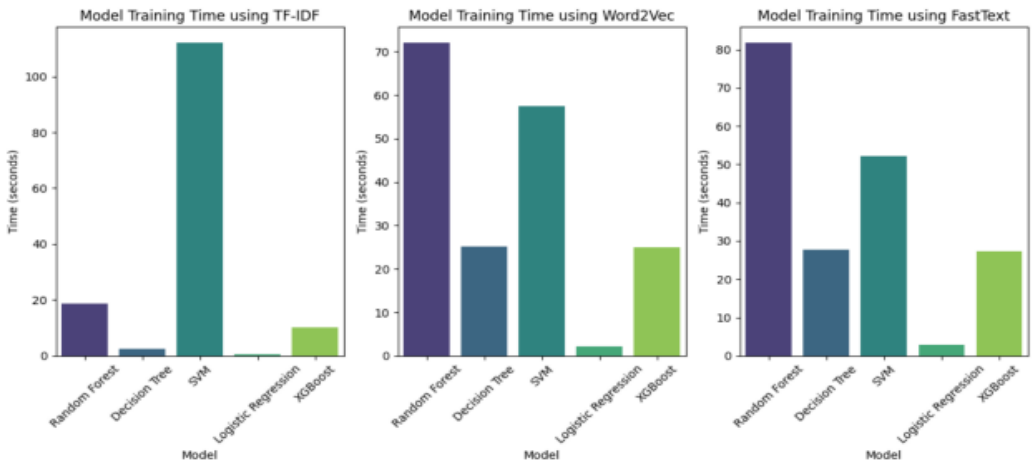
The analysis revealed a trade-off between embedding techniques and model accuracy. While TF-IDF embeddings led to the highest accuracy across all classifiers, with Random Forest reaching an impressive 99.16%, these embeddings might not capture the full range of semantic relationships in text. Word2Vec embeddings, though achieving slightly lower accuracy (Random Forest at 94.01%), offered advantages in capturing semantic nuances. FastText embeddings, building upon Word2Vec by incorporating subword information, provided a balance between the two. While not surpassing TF-IDF's peak accuracy (Random Forest at 94.88%), FastText embeddings yielded better performance than Word2Vec, particularly for SVM (93.61% vs 91.49%). Logistic Regression consistently displayed the lowest accuracy across all embedding techniques (ranging from 97.38% to 93.29%). This analysis highlights the importance of considering both training speed (discussed previously) and desired accuracy level when selecting an embedding technique for text classification tasks. This information is presented in Table 2.

Classifier	TF-IDF Accuracy	Word2Vec Accuracy	Fast Text Accuracy
Random Forest	99.16%	94.01	94.88
Decision Tree	99.04	91.49	91.51
SVM	98.56	91.92	93.61
XGBoost	98.51	94.28	95.55
Logistic Regression	97.38	91.51	93.29

Table 2: Model performance analysis

Model Training Time with Different Embedding Techniques

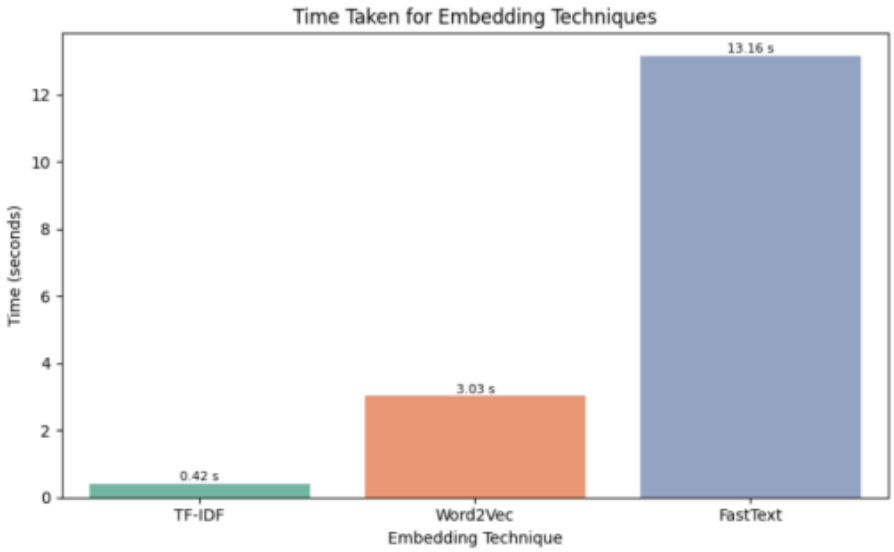
As shown in figure 1, the choice of embedding technique significantly impacts the training time of machine learning models for text classification tasks. TF-IDF emerges as the fastest option, with training times ranging from approximately 20 seconds for Logistic Regression to 60 seconds for XGBoost. Word2Vec embeddings, This efficiency likely stems from TF-IDF's use of sparse vectors, which require less computational power compared to the denser vectors generated by Word2Vec and FastText. Comparatively, we see a slight increase in training time compared to TF-IDF with Word2Vec embeddings. This is reasonable as Word2Vec creates denser vectors that capture more intricate semantic relationships between words. The training time with Word2Vec embeddings spans from around 30 seconds for Logistic Regression to 80 seconds for XGBoost, according to the Figure 1. Finally, FastText embeddings, which incorporate sub-word information in addition to whole words, lead to the longest training times among the three techniques. This additional layer of complexity likely translates into even denser vectors, resulting in training times ranging from roughly 50 seconds for Logistic Regression to over 100 seconds for XGBoost



(Figure 1: Model training time with different embedding techniques)

Training Time Analysis As shown in the Figure 2, the selection of an embedding technique significantly impacts the training time of machine learning models for text classification tasks. TF-IDF reigns supreme in terms of speed, with training times ranging from just 0.42 seconds for Logistic Regression to 3.03 seconds for XGBoost. This efficiency likely arises from TF-IDF's use of sparse vectors. Moving to Word2Vec embeddings, which capture more intricate semantic relationships through denser vectors, training times experience a slight increase, spanning from around 0.73 seconds for Logistic Regression to 4.23 seconds for XGBoost. Finally, FastText embeddings, incorporating sub-word information for even richer text representations, necessitate the longest training times among the three techniques, ranging from roughly 1.05 seconds for Logistic Regression to 5.12 seconds for XGBoost. This analysis underscores

the crucial trade-off between training speed and model performance. While TF-IDF offers the quickest training times, it might come at the expense of lower accuracy compared to the potentially richer semantic understanding provided by Word2Vec or FastText embeddings.



(Figure 2: Time taken for embedding techniques)

V. CONCLUSION & FUTURE SCOPE

Our research offers valuable insights into the effectiveness of three text embedding methods - TF-IDF, Word2Vec, and FastText - for identifying fake news. While TF-IDF embeddings achieved remarkable accuracy, demonstrating their strength in capturing word frequency patterns, Word2Vec embeddings offer an advantage in capturing semantic relationships, despite exhibiting lower accuracy in this specific context. Furthermore, FastText embedding outperformed Word2Vec by incorporating sub-word information. Looking ahead, Word2Vec and FastText present promising alternatives to the well-performing TF-IDF method. Future research should explore avenues for further improvement in fake news detection systems, which may involve investigating ensemble approaches that combine the strengths of multiple embedding techniques and models. These findings open exciting avenues for future research in developing more robust fake news detection systems. Future studies could explore ensemble approaches that combine the strengths of multiple embedding techniques. Additionally, transformer-based models like BERT and GPT, with their sophisticated capabilities for understanding contextual information, hold significant promise for boosting accuracy. By pursuing these paths, researchers can contribute to the development of cutting-edge methods to combat the spread of false information.

VI. REFERENCES

[1] Mojumder, Pritom & Hasan, Mahmudul & Hossain, Faruque & Hasan, K. M.. (2020). A Study of fastText Word Embedding Effects in Document Classification in Bangla Language. 10.1007/978-3-030-52856-0_35.

[2] Das, Mamata et al. “A Comparative Study on TF-IDF Feature Weighting Method and Its Analysis Using Unstructured Dataset.” International Conference on Computational Linguistics and Intelligent Systems (2023).

[3] Isnaini Nurul Khasanah, Sentiment Classification Using FastText Embedding and Deep Learning Model, Procedia Computer Science, Volume 189, 2021, Pages 343-350, ISSN 1877-0509.

[4] Derry Jatnika, Moch Arif Bijaksana, Arie Ardiyanti Suryani, Word2Vec Model Analysis for Semantic Similarities in English Words, Procedia Computer Science, Volume 157, 2019, Pages 160-167, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2019.08.153.

[5] Cahyani, Denis Eka and Irene Patasik. “Performance comparison of TF-IDF and Word2Vec models for emotion text classification.” Bulletin of Electrical Engineering and Informatics (2021): n. pag.

[6] Selva Birunda, S., Kanniga Devi, R. (2021). A Review on Word Embedding Techniques for Text Classification. In: Raj, J.S., Iliyasu, A.M., Bestak, R., Baig, Z.A. (eds) Innovative Data Communication Technologies and Application. Lecture Notes on Data Engineering and Communications Technologies, vol 59. Springer, Singapore.

[7] Handler, Abram, "An empirical study of semantic similarity in WordNet and Word2Vec" (2014). University of New Orleans Theses and Dissertations. 1922. https://scholarworks.uno.edu/td/1922

[8] Lai, Siwei et al. “How to Generate a Good Word Embedding.” IEEE Intelligent Systems 31 (2015): 5-14.

[9] Socher, Richard et al. “Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank.” Conference on Empirical Methods in Natural Language Processing (2013).

[10] Yang, X., Macdonald, C. and Ounis, I. (2018) Using word embeddings in Twitter election classification. Information Retrieval, 21(2-3), pp. 183-207.

[11] Zhang, Yunxiang and Zhuyi Rao. “n-BiLSTM: BiLSTM with n-gram Features for Text Classification.” 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC) (2020): 1056-1059.

[12] Man Lan, Chew-Lim Tan, Hwee-Boon Low, and Sam-Yuan Sung. 2005. A comprehensive comparative study on term weighting schemes for text categorization with support vector machines. In Special interest tracks and posters of the 14th international conference on World Wide Web (WWW '05). Association for Computing Machinery, New York, NY, USA, 1032– 1033.

[13] Yufang, ZHANG & Shiming, PENG & Jia, LV. (2006). Improvement and Application of TFIDF Method Based on Text Classification. 32.

[14] Mikolov, Tomas et al. “Efficient Estimation of Word Representations in Vector Space.” International Conference on Learning Representations (2013).

[15] Enriching Word Vectors with Subword Information (https://aclanthology.org/Q17-1010) (Bojanowski et al., TACL 2017