Enhancing Fake News Detection with Federated Learning and Word Embedding

MSc Research Project
MSc Artificial Intelligence

Ramanathan Arunachalam
Student ID: x22142401

School of Computing
National College of Ireland

Supervisor: Prof. Abdul Razzaq

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ramanathan Arunachalam
Student ID:	x22142401
Programme:	MSc Artificial Intelligence
Year:	2023
Module:	MSc Research Project
Supervisor:	Prof. Abdul Razzaq
Submission Due Date:	14/12/2023
Project Title:	Enhancing Fake News Detection with Federated Learning and Word Embedding
Word Count:	5326
Page Count:	15

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	31st January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Fake News Detection with Federated Learning and Word Embedding

Ramanathan Arunachalam
x22142401

Abstract

This research paper explores the effectiveness of combining word embedding techniques and federated learning in fake news detection. The research was conducted using various datasets and three machine learning models (LSTM, CNN, and BERT) were trained using four different word embedding techniques (Word2Vec, Glove, FastText, and Doc2Vec). The results demonstrate that LSTM and CNN models, when combined with either Word2Vec or Doc2Vec, can significantly improve fake news detection accuracy. Federated learning also had a positive impact on accuracy. The research has several key contributions: 1) It demonstrates the effectiveness of combining word embedding techniques and federated learning in fake news detection. 2) It identifies the most effective word embedding techniques and machine learning models for fake news detection. 3) It suggests future directions for research on fake news detection.

1 Introduction

The evolution of social media and easy its accessibility has created an environment where misleading, fake information can be shared. Amount of online users are astonishing, in 2019, there were 7.7 billion people worldwide, with at least 3.5 billion online. This means social media platforms were used by one in three people worldwide. Ortiz-Ospina (2019) A fake news can have considerable impact including changing market share prices, political results and inciting violence.

One of the biggest challenge will be to distinguish between which is fake and real when you consuming news via social media platforms. During the COVID-19 pandemic, 80% of US adults have consumed fake news about it says TechJury (2023). But every manipulation results to a harmful implications, that range from poor decision-making to violence. Figures 1(a) and (b) are two common examples of fake news on social media platforms.

Figure 1: (a) Misinformation and (b) Fake News

Fake news is not only the danger in Internet. Among many others, cyberbullying, online privacy, hacking, phishing attackers, malware, and more. According to Stanford University researchers, 72.3 percent of fake news comes from official news outlets and online social media platforms.Allcott and Gentzkow (2017) To get more clarity on the concept of fake news in current day world, one needs to understand the difference between disinformation, misinformation and fake news. Therefore understanding these terms help one to solve the problem in a better way. The Figure 2 below helps with the definitions and there differences

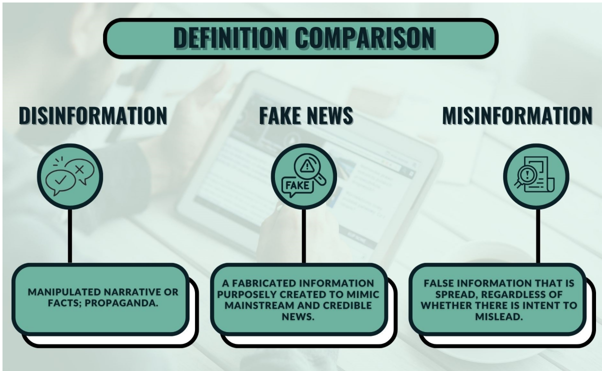

Figure 2: Terms associated with fake news

With rapid developments in Artificial Intelligence (AI), a large number of experiments are being conducted to address the topics like fake news, weather prediction, network monitoring that causes trouble to society. However, with the massive amount of news that is always increasing, how good are these AI modules in detecting fake news is a question to be answered. In this instance, we might have to rely on artificial intelligence (AI) to do the heavy lifting and tell us if there are any obvious linguistic patterns. In this paper we have illustrated and implemented a fake news classification using different machine learning techniques based on word embedding from a huge open labelled corpus.

Despite significant achievements in detecting fake news from legitimate content, machine and deep learning models continue to fall short in terms of computational cost, accuracy, and data privacy. Because the architecture of these computational models is incompetent, a new efficient and data privacy based architecture is necessary for the fake news classification. Existing methods for detecting fake news employ common and shared datasets, hence there is no data privacy from one system to another in a distributed context. In this architecture the models to be trained without the need to gather data in a centralized location. Decentralization helps with data privacy and it significantly increased the accuracy. Khullar and Singh (2023)

The ability to train different models on different datasets and aggregating the model training into one global model, increases the accuracy and also decrease the bias in over all model because of the use of diverse dataset.

Many researchers try to develop different machine learning (ML) models with different sets of features to answer the problem of the fake news detection process. Lancaster et al. (2018) and Verma and Agrawal (2020) using various text-based linguistic approaches. However, the following three questions are not answered.

1. Which word embedding (WE) technique predicts fake news better?
2. Does decentralized learning impact fake news detection?
3. Incorporation of privacy measures and bias management considered in the modeling process?

To answer the above three questions, we have done an extensive research using 8 different word embedding across 5 different dataset. And finally implemented federated learning that answers privacy concern. Among the used word embedding, the best word embedding are chosen and the results are aggregated to improve the accuracy.

The report has 7 sections. Section II gives a detailed overview on other related works. Section III elaborates on the methodology. Section IV gives you the design specification and Section V gives the implementation that describes the proposed methodology, followed by the evaluation in Section VI. Section VII concludes the article and highlights the future work.

2 Related Work

In an age where false WhatsApp forwards and Tweets can influence naive minds, tools and expertise must be put to practical use in reducing the spread of fake news. In this section, we have provided a detailed overview of existing works on fake news and federated learning, and the intricacies of these concepts, which determine the scope of our current research.

2.1 Fake News Classification

Rai et al. (2022) propose a deep learning-based fake news classification framework that leverages the combined strengths of transformer-enhanced Long Short-Term Memory (LSTM) networks and Bidirectional Encoder Representations from Transformers (BERT). A final classification layer integrates the embedded features from both LSTM and BERT to accurately classify news. This work used three dataset and showcased an average accuracy of 93.33%.

Mahabub (2020) propose Ensemble Voting Classifier technique for fake news detection, which combines the strengths of multiple machine learning algorithms to achieve better performance. In the proposed technique on two real-world fake news datasets, obtaining an accuracy of 94.5% and 92.2%, respectively. Though the author achieves high accuracy this method is computationally expensive and it can be sensitive to the training data.

In the proposed model Lai et al. (2022) consists of two main components Sentiment Analysis and Topic Modeling. The authors use the SentiWordNet lexicon to assign sentiment scores to individual words and phrases in the articles and Latent Dirichlet Allocation (LDA) to identify the dominant topics in each article. The results demonstrate that the proposed method achieves an average accuracy of 92.78% and 90.43% on the two datasets, respectively. But the major drawback would be the choice of method to evaluate fake news both Sentiment Analysis and Topic Modeling may not be sufficient to capture all the variations of fake news because fake news articles may often use emotional language or misleading topics.

The paper "DeepFakE by Kaliyar et al. (2020) proposes a novel method for detecting fake news using a coupled matrix-tensor factorization (CMTF) algorithm to extract social context features from news articles and a deep neural network to classify news articles as either fake or real. DeepFakE outperforms above mentioned methods, achieving an average accuracy of 85.86% and 88.64% on the two datasets. Authors have used bag-of-words (BoW) representation which has limited ability to capture semantic and syntactic relationships.

User-centered fake news detection by Minjung Park (2023) suggest a model that highlights the relevance of taking user behaviour into account when detecting fake news. The model can better judge the reliability of news articles by analysing a user's interests and preferences. The model is reliant on user profile data, which may be incomplete or inaccurate. This could lead to false positives or negatives in fake news detection. And the model does not explicitly consider the temporal dimension of news articles. This means that the model may not be able to effectively detect fake news that is spreading quickly or that has been recently created.

2.2 Federated Learning

Fake news is like wild fire, its crucial to develop effective and privacy-sensitive solutions for identifying and combating fake news. In Khullar and Singh (2023) works on distributed training process across multiple clients, leveraging their computational resources and reducing the burden on the central server. It emerges as a promising solution for effectively combating fake news while safeguarding data privacy. But it only employs a hybrid model that combines the strengths of CNN and Transformer architectures. Which is good, but word embedding technique can add more value to the CNN architecture by giving semantic meaning of text. This will be more appropriate for fake news as context is important in this learning process. In the below subsections we can comprehensively review about fake news analysis and federated learning.

In Ouyang et al. (2021) The proposed model, named FL_FNDM, consists of two main components: a deep self-attention network (DSAttNet) for extracting linguistic features from news headlines and a federated learning framework for training the model without compromising user privacy. The DS-AttNet is able to learn both local and global linguistic features from COVID-19 news headlines, which is crucial for identifying fake news. Overall, the FL_FNDM model was a promising new approach for detecting COVID-19 fake news. It is accurate, privacy-preserving, and scalable.

Islam et al. (2023) presents a federated learning-based text classification framework that utilizes natural language processing (NLP) techniques to effectively classify text data. The proposed framework consists of four main components, the first being Local Feature Extraction followed by feature aggregation where central server receives encrypted local features from all participating devices. And finally model is trained and and deployed. This is one promising framework for privacy-preserving text classification.

3 Methodology

In this section a detailed overview of this research Fake News Detection with Federated Learning and Word Embedding is discussed.

Table 1 Comparative Analysis of related studies highlighting performance metrics. 1(2)

		This Paper	Rai et al. (2022)	Lai et al. (2022)	Choudhary and Arora (2021)	Mahabub (2020)
Data Analysis	Model	CNN, LSTM, FL	LSTM	CNN, LSTM	LSTM	NB, KNN, SVM, RF, LR, SGD
	No. of Data Set	5	2	2	2	1
	Word Embedding	Word2Vec, Doc2Vec, Fast-Text, GLoVe		Word2Vec		
Performance Metrics	Accuracy %	96.5	84.10 - 88.75	90 - 94	86	93-94
Federated Learning	Data Privacy	Yes	No	No	No	No
	Resource Optimization	Yes	No	No	No	No

Table 2 Comparative Analysis of related studies highlighting performance metrics. 2(2)

		Khullar and Singh (2023)	Zhu et al. (2023)	Keskar et al. (2020)	Kaliyar et al. (2020)	Ouyang et al. (2021)
Data Analysis	Model	LSTM, CNN-LSTM, Bi LSTM, CNN-Bi LSTM	StyleLSTM, DualEmo Bi-GRU, TextCNN, RoBERTa	SDG, SVM, LSVM, KNN, LR, DT	XGBoostDNN (DeepFakE)	MMT-AN
	No. of Data Set	3	3	2	5	3
	Word Embedding	Glove		n-gram		GLoVe, Fast-Text
Performance Metrics	Accuracy %	90-92	97	60 - 92	(F-measure) 87	88-92
Federated Learning	Data Privacy	Yes	No	No	No	Yes
	Resource Optimization	Yes	Yes	No	No	Yes

3.1 Word Embeddings

Word Embeddings is a technique in NLP where individual words are represented as real-valued vectors and captures inter-word semantics. Each word is represented by a real-valued vector with tens or hundreds of dimensions. There are various word embeddings used in this research, each being different from one another they impact the final model performance.

- Word2Vec: The Word2Vec model learns word embeddings from huge text corpora using shallow neural networks. It consists of two model architectures: Skip-gram and Continuous Bag-of-Words.
- Doc2Vec: Doc2Vec is a Word2Vec model extension that generates fixed-size vectors to represent full documents or sentences in a continuous vector space. There are two main Doc2Vec implementations: PV-DM (Paragraph Vector - Distributed Memory) and PV-DBOW (Paragraph Vector - Distributed Bag of Words).
- FastText: FastText is an extension of Word2Vec that considers sub-word information by breaking words into character n-grams. This allows it to generate embeddings for out-of-vocabulary words and handle morphological rich languages more effectively.
- Glove: Global Vectors for Word Representationis (GlVE an unsupervised learning algorithm for obtaining word embeddings by aggregating global word-word co-occurrence statistics from a corpus.

3.2 Federated Learning

Federated learning is a machine learning approach that enables training of models across multiple decentralized devices or servers holding local data samples without exchanging the raw data. Federated learning enables edge devices to use state of the art machine learning without centralising data. Since raw data remains on local devices and only model weights are shared, the client trains with the shared weighs of the global model and after training it updates the new weighs to the global model back, this federated learning helps protect user privacy and confidentiality.

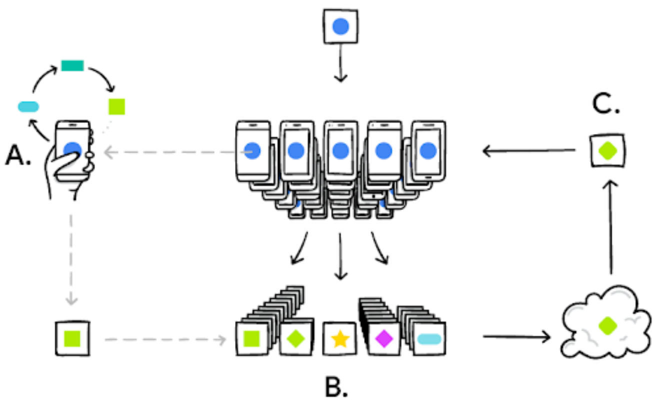

Figure 3: Federated Learning Depiction

At "A" our mobile gets a parameter(blue dot) train the model and get an update (green). Several such updates from many mobiles are aggregated at "B". At "C", the shared model parameter evaluates the parameters of a node based on its previous value and the shared value and send the previous update of the node and updates the localized model at the node.

3.3 Datasets

For this research we have used 5 datasets. The reason behind the use of five datasets is the five word embeddings we wanted to train on. Each dataset is completely different from one another and number of parameters are also different. Dataset 1: Truthseeker data is one new dataset that we used is published in this year. The detailed overview of the dataset used is given in table below for better understanding.

Table 3 Datasets Shape

Dataset Name	Rows	Columns
Truth_Seeker (2023)	134199	8
Kaggle Fake News Data 1	44920	4
LIAR	10239	14
WEFL	72134	4
Kaggle Fake News Data 2	6335	4

4 Design Specification

Federated learning uses decentralized architecture where there are five or three clients in our scenario. Each client is trained with its own data and word embedding, but a common machine learning module like LSTM, CNN and BERT models are used. The reason for common ML model is because of the shape mismatch when aggregating the weights in the global model. The below is a small depiction of the architecture we have deployed.

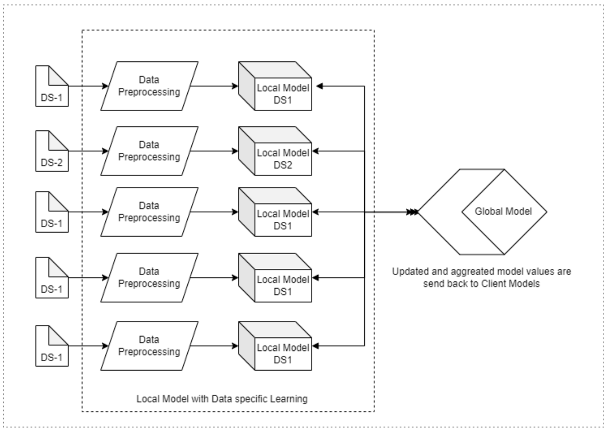

Figure 4: Federated Learning Architecture

Before global model is deployed, the client or local model has to be defined. The best model is selected based on the accuracy it produced by the word embedding model. Only the top models are selected for client server architecture of federated learning.

5 Implementation

5.1 Data Preparation

As this research contains five datasets loading all five data and understanding them was the key to this research. With all the required libraries installed, five dataset were imported as dataframes. These five dataframe were further used for Exploratery Data Analysis.

5.2 EDA

Once data was imported analyzing and investigating data sets and summarizing their main characteristics was the next step. Each dataset was taken one by one and data was cleaned first from missing values and duplicates. Then the cleaned df is used for plotting charts and graphs.

These graphs for each dataset, show the distribution of true and fake news, how other lables are distributed, and finally word cloud of the data set.

Word clouds are visual representations that display the most frequently occurring words in a text corpus, with the size of each word proportional to its frequency.

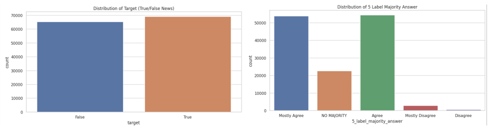

Figure 5: Truth Seekers Dataset

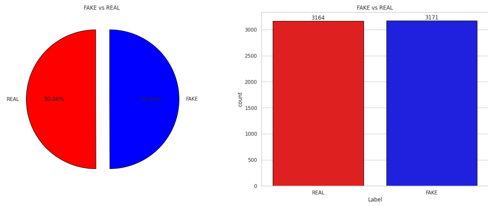

Figure 6: Dataset Kaggle 1

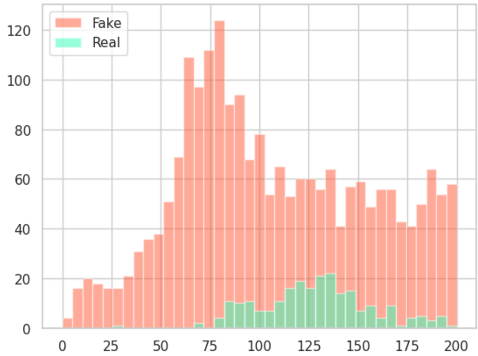

Figure 7: WEFL Dataset

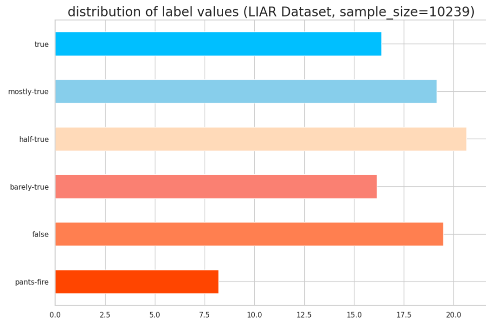

Figure 8: LIAR Dataset

5.3 Modeling

5.3.1 Text Preprocessing

Text preprocessing plays a crucial role in extracting meaningful insights from unstructured text data. It involves a series of cleaning and normalization techniques Kadhim (2018) that are essential for preparing the text for further analysis.

1. Stopword removal: Stop words refer to common words that do not carry significant meaning in the context of the text. These words often represent grammatical functions rather than conveying semantic weight. By eliminating stop words, the noise in the text is reduced, and the model can focus on the more meaningful words

2. Normalization: Is converting all text to a consistent case, such as lowercase or uppercase. This ensures uniformity in the text and reduces the number of unique words that need to be processed. Case normalization simplifies downstream tasks like tokenization and stemming/lemmatization

3. Punctuation removal is another crucial step in text preprocessing. Punctuation marks, such as commas, periods, and exclamation marks, can introduce noise and hinder the model's ability to understand the text. Removing these punctuation marks simplifies the text and ensures that the model focuses solely on the actual words.

4. Whitespace normalization involves consolidating multiple whitespace characters into a single space. This normalization technique helps maintain consistency in the text and prevents issues that may arise during tokenization—the process of breaking the text into individual words or tokens. By ensuring consistent whitespace, the model receives a standardized representation of the text.

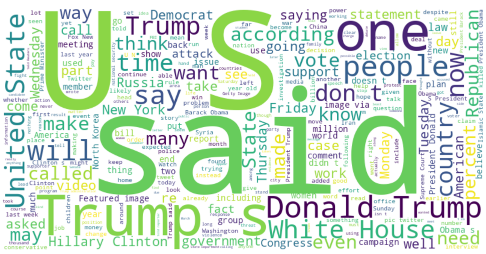

Figure 9: Word Cloud WEFL Dataset

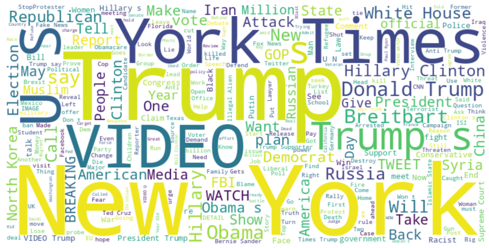

Figure 10: Word Cloud TruthSeeker Dataset

Overall, these preprocessing techniques help to clean and normalize the text, enabling more accurate and effective analysis of unstructured text data.

5.3.2 ML Model

There are three manin Machine Learning (ML) Model used in this analysis. LSTM (Long Short-Term Memory) Berraja (2022), CNN (Convolutional Neural Network) and BERT (Bidirectional Encoder Representations from Transformers). BERT by itself being a model that has a pre-trained language model developed by GoogleTenney et al. (2019). It uses Transformer architecture to understand the context of words in a sentence by capturing bidirectional dependencies, so other word embeddings was not used with this model.

Both with LSTM and CNN, four word embeddings were included in the model performance. When analysze all four word embedding gave good results but most of the results were similar. The epochs were kept minimum (5 or 10) for handling the memory.

5.3.3 Word Embedding

All the dataset are taken one by one and all four word embeddings were done on each dataset. The vector_size=100 was kept standard throughout the process. In this process all the words in the corpus were converted as array with multiple dimensions based on embedding used.

5.4 Federated Learning

Federated learning is an exciting approach that allows us to train machine learning models while keeping data privacy intact. It's a departure from traditional methods where all the data is sent to a central server. With federated learning, we can collaborate and train models across multiple devices or clients without revealing sensitive data to a central server. This way, we can ensure that individual data remains secure while still benefiting from shared knowledge to improve model performance.

To start the federated learning process, we need to select the best model from a set of candidates. We look for a model that can accurately predict the target variables, minimize loss, and demonstrate efficient learning behavior over time.

Once we have chosen the optimal model, we design the federated learning architecture. This architecture determines the number of participating clients and the communication protocols between them. For example, in this case, we decided to use a three-client architecture, which outperformed a five-client architecture. This suggests that having fewer participating clients can lead to faster training and more effective learning, thanks to reduced communication overhead.

In conclusion, federated learning provides a powerful solution for training models while preserving privacy. It allows us to leverage the collective power of distributed data without compromising individual privacy. By carefully selecting the right model and optimizing the federated learning architecture, organizations can enjoy improved model performance while upholding the principles of data privacy.

6 Evaluation

Evaluating the performance of all the models mentioned in above section are extremely critical. All the model have common evaluation metrics

- Classification Accuracy: It is the number of right predictions divided by the total number of predictions made.

$$\text{Accuracy} = \frac{\text{Number of Correct predictions}}{\text{Total number of predictions made}} \tag{1}$$

- Logarithmic Loss: Loss is the value that represents the summation of errors in our model. To calculate the loss, a loss or cost function is used. In our case MSE (Mean Squared Error) calculates the validation loss of the model.

$$\text{Validation Loss} = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2 \tag{2}$$

  Similarly for classification problem, cross-entropy loss is commonly used.

$$\text{Binary Cross-Entropy Loss} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)) \tag{3}$$

- Mean Squared Error: Mean Squared Error gives the average of the squared difference between the original and predicted values in the data set. It measures the variance of the residuals.

$$\text{Mean Squared Error} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{4}$$

6.1 Evaluation and Results of LSTM

In this study, five datasets—Truthseeker, Kaggle 1, Kaggle 2, LIAR, and WEFL—were used to compare the performance of four distinct word embedding models (Word2Vec, GloVe, FastText, and Doc2Vec) using two performance metrics: accuracy and loss.

As shown in the table below, Word2Vec consistently demonstrated superior performance across all five datasets, achieving the highest accuracy scores of 99.92, 99.98, and 99.99. In contrast, GloVe consistently exhibited the lower accuracy scores, with its performance ranging from 68.78 for LIAR to 91.00 for WEFL. Doc2Vec models was a strong performing model among all, as its accuracy is not less than 89.20 for all datasets.

Table 4 Evaluation Results - LSTM

LSTM						
Model	Metrices	Truthseeker	Kaggle 1	Kaggle 2	LIAR	WEFL
Word2Vec	**Accuracy**	99.98	53.54	99.52	99.92	99.99
	Loss	0.0106	0.04	0.008	0.053	0.034
GloVe	**Accuracy**	78.64	91.76	74.23	68.78	91.00
	Loss	0.134	0.890	1.432	0.439	0.064
FastText	**Accuracy**	98.98	96.76	85.40	78.90	99.99
	Loss	0.809	0.538	0.573	0.002	0.063
Doc2Vec	**Accuracy**	98.90	89.23	89.32	93.33	92.30
	Loss	0.614	0.037	0.223	1.223	0.049

In terms of datasets, WEFL dataset was with hing accuracy and low loss for all five word embedding. Worst performing dataset in each word embedding are Kaggle 2 vs Glove, Truthseeker vs FastText, LIAR vs Doc2Vec. This is because of high accuracy and high loss.

Increasing the vector size from 100 or increasing LSTM window to 128 might have helped in better performance in some cases. Hyper parameter tuning was not done due to hardware limitations. So these results was not the best achieved

Overall, these findings demonstrate the importance of carefully selecting the appropriate word embedding model for a given task. With consistently better results across a range of measures and datasets, Word2Vec is the best choice in word embedding. GloVe, on the other hand, has consistently poor performance. While Doc2Vec has a tiny advantage in accuracy, and FastText has a slight in loss, both Doc2Vec and FastText offer a balance between accuracy and loss.

6.2 Evaluation and Results of CNN

Table 5 Evaluation Results - CNN

CNN						
Model	Metrices	Truthseeker	Kaggle 1	Kaggle 2	LIAR	WEFL
Word2Vec	**Accuracy**	99.98	98.54	99.52	94.00	99.00
	Loss	0.0106	1.223	0.892	0.053	0.034
GloVe	**Accuracy**	54.98	69.09	55.63	43.99	91.00
	Loss	0.035	0.058	0.035	0.038	0.064
FastText	**Accuracy**	98.89	98.87	99.99	86.98	89.32
	Loss	0.068	1.342	0.957	0.764	0.063
Doc2Vec	**Accuracy**	93.23	79.00	68.45	85.45	79.04
	Loss	0.064	0.037	0.043	0.042	0.049

CNN Model performed better than LSTM. Most of the accuracy was above 90% and minimum loss, but learning rate was very low. Model over fitting was evident and was addressed while training and the results were better after that. Achieving high accuracy throughout was a concern. Hyper-parameter tuning must be done to see better results as like in LSTM.

Here the evaluation for the results we have would be Word2Vec model outperformed other Model, and like previous case GloVe model was the least performing. These results leads to the best model use for federated learning.

6.3 Evaluation and Results of Federated Learning

In the end, the Federated Learning Architecture training is crucial because the project's success depends on it. The table below shows the accuracy and loss. The model's ability to fit the training set of data is shown by the training loss. A smaller training loss suggests that the model fits the data more well. The training loss of the model trained with five clients is 0.68, but the training loss of the model trained with three clients is 3.56. This shows that compared to the model trained with three clients, the model trained with five clients is able to match the data better.

Even though the 3 Client Model's accuracy is higher, the training was discontinued in 3 epoches because of the large loss. Allowing the model to use all available resources despite being aware of its poor performance was discontinued. This further highlights the sustainability of the federated learning programme.

Table 6 Federated Learning

Architecture	Training Loss	Accuracy
5 Clients (10 epoch)	0.68	96.5
3 Clients (3 epoch)	3.56	98.89

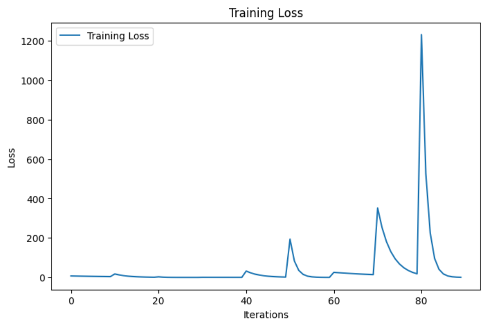

Figure 11: Loss when Less Iterations

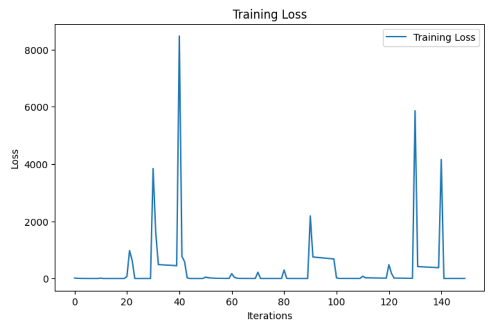

Figure 12: Loss when Iteration Increased

The five client architecture is best performing. And the results suggest that more clients perform better together, whereas three client will also perform better for a different combination of embedding.

6.4 Discussion

The given three research objective was achieved, after the a detailed evaluation of different modules we can be sure that Glove Model and Fast Text model perform better. The predict the fake news better when combined with LSTM and CNN frameworks. The next big question is does decentralized learning impact fake news detection, the answer for that would be yes. As before the research it was expected that federated learning will have drastic impact, but impact in terms of accuracy and learning rate is low but other benefits of federated learning are achieved. With the use of Federated learning and different data set for solving the same problem might have reduced the bias. As each data was independent from one another. Overall the research gave a satisfactory output with all the work being done. For improvement the modeling parameters must be changed and high computational power is need for this level of project. So increasing the hidden layers, increasing epoch might impact the results.

7 Conclusion and Future Work

Word Embedding techniques like Word2Vec, Fast Text provide semantic relationship between words. Using word embeddings was a good choice for research. Federated learning allows training models across multiple decentralized devices keeping data localized. More over the results show how each word embedding is related or how they showcase same output scenarios. This finding might be helpful for future researchers on what embedding to be chosen for their NLP task. Apart from this, the project used diversified datasets from various sources with a mix of new and old data, I believe this might reduce the bias in the dataset used.

In future works Enhancing Model Robustness will be one major task. Future research could focus on improving the robustness of fake news detection models by trying out various iterations of machine learning, which, in this research was limited. Integrating multiple data formats (text, photos, videos) for fake news identification could be investigated. Combining word embeddings with visual or audio embeddings may provide a more complete grasp of the context, enhancing detection accuracy.

Using federated learning and word embeddings to develop real-time fake news detection algorithms would be quite advantageous. Implementing such techniques on platforms to counteract misinformation's quick spread might dramatically reduce its impact.

Finally, the combination of word embedding techniques and federated learning provides a feasible solution for detecting fake news. Future research should concentrate on improving model robustness, studying multi modal techniques, addressing ethical concerns, and enabling real-time deployment to combat disinformation dissemination.

References

Allcott, H. and Gentzkow, M. (2017). Social media and fake news in the 2016 election, *Journal of Economic Perspectives* **31**(2): 211–236.

Berraja, A. (2022). Natural language processing for the analysis sentiment using a lstm model, *International Journal of Advanced Computer Science and Applications (IJACSA)* **13**(5).

Choudhary, A. and Arora, A. (2021). Linguistic feature based learning model for fake news detection and classification, *Expert Systems with Applications* **169**: 114171.
URL: *https://www.sciencedirect.com/science/article/pii/S095741742030909X*

Islam, M., Iqbal, S., Rahman, S., Sur, S. I. K., Mehedi, M. H. K. and Rasel, A. A. (2023). A federated learning approach for text classification using nlp, *Image and Video Technology*, Springer International Publishing, Cham, pp. 14–23.

Kadhim, A. (2018). An evaluation of preprocessing techniques for text classification, *International Journal of Computer Science and Information Security,* **16**: 22–32.

Kaliyar, R. K., Goswami, A. and Narang, P. (2020). Deepfake: improving fake news detection using tensor decomposition-based deep neural network, *Journal of Supercomputing* .
URL: *https://doi.org/10.1007/s11227-020-03294-y*

Keskar, D., Palwe, S. and Gupta, A. (2020). Fake news classification on twitter using flume, n-gram analysis, and decision tree machine learning technique, *Book Title*, Springer, pp. 139–147.

Khullar, V. and Singh, H. P. (2023). f-fnc: Privacy concerned efficient federated approach for fake news classification, *Journal Name or Conference Proceedings* .

Lai, C.-M., Chen, M.-H. and Kristiani, E. (2022). Fake news classification based on content level features, *Applied Sciences* **12**(1116).
URL: *https://doi.org/10.3390/app12031116*

Lancaster, E., Chakraborty, T. and Subrahmanian, V. S. (2018). MALT P: Parallel prediction of malicious tweets, *IEEE Transactions on Computational Social Systems* **5**(4): 1096–1108.

Mahabub, A. (2020). A robust technique of fake news detection using ensemble voting classifier and comparison with other classifiers, *SN Applied Sciences* **2**: 525.
URL: *https://doi.org/10.1007/s42452-020-2326-y*

Minjung Park, S. C. (2023). Constructing a user-centered fake news detection model by using classification algorithms in machine learning techniques, *IEEE Access* .

Ortiz-Ospina, E. (2019). The rise of social media, *Our World in Data* . https://ourworldindata.org/rise-of-social-media.

Ouyang, S., Du, J., Wang, B., Yu, B., Wang, Y. and Liang, M. (2021). Federal learning based covid-19 fake news detection with deep self-attention network, *2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS)*, pp. 296–299.

Rai, N., Kumar, D., Kaushik, N., Raj, C. and Ali, A. (2022). Fake news classification using transformer based enhanced lstm and bert, *International Journal of Cognitive Computing and Engineering* **3**: 98–105.
URL: *https://doi.org/10.1016/j.ijcce.2022.03.003*

TechJury (2023). Fake news statistics, `https://techjury.net/blog/fake-news-statistics/`.

Tenney, I., Das, D. and Pavlick, E. (2019). Bert rediscovers the classical nlp pipeline, *arXiv preprint arXiv:1905.05950* .

Verma, P. K. and Agrawal, P. (2020). Study and detection of fake news: P2c2-based machine learning approach, *Proceedings of the International Conference on Data Management, Analytics, and Innovation*, Vol. 1175, Springer, Singapore, pp. 261–278.

Zhu, Y., Sheng, Q. and Cao, J. (2023). Memory-guided multi-view multi-domain fake news detection, *IEEE Transactions on Knowledge and Data Engineering* **35**(7).