# hFedLAP: A Hybrid Federated Learning to Enhance Peer-to-Peer

**Ismail Elshair**
Information Systems Department, Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia
ielshair@stu.kau.edu.sa (corresponding author)

**Tariq J. S. Khanzada**
Information Systems Department, Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia | Computer Systems Engineering Departments, Mehran UET, Pakistan
tkhanzada@kau.edu.sa

*Received: 23 March 2024 | Revised: 22 April 2024 | Accepted: 23 April 2024*

 *DOI: https://doi.org/10.48084/etasr.7331*

## ABSTRACT

**The concept of Federated Learning (FL) is a branch of Machine Learning (ML) that enables localized training of models without transferring data from local devices to a central server. FL can be categorized into two main topologies: Aggregation Server Topology (AST) and Peer-to-Peer (P2P). While FL offers advantages in terms of data privacy and decentralization, it also exhibits certain limitations in efficiency and bottleneck. However, the P2P topology does not require a server and allows only for a small number of devices. To overcome these limitations, this study proposes a hybrid FL Aggregation of P2P (hFedLAP) that mitigates some of the limitations of AST by combining it with P2P. This fusion model helps to remove the bottleneck and combines the advantages of both topologies. In the proposed hFedLAP model, clients are organized into 49 groups, each consisting of 51 clients, including one in each group serving as a client and an admin node in a P2P setup. In these groups, communication is restricted to admin nodes, supporting a maximum of 2,495 devices. Platform accuracy is maintained by implementing measures to prevent new devices with inadequate accuracy levels from joining until they attain the minimum required accuracy. The experimental results of hFedLAP were compared with AST and P2P using the MNIST dataset, showing that hFedLAP outperformed AST and P2P, achieving remarkable accuracy and scalability, with accuracy levels reaching 98.81%.**

***Keywords-federated learning; aggregation server; AST FL; peer-to-peer FL; machine learning; hybrid FL model***

## I. INTRODUCTION

Federated Learning (FL) was introduced in 2016 as a novel approach within the broader field of Machine Learning (ML) [1]. Many studies on ML aim to improve accuracy in prediction models and address some limitations, such as the centralization of training data, data privacy concerns, and the need to handle large amounts of data stored on a single server to enhance the reliability and accuracy of generated models [1-3]. Classic ML models are typically trained following a centralized technique in large data centers that are mostly isolated. However, with increasing the costs of computing resources and the need to ensure data privacy and security coupled with the advent of new data collection and processing techniques, FL has emerged as a viable alternative [4]. FL enables users to train their models locally and implicitly share the generated models [5]. FL ensures that all data remain on a local host and is a method for collectively training an ML model without the need to exchange local data. This training approach yields advantages in multiple aspects:

- Privacy-sensitive: Data are generally privacy-sensitive, which means that, for example, a hospital cannot share patient data with third parties (other hospitals) due to privacy policies and regulations [6]. The isolation concept of multiple parties is called Data Islands.
- Scalability: The FL concept requires multiple devices (ranging from tens to thousands), each with its own local computing resources/environment [6-7]. This approach not only eliminates the need for a large data center or expensive computing resources but also improves scalability and enables participants to train independently in parallel [7.
- Unbiased models: The models generated from training can be biased when the training is conducted with similar servers or when one server contains a disproportionately large amount of data, which can negatively affect other clients. This issue is prevalent in all topologies and directly affects the quality of generated models by decreasing the accuracy [7].

FedAvg is a typical FL algorithm, in which the models collected from the participants are aggregated on the server [1]. Traditionally, the relationship between the server and the devices is categorized into two main FL topologies with unique advantages: Aggregation Server Topology (AST) and Peer-to-Peer (P2P). AST offers structured and centralized aggregation between millions of devices, ensuring easier global model updates. On the other hand, P2P facilitates direct node-to-node communication, making it capable of adjusting to changes. However, both approaches come with their respective challenges. The centralized structure of AST can become a bottleneck due to its dependence on a single server, whereas P2P has limitations in maintaining the accuracy of generated models and the scalability of the framework. According to [7], the maximum number of devices within a P2P platform is limited to 100 clients. The convergence of these two approaches offers an opportunity to leverage their respective advantages and mitigate their weaknesses. This study not only elucidates the advantages of the proposed hybrid model, but also evaluates its efficiency compared to traditional FL topologies. The primary contributions of this work can be outlined as follows:

- Introduce the novel hFedLAP, a hybrid FL model that seamlessly merges the robustness of the AST and P2P topologies. This innovative approach increases prediction accuracy and also provides a unique mechanism whereby incoming devices must attain a predetermined accuracy threshold to participate in training. Furthermore, hFedLAP judiciously circumvents the challenges posed by an extensive number of devices in P2P configurations, representing a significant step forward in scalable and efficient FL.
- Evaluate the accuracy and F1-score of hFedLAP and compare it against AST and P2P topologies.

## II. RELATED WORKS

Centralized FL [1] has addressed some of the challenges in ML by training data on mobile devices without compromising user privacy. The FedAvg algorithm collects and averages all shared models on a server and then shares the global model with the participant devices. Several implementations and algorithms have been introduced for the aggregation server. In [8], the FL concept was applied using the FedAvg algorithm on user input on mobile devices from 7.5 B sentences to enhance the prediction of words while typing. In [9], a personalized FL framework, called pFedLA, was introduced, dubbing layer-wise personalized FL. pFedLA aims to personalize clients with heterogeneous data by assigning unique weights and dedicated hypernetwork parameters on the server for each client. When applied to several datasets, such as EMNIST and CIFAR100, pFedLA exhibits higher performance and higher aggregated weight compared to other customized FL frameworks. Centralized FL includes different topologies, such as:

- AST: The training process depends on aggregating models directly into a single server, where devices share local models with the server. The server collects and aggregates models to share a global model with the devices, as noticed in Figure 1 [4, 8-11].
- The hierarchical topology depends on adding additional layer(s) between the server and clients (edge nodes) to reduce direct communication between them, as observed in Figure 1 [12-14].

Decentralized FL has emerged as an alternative to previously introduced models to eliminate the dependence on a central server. This serverless architecture relies on direct communication between clients [7]. It provides users with greater autonomy in process control and reduces the risks of bottlenecks and data leakage caused by server issues [8]. Furthermore, decentralized learning supports various topologies, namely:

- P2P topology: The training process depends on direct communication between participant nodes, where each node communicates and shares a local model [15-16], as portrayed in Figure 1, where all devices are capable of communicating, sharing, and aggregating local models directly.
- Ring topology: All nodes within the same platform communicate directly with the neighbor node, where each node is linked to two nodes, predecessor and successor [17-18], as depicted in Figure 1, and each device has its local model while being linked and connected circularly.

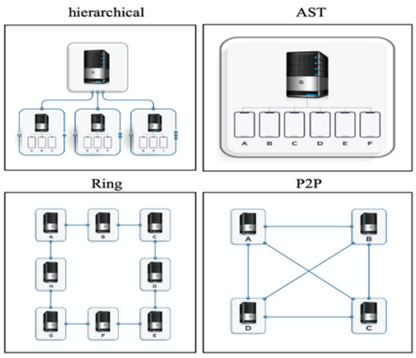


Fig. 1. FL topologies and frameworks.

Typical decentralized learning can be found mainly in healthcare institutes, banking, and industries that have high-specification servers and require additional security measures due to data sensitivity [19]. BrainTorrent [16] is an FL framework that does not require a central server. This model was applied to the Multi-Atlas Labeling Challenge (MALC) dataset that contains 30 manually labeled brain MRI scans. BrainTorrent training results demonstrated similar performance compared to the training on data collected on a single server, which is considered a classic ML. In [13], the Hierarchical Federated Averaging (HierFAVG) algorithm, which incorporates an edge layer to aggregate the results of clients before sharing them with the server, was introduced. This model was evaluated in the MNIST dataset, achieving accuracies of approximately 96.2% and 95% for the EDGE-IID and EDGE-NIID scenarios, respectively. In [20], Federated Knowledge Transfer (FedKT) was proposed, utilizing knowledge transfer techniques to distribute the subsets of the original dataset between client devices, achieving 90.5%

accuracy in the MNIST dataset. The Iterative Federated Clustering Algorithm (IFCA) [21] enables the creation of user clusters that exchange global models. This framework assigns devices to clusters with similar devices, and the evaluation on the MNIST dataset yielded accuracies up to 96%. In [22], a hybrid classifier was introduced for hospital systems, where each server trains multiple models locally and then averages the results of these models from multiple local servers to generate the final version. This is considered a hybrid approach, where each silo aggregates models and then adds another layer to generate a global average model that leads to increased accuracy and reduces biases when averaging generated results. In [23], the Federated Hierarchical Synchronous platform (FedHiSyn) was proposed by combining the ring and hierarchical topologies into a hybrid model, where devices, which are clustered based on computational capability, train models locally and share the generated models with an edge server. In [24], the concept of Multi-stage Hybrid FL (MH-FL) was presented.

This study extends the FL framework into a multilayer network structure, featuring multiple layers of nodes between end devices and the central server. This model has a multilayer structure, where the middle layer communicates vertically with either higher or lower devices. This setup requires having a main server to aggregate models at the top level, which is not demanded in hFedLAP as it communicates horizontally. In the proposed conceptual framework, the admin node compiles models from various clients within the group, operating in a P2P relationship. Contrary to the method presented in [25], this aggregation does not aim to develop a global model. Additionally, this method does not require the server to distribute the model among all members of the cluster. Moreover, the proposed approach intentionally avoids categorizing clients based on data similarity, a practice that could potentially lead to bias in the model.

## III. PROPOSED HYBRID APPROACH

### A. *Hybrid Approach Design*

The proposed hFedLAP framework is a hybrid approach that combines both P2P and centralized FL mechanisms. The proposed framework overcomes the challenge of not having a server in P2P by selecting one of the participant nodes to act as the admin node, responsible for administering the platform. Initially, direct training occurs between the participant nodes. After a predefined number of training rounds, all nodes share their last accuracy results, and the node with the highest accuracy is selected as the admin node for the group. The admin node, functions as a regular node but with the additional functionality of monitoring the platform without centralizing models as in AST. However, in P2P, all clients can be considered servers with high hardware specifications [26], indicating that any client has the potential to operate as an admin node. To increase the number of devices, participant devices are organized into groups consisting of multiple devices, each of them monitored by the admin node in the group. The maximum number of clients in a single group is limited to 51 devices. Once the number of devices exceeds 51, a new group is formed with new joiners and an admin node. In the hFedLAP framework, only the admin node is capable of communicating outside the group with other admin nodes. Therefore, the total number of nodes can be increased to more than 100 devices, which is considered the maximum number of clients in P2P [7]. Furthermore, to overcome the challenge of a possible bottleneck that can be generated by having a single server in AST, hFedLAP prepares another device as a backup in case the primary admin client goes down, thus reducing the bottleneck. The chosen backup node is the one with the second-highest accuracy in its group. Moreover, adding a new client to the group is one of the optimizations in hFedLAP. The admin node will first share the latest model and then train the new clients individually to measure and increase their accuracy. If a client fails to reach an accuracy of 90%, it will not be allowed to join the group and will wait until the new round is completed. The process is then repeated, as shown in Figure 2. The admin nodes are capable of directly communicating with each other to share global models between groups. This process maintains the mean accuracy between groups and reduces the bias generated by having isolated groups. Table I introduces some key notations used in the algorithm's structure and operations.

TABLE I. NOTATIONS OF HFEDLAP

| Notation | Meaning |
|---|---|
| $K$ | The number of initial clients |
| $K_{add}$ | The number of additional clients that can join the loop |
| $N_K$ | The number of random clients (set of clients noted $S$) used for aggregation with each client. If one wants to select all the other clients for aggregation (or most likely set to a percentage set to choose a random set of clients) |
| $n_{init}$ | Number of initial data instances for client $D_i$ |
| $n_{current}$ | Number of current data instances for client $D_i$ |
| $n_{add}$ | Number of data instances added when the devices are augmented |
| *active_users* | Set of active users |
| *pending_users* | Set of inactive users waiting to be potentially added to the loop |
| *global_epochs* | number of global iterations |
| *p2p_epochs* | number of peer-to-peer epochs |
| *agg_epochs* | number of aggregation epochs |
| *local_epochs* | number of local iterations (number of epochs for each client training in one global epoch) |
| *new_threshold* | a probability threshold to decide whether to add a new client or not |
| *Perf_threshold* | performance threshold to decide whether a performance score is good enough |

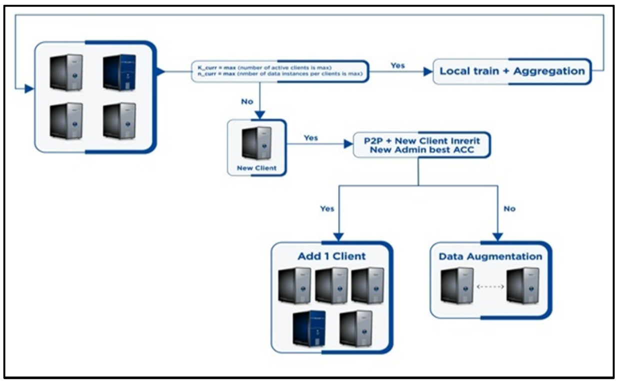


Fig. 2. hFedLAP Framework.

ALGORITHM 1: HYBRID APPROACH

```
procedure HYBRID (active_users, pending_users, K, K_add, N_k, n_init, n_current, n_add, global_epochs, p2p_epochs,
agg_epochs, local_epochs, new_threshhold, perf _threshhold)
do_peer_to_peer, pending_device, device_added, new_client ← True, False, False, False
admin_client, n_current ← 0, n_init (initialization of the index of administrator node)
for global_epoch = 1 : global_epochs do :
  if n_current reaches the maximum possible (all possible data instances were added) or K_current
  reaches the maximum possible number of clients (all possible devices were added) then:
    for client_idx = 1 : n_current do:
      Local train of the client client_idx for local_epochs
      Aggregate client models to admin node admin_client and send admin model weights to all clients
  else
    if peer_to_peer = True then:
      for p2p_epoch = 1 : p2p_epochs do:
        for client_idx = 1 : n_current do:
          Local train the client client_idx for local_epochs
          Each device is peer-to-peer aggregated with respective random N_k clients
    else:
      for agg_epoch = 1 : agg_epochs do:
        for client_idx = 1 : n_current do:
          Local train the client client_idx for local_epochs
          Aggregate client models to admin node admin_client and send admin model weights to all clients
    if pending_device = True then:
      new_client ← True (proceed to test whether a new client is eligible to be added to active_clients)
    else:
      p ← random(0, 1) (random float from 0 to 1)
      if p < perf_threshhold then:
        new_client ← True
      else:
        new_client ← False
    if new_client = True then:
      new_client model inherits the weights from admin node and tests the accuracy on validation data
      (val_acc)
      if val_acc > perf_threshhold then:
        add the new client to active_clients
        do_peer_to_peer, pending_device ← True, False
      else:
        n_current ← n_current  + n_add (do data augmentation of all clients)
        do_peer_to_peer, pending_device ← True, True
    else:
      do_peer_to_peer, pending_device ← False, False
end
```

## B. Hybrid approach algorithm

Within the architecture of hFedLAP, the training process is a local training and aggregating in P2P form. Models are exchanged among devices that form a group. Algorithm 1 displays the verification process of adding new clients to the group and the training process.

## C. The Prediction Model Architecture

The defined model is a Convolutional Neural Network (CNN) composed of multiple stacked layers:

- 2D-Convolution layer: input channels = 1, output channels = 10, kernel size = 5.
- 2D-Convolution layer: input channels = 10, output channels = 20, kernel size = 5.
- Dropout layer: p = 0.5
- Linear layer: input dimension = 320, output dimension = 50.
- Linear layer: input dimension = 50, output dimension = num_classes = 10.

## D. Validation Metrics

Since the dataset is equally and identically distributed among the users, accuracy and F1-score are appropriate metrics to evaluate the performance of the proposed methods using the following equations:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$F1-score = \frac{2\,x\,(Precision\ x\ Recall)}{Precision + Recall} \quad (2)$$

True Positives (*TP*), True Negatives (*TN*), False Negatives (*FN*), and False Positives (*FP*) are defined to calculate accuracy and evaluate performance. The higher the accuracy and the F1-score is, the better the model is perceived. The average accuracy is used to evaluate the whole system, composed of multiple clients.

# IV. EXPERIMENTS

The PyTorch platform was employed to implement the introduced model, with hardware specifications as follows: Intel Xeon CPU with 2 virtual CPUs and 13GB of RAM and NVIDIA Tesla K80 GPU with 12GB of VRAM. The MNIST

[27] dataset, which contains 28×28 pixels grayscale images of handwritten digits labeled 0 to 9, almost evenly distributed, was utilized. MNIST is made up of 60,000 train images and 10,000 test images. This experiment deploys the train instances for training and the test ones for validation. Both sets are equally divided between the clients participating in FL.

### A. Centralized Federated Learning

Table II provides the training results for the centralized FL approach. Four isolated experiments were carried out, C1, C2, C3, and C4. The difference between the four experiments was in device size (8k or 10k) and the number of devices (4 or 6). Having more data points per client and more clients tends to result in better performance, up to a certain extent.

TABLE II. CENTRALIZED FL FRAMEWORK RESULTS

| Parameters | C1 | C2 | C3 | C4 |
|---|---|---|---|---|
| $K$ | 6 | 6 | 4 | 4 |
| $N_k$ | 3 | 3 | 2 | 2 |
| $n$ | 10k+1.67k | 8k+1.67k | 10k+1.67k | 8k+1.67k |
| $N$ | 60k+10k | 48k+10k | 40k+6.67k | 32k + 6.67k |
| *global_epochs* | 6 | 6 | 6 | 6 |
| *local_epochs* | 4 | 4 | 4 | 4 |
| F1-score | 95.07% | 95.02% | 95.92% | 95.87% |
| Accuracy | 95.08% | 95.04% | 95.96% | 95.90% |

### B. Decentralized P2P Federated Learning

Similarly to centralized FL, the parameters in P2P FL were fixed, except for the device size and number of devices (D stands for decentralized). Table II manifests the results.

TABLE III. DECENTRALIZED FL FRAMEWORK RESULTS

| Parameters | D1 | D2 | D3 | D4 |
|---|---|---|---|---|
| $K$ | 6 | 6 | 4 | 4 |
| $N_k$ | 3 | 3 | 2 | 2 |
| $n_i$ | 10k+1.67k | 8k+1.67k | 10k+1.67k | 8k+1.67k |
| $n$ | 60k+10k | 48k+10k | 40k+6.67k | 32k + 6.67k |
| *p2p_epochs* | 6 | 6 | 6 | 6 |
| *local_epochs* | 4 | 4 | 4 | 4 |
| F1-score | 96.26% | 96.18% | 96.23% | 96.30% |
| Accuracy | 96.28% | 96.19% | 96.25% | 96.31% |

### C. Hybrid Approach hFedLAP

The proposed hFedLAP approach was followed with a changed set of variables: *agg_epochs*, *p2p_epochs*, and *global_epochs* for simplicity, and both *agg_epochs* and *p2p_epochs* were not running at the same time. *agg_epochs* was equal to *p2p_epochs*, and the rest of the variables were fixed. The 2-3 in $N_k$ means that the selected number of clients in the FedAvg or P2P aggregation is either 2 or 3, depending on the number of active devices, which varies from 4 to 6. In addition, for all experiments $N_{init}$ was 8K + 1.67K, *perf_threshold* = 0.98, *new_threshold* = 0.8, *global_epochs* = 10, and *local_epoch* = 4. Table IV and Figure 3 portray the results of the proposed hybrid method. Specific configurations were observed, particularly for H1 (*p2p_epochs* = *agg_epochs* = 6) and H2 (*p2p_epochs* = *agg_epochs* = 3), where no data augmentation was activated at any global epoch. This implies that the admin server could train and achieve over 98% validation accuracy for every newly added client from the first global epoch to the last. In contrast, for H3 (*p2p_epochs* = *agg_epochs* = 2), instances were observed where the new device validation accuracy fell below 98%. This necessitated the activation of data augmentation to train the active clients, thereby enhancing the effectiveness of the admin node model. This requirement for data augmentation in H3 can be attributed to the insufficiency of just two *p2p_epochs* to attain a 98% accuracy on new data, particularly evident from the third global model onwards. The parameter combination H3 facilitates greater participation of data instances in the experiment through progressive data augmentation compared to H1 and H2 during training. It was noticed that accuracy tends to improve with the addition of more data in active learning scenarios. This suggests that H1 and H2 may not require data augmentation, as active clients achieve desired accuracies within *p2p_epochs* = *agg_epochs* = 6.

H1 exhibits superior performance, although at the cost of increased aggregation/P2P iterations and without utilizing data augmentation. H2 and H3 offer more balanced configurations. The H4 experiments were analogous to H3 (2 epochs) but with varying *n_add* values. Experimenting with a higher *n_add* could accelerate the meeting of performance criteria. For instance, a new device A might reach the 98% accuracy threshold after a single augmentation of 1000 instances, rather than after two augmentations of 500 instances each. This constitutes a more rapid data augmentation approach, entailing fewer augmentation iterations. However, the experiments indicate that configuration H4 is optimal assuming that only two data augmentations (1k + 1k) would suffice for all clients to meet the desired threshold is speculative, particularly if more than two clients are added.

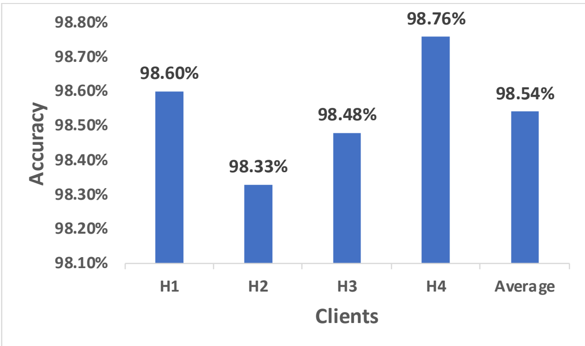


Fig. 3. Accuracy of hFedLAP FL models.

TABLE IV. HFEDLAP FRAMEWORK RESULTS

| Parameters | H1 | H2 | H3 | H4 |
|---|---|---|---|---|
| $K_{init}$ | 4 | 4 | 4 | 4 |
| $K_{add}$ | 2 | 2 | 2 | 2 |
| $K$ | 6 | 6 | 6 | 6 |
| $N_k$ | 2-3 | 2-3 | 2-3 | 2-3 |
| $n_{add}$ | 500 | 500 | 500 | 1000 |
| *p2p_epochs* | 6 | 3 | 2 | 2 |
| *agg_epochs* | 6 | 3 | 2 | 2 |
| F1 | 98.58% | 98.31% | 98.47% | 98.74% |
| Accuracy | 98.60% | 98.33% | 98.48% | 98.76% |

### *D. Result Comparison*

To make the comparison as fair as possible, the same specifications were defined for all approaches and then the experiment was run on the same dataset (MNIST). In addition, similar data sizes were applied in each local epoch and round. The average accuracy of each approach was compared, as demonstrated in Figure 4, indicating that the proposed hybrid approach reached a higher average accuracy compared to the others.

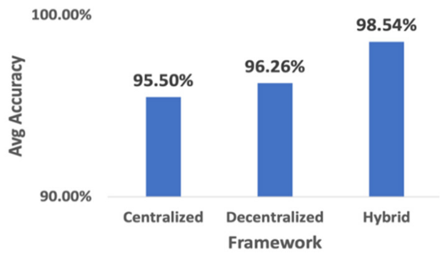


Fig. 4. Average accuracy comparison in centralized, decentralized, and hFedLAP.

### *E. Evaluating hFedLAP Performance and Scalability*

Setting the P2P epochs and aggregation epochs to 4 allows the local clients to aggregate and achieve consistent performance for the admin node across the epochs. To measure the performance of a single group, the model was run using a total of 6, 12, and 16 devices. The size of the extra devices was set to 500, 250, and 187, respectively, to allow the system to perform data augmentation up to 4 times, if necessary, as depicted in Table V.

TABLE V. HFEDLAP ACCURACY AND F1-SCORES

| Parameters | 6 | 12 | 16 |
|---|---|---|---|
| $K_{init}$ | 4 | 8 | 11 |
| $K_{add}$ | 2 | 4 | 5 |
| $K$ | 6 | 12 | 16 |
| $N_{add}$ | 500 | 250 | 187 |
| F1-score | 0.98415 | 0.9878 | 0.9843 |
| Accuracy | 0.98419 | 0.9881 | 0.9845 |

Furthermore, each group had initial devices and additional devices joined during training. It was observed that the accuracy improved as the number of clients increased. This improvement can be attributed to the reduced amount of data for each client, resulting in an increased number of global rounds and, consequently, more aggregation between the clients. Figure 5 also displays the F1-score and accuracy for 6 global epochs. Essentially, in principle, hFedLAP demonstrates its potential to extend the total device capacity up to 2,495 devices. This expansion is achieved by enabling each admin node to establish communication with a restricted subset of 100 devices (comprising 51 devices within the same subgroup and 49 admin nodes in other subgroups). In particular, the admin node can exclusively engage in cross-subgroup communication, connecting with fellow admin nodes located in distinct subgroups, as shown in Figure 6. However, due to MNIST being a comparatively small dataset, the experiments were restricted to a maximum of 16 clients.

(a)

(b)

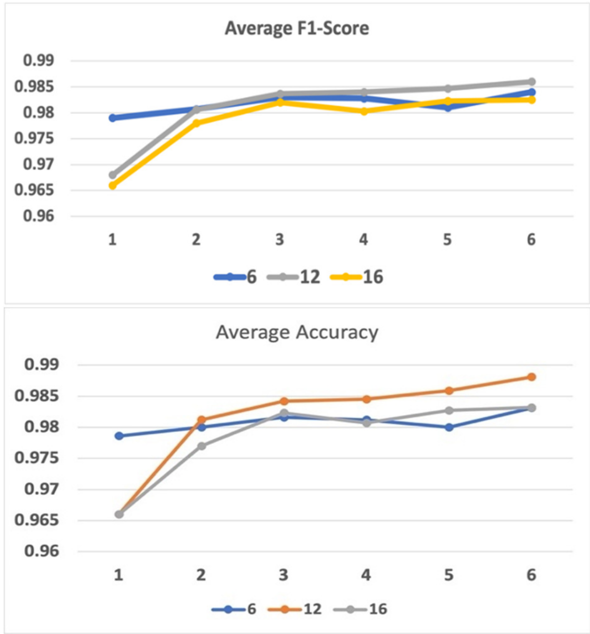


Fig. 5. Accuracy and F1-score of the different configurations (6 global epochs).

Fig. 6. Scalability in hFedLAP.

## V. CONCLUSION

This study introduced hFedLAP, a hybrid FL approach that enhances scalability, reliability, and accuracy by combining features of both AST and P2P models. By strategically grouping devices, hFedLAP not only mitigates the scalability limitations typically observed in P2P, but also prevents bottlenecks in AST and ensures system resilience during server outages. The fusion is applied to have 49 groups, each having 51 clients, supporting 2495 devices in total. The results demonstrated that hFedLAP achieved an accuracy of 98.81%, whereas the classical models AST and P2P attained 95.96% and 96.31%, respectively. In particular, the strategy of limiting new device additions based on the accuracy obtained after the first epoch has proven to be effective in maintaining and even enhancing the overall platform accuracy while reducing the impact on the existing performance metrics. This study addresses a significant knowledge gap in the scalability and reliability of federated learning systems, particularly in heterogeneous network environments. Compared to existing works, hFedLAP introduces a novel model that robustly supports an increased number of devices while improving accuracy. In future work, this research aims to further evaluate the time efficiency and scalability of hFedLAP by testing it on larger and higher-resolution datasets using a network of 2,495 devices to validate the model's effectiveness on a larger scale and more importantly to explore potential optimizations that could benefit a broader range of applications in FL.

## ACKNOWLEDGMENT

The authors wish to thank the Information Systems Department, Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia.

## FUNDING

This research work was funded by the Institutional Fund Project under grant no. (IFPIP-620-611-1443), the Ministry of Education and King Abdulaziz University (KAU), and the Deanship of Scientific Research (DSR) of KAU, Jeddah, Saudi Arabia.

## REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, Apr. 2017, pp. 1273–1282.

[2] H. A. Owida, A. Al-Ghraibah, and M. Altayeb, "Classification of Chest X-Ray Images using Wavelet and MFCC Features and Support Vector Machine Classifier," *Engineering, Technology & Applied Science Research*, vol. 11, no. 4, pp. 7296–7301, Aug. 2021, https://doi.org/10.48084/etasr.4123.

[3] B. K. Ponukumati, P. Sinha, M. K. Maharana, A. V. P. Kumar, and A. Karthik, "An Intelligent Fault Detection and Classification Scheme for Distribution Lines Using Machine Learning," *Engineering, Technology & Applied Science Research*, vol. 12, no. 4, pp. 8972–8977, Aug. 2022, https://doi.org/10.48084/etasr.5107.

[4] W. Li *et al.*, "Privacy-Preserving Federated Brain Tumour Segmentation," in *Machine Learning in Medical Imaging*, Shenzhen, China, 2019, pp. 133–141, https://doi.org/10.1007/978-3-030-32692-0_16.

[5] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated Machine Learning: Concept and Applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, Jan. 2019, Art. no. 12, https://doi.org/10.1145/3298981.

[6] Q. Li *et al.*, "A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3347–3366, Apr. 2023, https://doi.org/10.1109/TKDE.2021.3124599.

[7] P. Kairouz *et al.*, "Advances and Open Problems in Federated Learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1-2, pp. 1–210, Jun. 2021, https://doi.org/10.1561/2200000083.

[8] A. Hard *et al.*, "Federated Learning for Mobile Keyboard Prediction." arXiv, Feb. 28, 2019, https://doi.org/10.48550/arXiv.1811.03604.

[9] X. Ma, J. Zhang, S. Guo, and W. Xu, "Layer-wised Model Aggregation for Personalized Federated Learning," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 10082–10091, https://doi.org/10.1109/CVPR52688.2022.00985.

[10] J. Goetz, K. Malik, D. Bui, S. Moon, H. Liu, and A. Kumar, "Active Federated Learning." arXiv, Sep. 27, 2019, https://doi.org/10.48550/arXiv.1909.12641.

[11] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated Learning with Matched Averaging." arXiv, Feb. 15, 2020, https://doi.org/10.48550/arXiv.2002.06440.

[12] N. Mhaisen, A. A. Abdellatif, A. Mohamed, A. Erbad, and M. Guizani, "Optimal User-Edge Assignment in Hierarchical Federated Learning Based on Statistical Properties and Network Topology Constraints," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 55–66, Jan. 2022, https://doi.org/10.1109/TNSE.2021.3053588.

[13] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-Edge-Cloud Hierarchical Federated Learning," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Dublin, Ireland, Jun. 2020, pp. 1–6, https://doi.org/10.1109/ICC40277.2020.9148862.

[14] S. Hosseinalipour *et al.*, "Multi-Stage Hybrid Federated Learning Over Large-Scale D2D-Enabled Fog Networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1569–1584, Aug. 2022, https://doi.org/10.1109/TNET.2022.3143495.

[15] Y. Zhao *et al.*, "Privacy-Preserving Blockchain-Based Federated Learning for IoT Devices," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1817–1829, Oct. 2021, https://doi.org/10.1109/JIOT.2020.3017377.

[16] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "BrainTorrent: A Peer-to-Peer Environment for Decentralized Federated Learning." arXiv, May 16, 2019, https://doi.org/10.48550/arXiv.1905.06731.

[17] J. Lee, J. Oh, S. Lim, S. Y. Yun, and J. G. Lee, "TornadoAggregate: Accurate and Scalable Federated Learning via the Ring-Based Architecture." arXiv, Oct. 01, 2021, https://doi.org/10.48550/arXiv.2012.03214.

[18] Z. Wang, Y. Hu, J. Xiao, and C. Wu, "Efficient Ring-topology Decentralized Federated Learning with Deep Generative Models for Industrial Artificial Intelligent." arXiv, Apr. 15, 2021, https://doi.org/10.48550/arXiv.2104.08100.

[19] O. Marfoq, C. Xu, G. Neglia, and R. Vidal, "Throughput-Optimal Topology Design for Cross-Silo Federated Learning," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 19478–19487.

[20] Q. Li, B. He, and D. Song, "Practical One-Shot Federated Learning for Cross-Silo Setting." arXiv, May 20, 2021, https://doi.org/10.48550/arXiv.2010.01017.

[21] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An Efficient Framework for Clustered Federated Learning," *IEEE Transactions on Information Theory*, vol. 68, no. 12, pp. 8076–8091, Sep. 2022, https://doi.org/10.1109/TIT.2022.3192506.

[22] M. M. Yaqoob, M. Nazir, M. A. Khan, S. Qureshi, and A. Al-Rasheed, "Hybrid Classifier-Based Federated Learning in Health Service Providers for Cardiovascular Disease Prediction," *Applied Sciences*, vol. 13, no. 3, Jan. 2023, https://doi.org/10.3390/app13031911.

[23] G. Li *et al.*, "FedHiSyn: A Hierarchical Synchronous Federated Learning Framework for Resource and Data Heterogeneity," in *Proceedings of the 51st International Conference on Parallel Processing*, Bordeaux, France, Sep. 2022, pp. 1–11, https://doi.org/10.1145/3545008.3545065.

[24] S. Hosseinalipour *et al.*, "Multi-Stage Hybrid Federated Learning Over Large-Scale D2D-Enabled Fog Networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1569–1584, Dec. 2022, https://doi.org/10.1109/TNET.2022.3143495.

[25] M. Cao, Y. Zhang, Z. Ma, and M. Zhao, "C2S: Class-aware client selection for effective aggregation in federated learning," *High-Confidence Computing*, vol. 2, no. 3, Sep. 2022, Art. no. 100068, https://doi.org/10.1016/j.hcc.2022.100068.

[26] S. Wang *et al.*, "Adaptive Federated Learning in Resource Constrained Edge Computing Systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, Mar. 2019, https://doi.org/10.1109/JSAC.2019.2904348.

[27] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, https://doi.org/10.1109/5.726791.