FEDERATED LEARNING FOR NEXT-WORD PREDICTION: A DISTRIBUTED APPROACH FOR IMPROVED LANGUAGE MODELS

P. Narendra Babu[1], K. Likhitha Suma Venkat[2], T. Praveena[3], G. Pradeep Kumar[4]

[1] *Assistant Professor, AI&DS, Lakireddy Bali Reddy College of Engineering, A.P, India*
[2] *Student, AI&DS, Lakireddy Bali Reddy College of Engineering, A.P, India*
[3] *Student, AI&DS, Lakireddy Bali Reddy College of Engineering, A.P, India*
[4] *Student, AI&DS, Lakireddy Bali Reddy College of Engineering, A.P, India*

ABSTRACT

The surge in machine learning necessitates novel techniques that prioritize data privacy and decentralized training. Federated learning (FL) emerges as a promising solution in this domain. This research explores the potential of FL, particularly focusing on Long Short-Term Memory (LSTM) networks for next-word prediction in a Homogeneous FL setting. We propose a secure FL architecture involving three client devices and a central server, each contributing to the learning process. To ensure data security, a robust two-factor authentication framework is implemented for all participating clients. This framework merges mobile One-Time Passwords (OTPs) with traditional username/password credentials, bolstering security against unauthorized access and data breaches. Within the FL setup, client devices independently train the LSTM model on their local datasets. Subsequently, the server aggregates these local models into a global model, which is then distributed back to the clients for further training iterations. This cyclical process continues until the desired level of prediction accuracy is achieved. Furthermore, this research delves into the impact of the proposed secure FL setup on the performance of the LSTM-based next-word prediction model. We analyze the influence of secure data transmission and robust authentication on the learning outcomes, comparing model performance across different scenarios. This investigation sheds light on the trade-offs between security and performance in FL for next-word prediction tasks. Additionally, this study explores how the LSTM-based next-word prediction model performs in relation to the suggested secure FL setup. We examine how reliable authentication and secure data transmission affect learning outcomes by contrasting model performance in various contexts. The trade-offs between security and performance in FL for next-word prediction tasks are clarified by this work.

Keyword: *Federated Learning, LSTM , Authentication, and secure data transmission*

1. INTRODUCTION

The realm of machine learning (ML) has witnessed remarkable advancements in recent years, fundamentally transforming various aspects of our lives. These advancements are driven by powerful algorithms like deep learning (DL), which excel at extracting patterns from massive datasets. However, leveraging these techniques often necessitates the collection and centralization of vast amounts of data, raising concerns about data privacy and security.

Federated learning (FL) emerges as a groundbreaking solution that addresses these privacy concerns by enabling collaborative learning without compromising sensitive information. In contrast to traditional centralized learning, where data is aggregated on a central server, FL empowers devices to train models directly on their local datasets. These local models are then transmitted to a central server, where they are aggregated to create a global model. This global model, devoid of the original data itself, is then disseminated back to the participating devices for further refinement. This iterative process continues until the model achieves a satisfactory level of performance.

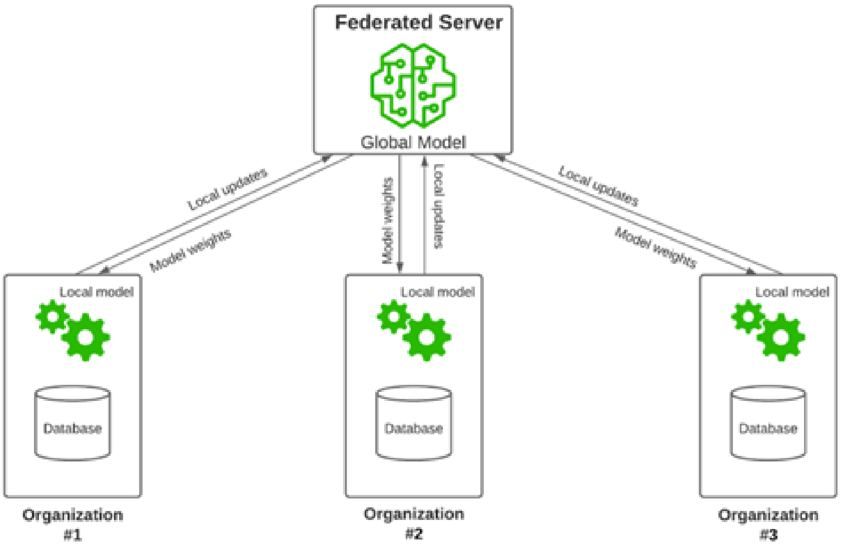


Fig-1: Federated learning Process

1.1 Disadvantages of Machine Learning and Deep Learning:

- **Data Privacy Concerns:** Centralized ML and DL approaches often raise privacy concerns as they necessitate the collection and storage of large datasets, potentially containing sensitive information. This centralized storage creates vulnerability in case of security breaches.
- **Data Bias and Fairness:** ML models are susceptible to inheriting biases present within the training data. If the training data is not diverse and representative, the model's predictions can be biased and unfair to certain demographics.
- **Computational Requirements:** Training complex deep learning models often requires significant computational resources, making them inaccessible for devices with limited processing power. This can hinder the widespread adoption and deployment of such models.

1.2 Advantages of Federated Learning:

- **Data Privacy:** FL safeguards user privacy by keeping data on individual devices, eliminating the need for direct data transfer. This is particularly advantageous for applications handling sensitive data like healthcare information or financial records.
- **Decentralized Learning:** FL facilitates collaborative learning across geographically dispersed devices, eliminating the need for a central repository of data. This distributed approach minimizes the risk of single points of failure and fosters collaboration within constrained network environments.
- **Improved Scalability:** FL leverages the computational power of participating devices, enabling the training of models on massive datasets that would be impractical for a single server to handle. This distributed processing capability fosters scalability and cost-effectiveness.

By mitigating data privacy concerns and fostering decentralized learning, FL offers a compelling alternative to traditional centralized ML and DL approaches. This research delves deeper into the potential of FL, particularly focusing on leveraging Long Short-Term Memory (LSTM) networks for next-word prediction in a secure Homogeneous FL setting.

2. LITERATURE SURVEY

[1] This research tackles the challenge of continuously evolving data in next-word prediction tasks. It proposes a federated learning approach with dynamic clients that can adapt to new data distributions without retraining the entire model.

In the study conducted by **Hui Jiang et al. (2020) [2],** a novel paradigm was developed to integrate multiple diverse devices into a unified server for Mobile Edge Computing (MEC). The researchers utilized two algorithms: FedAvg, which demonstrated an accuracy below 80%, and CuFL, which achieved accuracy exceeding 80%. While CuFL effectively tackles the challenge of resource scarcity, it struggles with maintaining high accuracy.

In 2018, **Professor Dr. Wolfgang Mulzer** introduced Federated Learning as an emerging domain within machine learning. This approach enables model training without the necessity of centralizing the data itself **[3].** Instead of transmitting raw data, participants jointly refine a model by solely exchanging weight updates with a central server. Although this method greatly enhances privacy and offers flexibility in certain contexts, it does entail associated trade-offs.

In their study **(H.B. McMahan, 2016) [4],** researchers focused on enhancing computational efficiency through two primary methods:

- **Enhanced Parallelism:** Expanding parallel processing by engaging more clients to operate independently during each communication round.
- **Augmented Computation at each Client:** Rather than executing basic tasks like gradient calculations, clients undertake more intricate computations between communication rounds. This approach has demonstrated the ability to train high-quality models with minimal communication rounds.

In the study by **Takayuki Nishio et al. (2018) [5],** the authors introduced the FedCS protocol, designed to establish a framework for setting deadlines for clients to download and upload machine learning models through the Federated Learning (FL) protocol. Within a Mobile Edge Computing (MEC) setting, they established a base station leveraging a cellular network with K=1000 clients, utilizing LTE for wireless communications.

In 2021, **Hangyu Zhu** delved into Communication Efficient and Secure Federated Learning **[6].** Traditional machine learning approaches demand users to share their personal data with a central cloud for model training, posing substantial privacy risks. Federated learning, however, offers a novel approach to preserving privacy in machine learning by decentralizing the training process, thereby mitigating these privacy concerns.

3. PROPOSED SYSTEM

Federated Learning Environment Setup: Start by setting up the necessary libraries and dependencies for your project, such as TensorFlow or PyTorch for implementing the LSTM model, and include socket programming libraries for communication.

Define Model Architecture: Design the architecture of the LSTM model for next word prediction within each client. Specify the number of LSTM layers, units, activation functions, and other relevant parameters.

Client Implementation:

- Implement the client-side code using socket programming techniques.
- Develop functions to preprocess data, tokenize sequences, define the LSTM model, train the model, and transmit model updates to the server.
- Ensure the client code includes mechanisms to handle authentication requests from the server.

Server Implementation:

- Create the server-side code utilizing socket programming methodologies.
- Implement functionalities to receive model updates from clients, perform aggregation of weights using the Federated Averaging (FedAvg) method, and distribute updated weights back to clients.
- Incorporate authentication mechanisms to ensure secure communication between clients and the server.

Training Initialization: Initialize the LSTM model weights on the server. And establish connections between the server and clients.

Federated Learning Process:

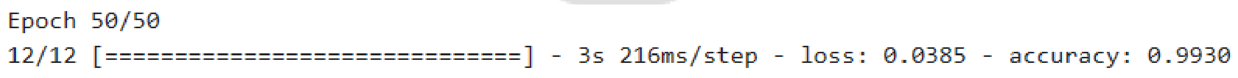
- Kickstart the federated learning process by dispatching initial model weights from the server to each client.
- Iterate over multiple rounds of training:
- Clients preprocess their local data and train their LSTM models on their respective datasets.
- Clients transmit their updated model weights to the server.
- The server aggregates the received model weights from all clients using the FedAvg method.
- The server sends the aggregated weights back to each client for the subsequent round of training.

Model Evaluation and Saving: After a predetermined number of rounds, evaluate the performance of the global model using a validation dataset. Finally, save the final trained model weights on the server.

Testing and Deployment: Assess the performance of the final trained model on unseen data through testing. Then deploy the trained model for real-world applications, ensuring compatibility and scalability while considering any additional data privacy or security measures as necessary.

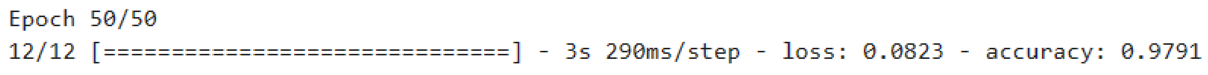
4. RESULTS

This research investigated the effectiveness of a federated learning (FL) approach for next-word prediction. We implemented server-client architecture with one server and three clients. Each client trained a Long Short-Term Memory (LSTM) model on a local dataset. The server then aggregated the model updates from the clients without exchanging raw data, ensuring privacy.



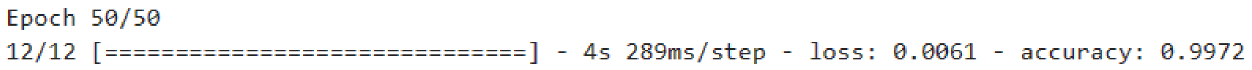
```
Epoch 50/50
12/12 [==============================] - 3s 216ms/step - loss: 0.0385 - accuracy: 0.9930
```

Fig-3: Model Accuracy of Client -1



```
Epoch 50/50
12/12 [==============================] - 3s 290ms/step - loss: 0.0823 - accuracy: 0.9791
```

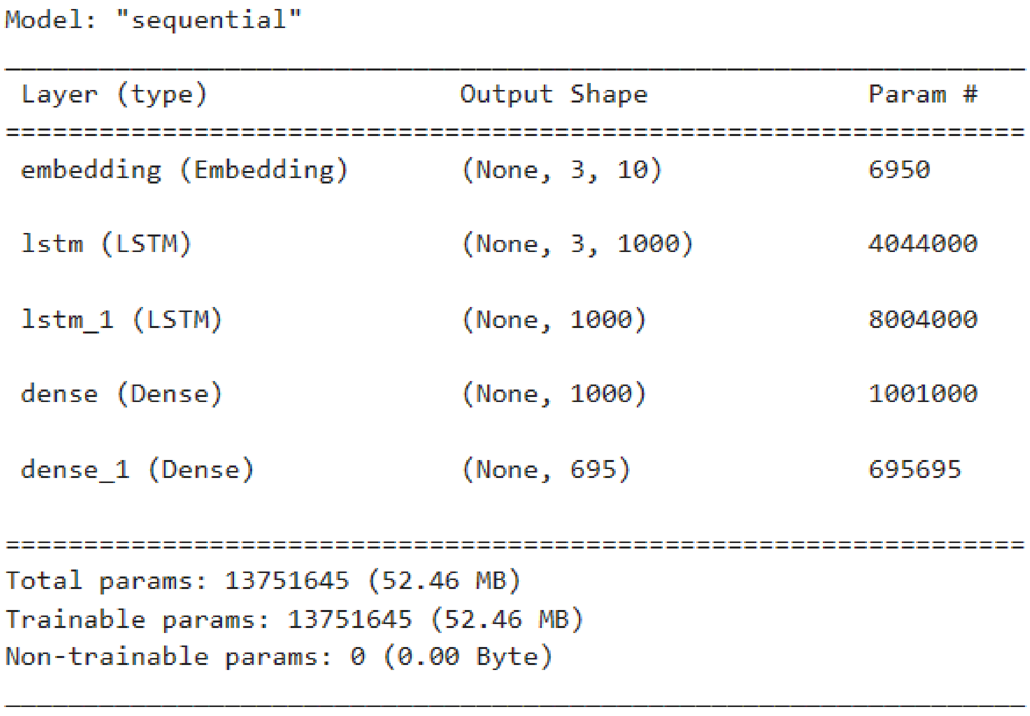
Fig-4: Model Accuracy of Client -2



```
Epoch 50/50
12/12 [==============================] - 4s 289ms/step - loss: 0.0061 - accuracy: 0.9972
```

Fig-5: Model Accuracy of Client -3

All three clients achieved high accuracy exceeding 95%, demonstrating the effectiveness of FL for next-word prediction. This approach offers a promising solution for training large language models while preserving data privacy on individual devices.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 3, 10)             6950

 lstm (LSTM)                 (None, 3, 1000)           4044000

 lstm_1 (LSTM)               (None, 1000)              8004000

 dense (Dense)               (None, 1000)              1001000

 dense_1 (Dense)             (None, 695)               695695

=================================================================
Total params: 13751645 (52.46 MB)
Trainable params: 13751645 (52.46 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig-2: Parameters of the LSTM model

5. CONCLUSION AND FUTURE SCOPE

The development of the LSTM-based word prediction model marks a significant milestone in natural language processing. Leveraging a diverse dataset sourced from books and online literature, the model demonstrates high accuracy in predicting the next word in a sequence. Meticulous preprocessing and model design highlight LSTM networks' effectiveness in capturing contextual dependencies for generating contextually relevant predictions.

Looking forward, further refinement and optimization of the model architecture and hyper parameters are essential for improved prediction accuracy and efficiency. Integration into various applications can aid users in text composition, typing assistance, and content generation. Exploring multimodal approaches and enhancing semantic understanding will enrich prediction capabilities. Rigorous evaluation against existing methods and datasets will benchmark performance and guide improvements.

Through continual research and development, we aim to enhance LSTM-based word prediction models, advancing natural language processing and its practical applications. This pursuit will contribute to a deeper understanding of language dynamics and facilitate more seamless human-computer interactions.

6. REFERENCES

[1] (n.d.). Retrieved from https://link.springer.com/article/10.1007/s10462-023-10563-8

[2] Hui Jiang a, M. L. (2020). Customized Federated Learning for accelerated edge computing with heterogeneous task targets. *elsevier* , 13.

[3] Prof. Dr. Wolfgang Mulzer [2018] Federated Learning.

[4] H.B. McMahan, E. M. (2016). *Federated Learning of Deep Networks using Model Averaging*. Retrieved from https://arxiv.org/pdf/1602.05629v1.pdf

[5] Takayuki Nishio, R. Y. (2018). *Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge*. Retrieved from https://arxiv.org/abs/1804.08333

[6] Hangyu Zhu[2021] [16] Communication Efficient and Secure Federated Learning.