Privacy-preserved Collaborative Federated Learning Platform for Industrial Internet of Things

Lakshan Pathiraja*, Isuru Lakshan†, Kavini Kushani‡,
Chamara Sandeepa§, Tharindu Gamage¶, Thilina Weerasinghe‖, Madhusanka Liyanage**
*†‡¶‖Faculty of Engineering, University of Ruhuna, Sri Lanka
§**School of Computer Science, University College Dublin, Ireland
Email: *eg183412@engug.ac.ruh.lk, †eg183365@engug.ac.ruh.lk, ‡eg183375@engug.ac.ruh.lk,
§abeysinghe.sandeepa@ucdconnect.ie, ¶tharindu@eie.ruh.ac.lk, ‖thilina@eie.ruh.ac.lk, **madhusanka@ucd.ie

***Abstract*—Federated learning (FL) is an intriguing approach to privacy-preserving collaborative learning. Decentralised FL is achieving increased favour for investigation due to the mitigation of vulnerability for a single point of failure and more controllability for end users over their models. However, many existing decentralised FL systems face limitations, such as privacy concerns, latency in aggregation, and real-world implementation challenges. To mitigate these issues, we introduce a novel FL protocol with a decentralised Peer-to-Peer (P2P) system using Differential Privacy (DP). It consists of a decentralised accuracy-based weighted averaging mechanism for both enhanced privacy and model aggregating accuracy. We implement our system in both virtual and real environments to evaluate the performance of the proposed mechanism. Moreover, we perform a comparative analysis of our proposal with both existing centralised and decentralised systems. To practically demonstrate the work, we consider a real-world use case of a recommendation system using smart carts. Experimental results show that our novel approach efficiently performs privacy-preserved aggregations over a decentralised peer network.**

***Index Terms*—Federated learning, Peer to Peer, Communication Networks, Privacy, Machine Learning**

I. Introduction

FL is gaining popularity across many real-world AI applications due to its ability to preserve data privacy while enabling highly accurate Machine Learning (ML) models trained in a collaborative manner. It allows organizations to leverage the collective intelligence of distributed edge devices, revolutionizing industries like healthcare, finance, and the Industrial Internet of Things (IIoT). In the 5G/6G era, the impact of FL is expected to be even higher, benefiting from increased bandwidth and lower latency to enable real-time, distributed learning. Instead of transmitting raw data across networks, FL aggregates locally trained models from user devices. By adopting this approach, FL effectively mitigates privacy concerns associated with sensitive data. It is implemented in two prominent types of network architectures: central server-based and decentralised architectures [1]. Several FL approaches were published for both centralised server [2] and decentralised architectures [3].

Decentralised FL settings are proposed as a potential alternative to address the vulnerability of central server-based single-point failures and server-oriented attacks in the centralised FL setting. For instance, they include P2P protocols [4], to improve communication efficiency [5], and computational cost [6] or employ mechanisms such as secure multi-party computation (MPC) [7], DP [8] or model segmentation [9] to ensure privacy preservation. The major challenge arises when attempting to seamlessly integrate these individual updates into a cohesive and comprehensive solution. For instance, some approaches prioritise communication efficiency but may not sufficiently address security and privacy concerns or computational costs. They may make assumptions like participants are semi-honest, which may not be realistic in real-world scenarios. Conversely, while others focus on achieving adequate security measures, they may have limitations in communication, where these approaches would be valid for systems with only a limited number of peers. Therefore, in this paper, we aim to provide a P2P mechanism that can efficiently handle many clients with increased efficiency of aggregation of models, meanwhile ensuring privacy and security of the local model updates. For this, we propose a novel protocol that aims to provide a fully decentralised aggregation via a public P2P network.

A. Our Contributions

The key contributions from our paper are listed as follows:

- A novel distributed model training mechanism using distributed FL on top of a P2P network.
- Ensure that the proposed protocol securely communicates over the public P2P network.
- Implementation of our novel protocol on a real-world use case of supermarket recommendation system.
- Enables dynamic node discovery in a real P2P FL system on a smart cart platform, unlike existing approaches that assume pre-established connectivity.

B. Outline

The rest of the paper is arranged as follows: Section II presents related works investigated. Section III provides an overview of the system model. The architecture of the proposed system is presented in Section IV. Section V provides the methodology of the implementation. Experimental details are provided in Section VI. The paper is summarised and concluded in Section VII.

979-8-3503-2687-1/23/$31.00 ©2023 IEEE

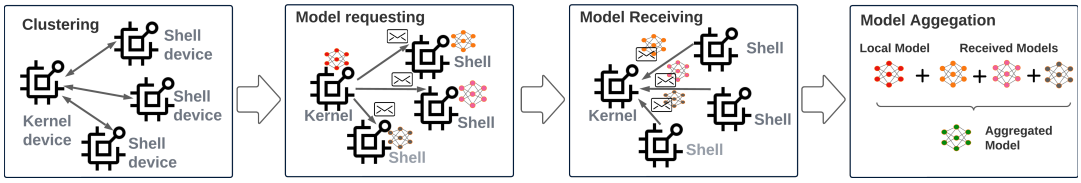

Fig. 1: Overview of the P2P FL protocol: KERNEL creates a cluster with SHELL peers. Then SHELL peers send model parameters to KERNEL. Next, KERNEL creates generalised model using the pre-trained and received model parameters.

II. Related Works

A. Centralised Federated Learning

The concept of FL was introduced by Google in 2016, focusing initially on a central server architecture. Many frameworks such as flower [10] are recently introduced for developing centralised FL systems. To enhance security, researchers have explored parameterized server architectures, leading to trade-offs between privacy and system performance. However, communication efficiency has often been the primary concern in centralised FL approaches. Striking a balance between privacy preservation, computational efficiency, and overall system performance remains crucial in the evolving field of FL.

B. Decentralised Federated Learning

Fully decentralised P2P FL has a high potential of enhancing privacy and communication efficiency over centralised FL. Existing P2P FL mechanisms focus on specific aspects like privacy or protocols. Comprehensive works addressing all aspects are still lacking. For example, even though the approach presented in [11] attempts to ensure the robustness of the aggregation procedure by detecting abnormal local gradients, still the mechanism may encounter privacy issues from backdoor attacks while attempting to preserve convergence guarantee. P2P Gossip protocol [5] is also customised according to different requirements of implementing a decentralised FL system. The approach introduced for gossip model segmentation mechanism in [9] to reduce training time and increase bandwidth utilization when achieving convergence. However, this convergence trend does not improve after a specific number of replicas. Thus, it limits the number of models segmenting replicas used.

Therefore, to balance the trade-offs over the model convergence, utility, and privacy, we propose a novel solution that enables privacy-preserved, fully decentralised FL among many peers. Furthermore, to show the practicality of our proposal, we implement the P2P system on a real-world use case scenario of a smart cart system.

III. System Model

An overview of our novel P2P FL protocol is shown in Fig. 1. This protocol has two types of peer nodes: SHELL and KERNEL peers. If a device hasn't gathered data for model training, it will be considered a SHELL device, while others will be classified as KERNEL devices. Continuous connection to the network is maintained by the SHELL devices. Upon request, their model parameters are sent to KERNEL peers. The KERNEL peers perform the aggregation process. If KERNEL requires an update to its local model from other models in the network, it requests a model from SHELL and downloads it. In the event that sufficient models are not received by the KERNEL peer, additional models are requested from another SHELL peer until enough models are obtained. This process is summarised in to four stages as follows:

- **Clustering** : The IoT devices, connected to the bridge module, form clusters with a single KERNEL peer and a selection of SHELL peers from the distributed network.
- **Model Requesting** : The KERNEL peer sends model request messages among the SHELL peers within the cluster.
- **Model Receiving** : When the SHELL peers receive a model request, they send the corresponding model parameters to the KERNEL peer.
- **Model Aggregating** : The KERNEL peer processes the received model parameters, aggregates them with its locally trained model, and creates a global model.

IV. Proposed Architecture

A. Novel P2P Federated Learning Process

The architecture of the proposed novel decentralised FL process is illustrated in Fig. 2. First, each peer checks whether it has a sufficient new quantity of locally collected dataset. If it is available, the peer assigns its role as a KERNEL. Otherwise, it is considered as a SHELL. Then, peers initiate the clustering process, ensuring that each cluster has only one KERNEL peer and the others are SHELL peers. This method is used to reduce the communication overhead. The KERNEL collects available local models from the SHELL peers connected to its cluster. During the model collection, it periodically checks the accuracy. If there is a large accuracy difference between the local model and the received model, KERNEL rejects the received model and removes that peer from the cluster. The process of collecting model parameters continues until the predefined requirements are fulfilled. Before the aggregation process, KERNEL adds differential privacy to the locally trained model to mitigate attacks on the training data when someone retrieves its local model. Next, aggregation is done by calculating the weighted average, where the weights are assigned based on each model's accuracy. Finally, KERNEL compresses the aggregated model and saves it in the cache memory, which is ready to be send it to the network. Then, the peer changes it role as a SHELL until the next round. This peer may get

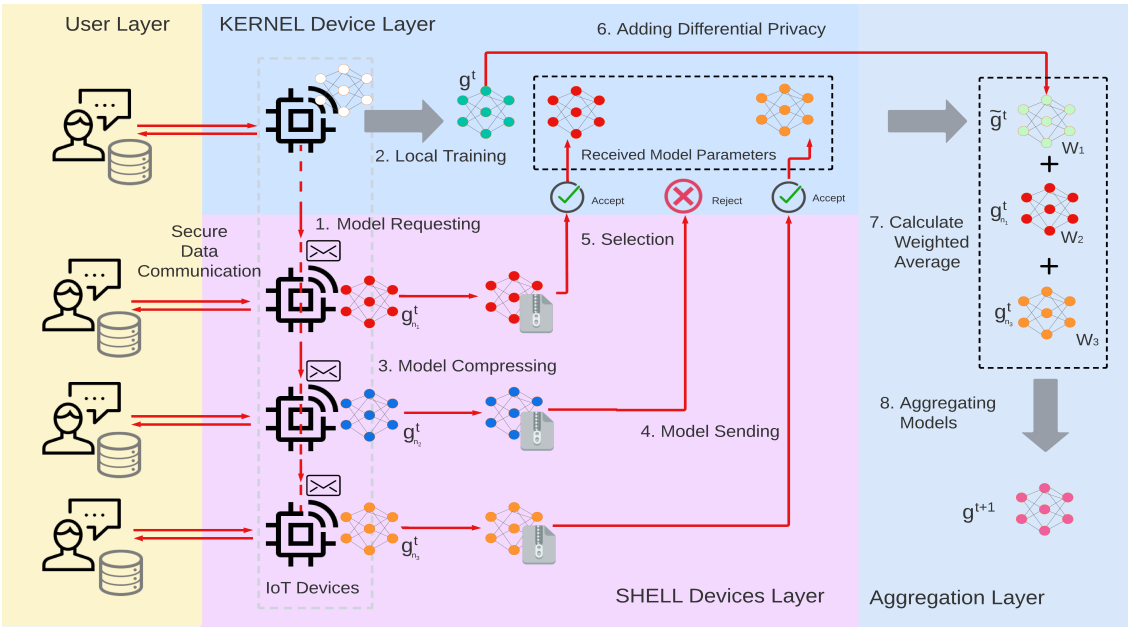

Fig. 2: The P2P FL process is as follows: (1) KERNEL peer sends a model request to selected SHELL peers; (2) each SHELL peer trains a local model using their respective dataset; (3) model parameters serialised and compressed; (4) SHELL peers send their model parameters; (5) KERNEL selects models to aggregate from the received models; (6) before running the aggregation process, DP is applied to the local model; (7) accuracy-based weights are calculated to determine the contribution of each model in obtaining the averaged model; (8) aggregated model is created by averaging all the models.

assigned when creating a new cluster with a different set of peers. This is continued until all peers get target accuracy.

B. P2P Communication Protocol

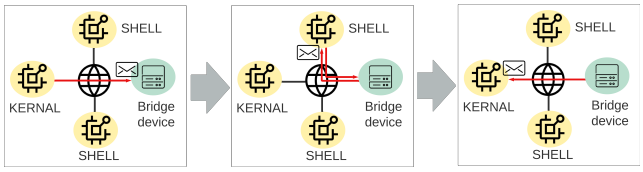

Fig. 3: In the system, the KERNEL peer sends a model parameter request to the bridge device. The bridge then forwards the request to the destination of the SHELL peer. The SHELL fulfills the request, and the bridge subsequently forwards response back to the KERNEL.

As per the aforementioned approach, main issues for accessing the public network (internet) to share trained model parameters from a SHELL to a KERNEL are the Network Address Translation (NAT) routers and firewalls. The most suitable solution for communication between peer devices is through an intermediary trusted device. This intermediary has a public static IP. Then, they can easily be located by any peer device from the internet. We define this intermediary device as the *bridge device*. As illustrated in Fig. 3, these devices have the capability to forward received data to the respective destinations. In this network, there can be multiple bridge devices that can act as intermediate devices.

C. Security and Privacy Mechanism

The user's personal data is collected on their mobile phone and shared among peers as model parameters. Here, the peers are the smart carts in a store, which are securely connected to the mobile phone. The smart cart acts as the first layer of privacy protection, ensuring that the user's data cannot be identified through the use of model parameters.

Algorithm 1 Aggregating recommendation models

Require: n = aggregation cluster size; p = parameter array; a = accuracy array; x = x test; y = y test;

function AGGREGATERECEIVEDMODELS(n,p,a,x,y):
  Set kernelModel to createModel()
  Set kernelModel weights to p[size(p)-1]
  Set model analysing Acc(x,y)
  for i = 0 to size(p) **do**
    $totalAcc$ = sum(a)
    $avgW = p[i] * a[i] + p[\text{size}(p) - 1] * a[\text{size(p)} - 1]$
    Set kernelModel weights to $avgW$
    Set a[size(p)-1] = accuracy of kernelModel
  end for
  return $avgW$
end function
Return averaged-weights

As illustrated in Algorithm 1, model parameters are shared in an aggregated form, making it impossible to recover data through these parameters. This is achieved through the use of an aggregation algorithm that incorporates more generalised models collected from other peer devices. This process serves as the second layer of privacy protection. In algorithm 1 the model analysis we tested received model accuracy with the local model. If the received model accuracy is in between

local model accuracy we accept that received model for the global model aggregation process.

The trained model further undergoes the addition of DP as the third defense layer before being used in the algorithm. This further ensures that the sharing parameters cannot retrieve the original training data, thus providing greater protection of the user's privacy.

V. Methodology

The primary objective of our work is to demonstrate a collaborative FL platform for privacy-preserved decentralised ML in the IIoTs. For this we implemented a simulation of a fully distributed P2P network. On top of the network, we run the proposed P2P FL platform. For the simulation of FL protocol, we selected a real IIoT use case. Therefore, we implement the proposed protocol by considering a supermarket smart cart system. The FL models are trained for a recommendation system in the supermarket app that provides user-specific recommendations based on purchases. As shown in Fig. 4, the smart cart behaves as an edge device in the network. In this network, there are two types of smart carts named child carts and parent carts. As previously discussed in Section IV, the parent cart has a bridge module with a public IP address. With this, a child cart does not necessarily need to have a public IP address. In the development of a P2P network comprising smart carts and mobile devices, after collecting enough data for model training, the smart cart connects to the network as a KERNEL and requests models from peer devices. Peer devices send the requested models to the smart cart.

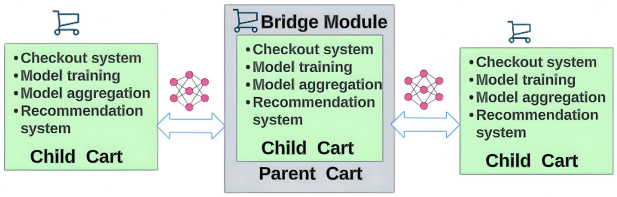

Fig. 4: Child carts and parent carts connection.

VI. Experiments Results

A. Experimental Settings

The experiments were conducted in both centralised and decentralised systems to evaluate the performance of our proposed protocol in comparison to existing FL algorithms. We employ both virtual and real physical environments for our experimentation, utilising docker containers for the virtual setup. In the virtual environment, we make a 6-peer network, while in real deployments, we use a 12-peer network. Our study focused on a Neural Network (NN)-based ML model, consisting of three dense layers with the Rectified Linear Unit (ReLU) activation function.

B. Dataset Information

Our NN model is trained using a supermarket dataset available in Kaggle [12]. Before training the ML model, the dataset undergoes a pre-processing step. The input parameters consist of customer gender, current date, and food items. For the testing dataset, we randomly selected 0.01% of the data from the selected dataset. With these pre-processing steps, the ML model is effectively trained using the supermarket dataset.

C. Convergence Performance of Novel protocol vs Centralised FL FedAvg Protocol in Virtual Environment

We test our proposed weighted average protocol in a decentralised network and compare it with the existing centralised FedAvg algorithm [2]. The experiments are conducted in a Docker virtual environment using a 6-peer network for both centralised and decentralised networks. We use neural network model for testing this experiment in both centralized and decentralized system. We trained the model for 10 epochs using a batch size of 128, and we specified a validation split of 20% of the training data. During this training process, we applied the Early Stopping callback. We repeat the experiment 12 times and calculate the average values of the results. Fig. 5 presents the results of the experiments. Our neural network model achieves convergence at an accuracy of 91.12%. The results indicate that our proposed method requires fewer iterations to reach convergence compared to the centralised network results. Based on these findings, we can conclude that our decentralised system performs well when compared to the centralised FedAvg algorithm.

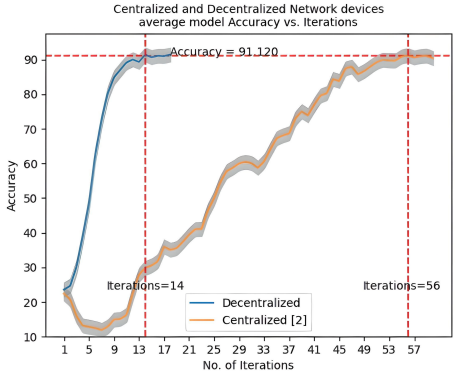

Fig. 5: Centralised federated averaging network convergence compared with decentralised weighted averaging algorithms.

D. Convergence Performance of Novel protocol vs Decentralised FL FedAvg Protocol in Virtual Environment

In this experiment, we compare our proposed weighted averaging system with the FedAvg algorithm in a decentralised network [5]. The experiment is conducted in a Docker virtual environment to obtain the test results. We utilise a decentralised weighted averaging 6-peer network and a decentralised fedAvg 6-peer network for the evaluation. For this test, we requested 2 models from the network for the model aggregation process and received a model aggregated

with the local model using a two-by-two model aggregation mechanism. The experiment is repeated 12 times, and the average details are calculated. The Fig. 6 illustrates the results obtained. According to the figure, both models achieve convergence at an accuracy of 91.12%. However, the weighted averaging system requires only 14 iterations to converge compared to the FedAvg method, which requires 29 iterations. We identify that the main reasons for this performance improvement are the utilization of model accuracy for weighted averaging and the filtering of biased models.

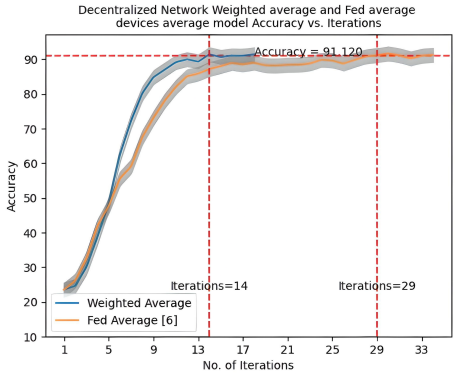

Fig. 6: Decentralised network weighted averaging and federated averaging algorithms convergence comparison.

E. Convergence Performance of Protocol in Real Environment

In this experiment, we conduct it in a real computer environment utilising 12 computers, each equipped with a Core i7 processor. P2P communication is established through the local network. We use a 3-layer NN Model for each and every computer and run the whole network. Then we gather convergence results of every computer. We measure the accuracy of the 12 devices and calculate the average values. According to Fig. 5, we observe that the FL system converges after 16 iterations. When comparing the results in a virtual environment in Fig. 8 to this real environment, there is an increase of 2 iterations. In the real environment, we identify network data losses when some KERNEL-requested models are not received due to the sender's network issues, such as latency. Therefore, such performance issues can cause additional iterations compared to a decentralised virtual environment. Nonetheless, our method still performs well when compared to both centralised and decentralised FedAvg methods.

F. Analysing the Impact of Cluster Size on Aggregation Time

In Fig. 7, we observe that increasing the cluster size results in longer processing times for processes such as data receiving, model analysis, and aggregation. Model receiving time, which is the time taken to receive all models from SHELL peers to KERNEL peers, varies linearly with increasing the cluster size and is also affected by network bandwidth, leading to varying times across different peers. The analysing time, which is the duration taken by the KERNEL to assess model suitability and accuracy, shows exponential variation with cluster size. When increasing cluster size, the received quantity of model parameters will increase, thereby increasing the number of analysing models. As a result, the analysis process takes an increased amount of time with an increase in cluster size. Therefore, maintaining a minimum cluster size is suggested to prevent efficiency compromise due to exponential variations in aggregation and convergence time.

G. Discussion

A comparison of the proposed method with existing pertinent works is presented in Table I. None of the existing works consider both the aspects of privacy-preservation and enhancing aggregation efficiency we present in our work. Based on the obtained results, we can consider that the proposed method outperforms both the existing centralised and decentralised networks while also achieving better convergence with a minimum number of iterations. The proposed decentralised method allows for separate aggregation, which enhances the accuracy of the model.

We have identified our proposed method has lower computational complexity compared to the centralised aggregation method because we aggregate model-limited models for end devices, and the model-sharing mechanism distributes the computation to other end devices. Unlike the centralised approach which requires all n clients for T rounds in the network to be aggregated, a KERNEL device only has to aggregate an average of n/k models, where k is the total number of KERNEL devices in the network. One KERNEL device only needs to perform aggregations only a limited $t; (t < T)$ rounds until they gather sufficient data to be switched to a SHELL. Therefore, the aggregation overhead is distributed over multiple devices. In experiments, when two KERNEL devices run two 6-peer networks, the aggregation overhead is reduced by 50% per aggregator, compared with a 12-client centralised aggregator.

These findings highlight the potential of the proposed concept and demonstrate its effectiveness in improving the accuracy of FL systems.

TABLE I: Comparison of our solution with existing works.

Characteristic	Ref. [13]	Ref. [14]	Ref. [15]	Ref. [16]	Ref. [17]	**Our Proposal**
Decentralised Model Aggregation	-	-	✓	✓	-	✓
Use Real-time data	✓	✓	✓	✓	✓	✓
Use of differential privacy	-	-	-	-	-	✓
Evaluate receiving model accuracy	-	-	-	-	-	✓
Aggregate model sequentially	-	-	-	-	-	✓
Peer forwards generalised model to the network	-	-	✓	✓	-	✓

VII. Conclusion

This research proposed and implemented a secure collaborative FL P2P system for model parameter sharing. The architecture is fully decentralised, encompassing both

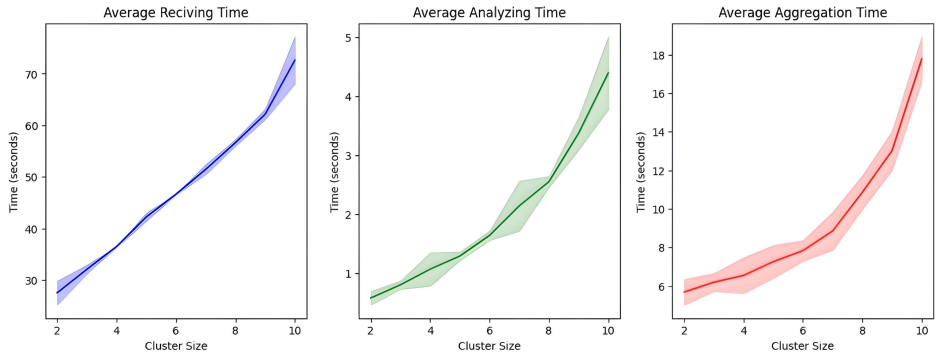

Fig. 7: Comparison of cluster size vs. receiving time, average analysing time, and average aggregation time.

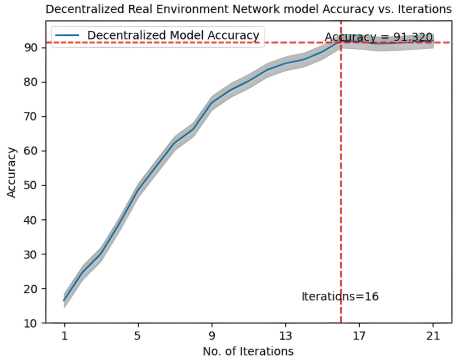

Fig. 8: Decentralised convergence in real environment.

model training and model aggregation. Security concerns are addressed through the application of differential privacy and generalized model building. Peer identification is achieved using a distributed hash table and routing table. Model aggregation is performed using an accuracy-based weighted averaging system. The protocol is evaluated in both real and virtual environments using real-world datasets. Based on the experimental results, our proposed system outperforms existing centralised and decentralised systems with similar aggregation accuracy over lesser iterations. Therefore, this protocol establishes a decentralised FL platform for model training and aggregation while ensuring user personal data security. Even though our protocol is implemented within an existing smart cart system, this generalised protocol can be applied to other decentralized FL systems, such as disease diagnosis prediction systems and fraud detection systems, among others, which will be investigated in future works.

Acknowledgment

This work has been partly supported by European Union in CONFIDENTIAL-6G (Grant No: 101096435), SPATIAL (Grant No: 101021808), and Science Foundation Ireland under CONNECT phase 2 (Grant no. 13/RC/2077_P2) projects.

References

[1] S. Abdulrahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5476–5497, 2021.

[2] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning*, DIDL '18, (New York, NY, USA), p. 1–8, Association for Computing Machinery, 2018.

[3] I. Hegedűs, G. Danner, and M. Jelasity, "Gossip learning as a decentralized alternative to federated learning," in *Distributed Applications and Interoperable Systems: 19th IFIP WG 6.1 International Conference, DAIS 2019, Held as Part of the 14th International Federated Conference on Distributed Computing Techniques, DisCoTec 2019, Kongens Lyngby, Denmark, June 17–21, 2019, Proceedings 19*, pp. 74–90, Springer, 2019.

[4] H. Ye, L. Liang, and G. Y. Li, "Decentralized federated learning with unreliable communications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 3, pp. 487–500, 2022.

[5] Z. Tang, S. Shi, B. Li, and X. Chu, "Gossipfl: A decentralized federated learning framework with sparsified and adaptive communication," *IEEE Transactions on Parallel and Distributed Systems*, 2022.

[6] W. Liu, L. Chen, and W. Zhang, "Decentralized federated learning: Balancing communication and computing costs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 8, pp. 131–143, 2022.

[7] T. Wink and Z. Nochta, "An approach for peer-to-peer federated learning," in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 150–157, 2021.

[8] S. Chen, D. Yu, Y. Zou, J. Yu, and X. Cheng, "Decentralized wireless federated learning with differential privacy," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6273–6282, 2022.

[9] C. Hu, J. Jiang, and Z. Wang, "Decentralized federated learning: A segmented gossip approach," *arXiv preprint arXiv:1908.07782*, 2019.

[10] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmão, *et al.*, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.

[11] C. Che, X. Li, C. Chen, X. He, and Z. Zheng, "A decentralized federated learning framework via committee mechanism with convergence guarantee," *IEEE Transactions on Parallel and Distributed Systems*, 2022.

[12] "Supermarket sales." kaggle.com/datasets/aungpyaeap/supermarket-sales.

[13] A. Mathur, "Federated Learning on Android devices with Flower." https://flower.dev/blog/2021-12-15-federated-learning-on-android-devices-with-flower/.

[14] "Federated Learning: Collaborative Machine Learning without Centralized Training Data." https://ai.googleblog.com/2017/04/federated-learning-collaborative.html, apr 6 2017.

[15] A. Lalitha, S. Shekhar, T. Javidi, and F. Koushanfar, "Fully decentralized federated learning," in *Third workshop on bayesian deep learning (NeurIPS)*, vol. 2, 2018.

[16] Z. Tang, S. Shi, B. Li, and X. Chu, "Gossipfl: A decentralized federated learning framework with sparsified and adaptive communication," *IEEE Transactions on Parallel and Distributed Systems*, 2023.

[17] "Acm Digital Library." dl.acm.org/doi/abs/10.1145/3286490.3286559.