Pattern Recognition Letters 177 (2024) 121–127

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Clustered FedStack: Intermediate Global Models with Bayesian Information Criterion

Thanveer Shaik [a,*], Xiaohui Tao [a], Lin Li [b], Niall Higgins [c,d], Raj Gururajan [e], Xujuan Zhou [e], Jianming Yong [e]

[a] School of Mathematics, Physics and Computing, University of Southern Queensland, Toowoomba, Australia
[b] Wuhan University of Technology, Wuhan, China
[c] Royal Brisbane and Women's Hospital, Brisbane, Australia
[d] Queensland University of Technology, Brisbane, Australia
[e] School of Business, University of Southern Queensland, Springfield, Australia

ARTICLE INFO

Editor: Li Liu

Dataset link: https://archive.ics.uci.edu/ml/datasets/PPG-DaLiA

Keywords:
Federated learning
FedStack
Clustering
Bayesian
Cyclical learning rates

ABSTRACT

Federated Learning (FL) is currently one of the most popular technologies in the field of Artificial Intelligence (AI) due to its collaborative learning and ability to preserve client privacy. However, it faces challenges such as non-identically and non-independently distributed (non-IID) data with imbalanced labels among local clients. To address these limitations, the research community has explored various approaches such as using local model parameters, federated generative adversarial learning, and federated representation learning. In our study, we propose a novel Clustered FedStack framework based on the previously published Stacked Federated Learning (FedStack) framework. Here, the local clients send their model predictions and output layer weights to a server, which then builds a robust global model. This global model clusters the local clients based on their output layer weights using a clustering mechanism. We adopt three clustering mechanisms, namely K-Means, Agglomerative, and Gaussian Mixture Models, into the framework and evaluate their performance. Bayesian Information Criterion (BIC) is used with the maximum likelihood function to determine the number of clusters. Our results show that Clustered FedStack models outperform baseline models with clustering mechanisms. To estimate the convergence of our proposed framework, we use Cyclical learning rates.

1. Introduction

As AI techniques have matured, a vast amount of human data is being generated every second around the world. To manage this huge data, technology giant Google introduced a mechanism that trains a machine learning (ML) algorithm across multiple decentralized devices or servers without exchanging their local data samples. This is called Federated Learning (FL), which is also known as collaborative learning [1]. FL overcomes the issues of data privacy that exist in traditional centralized learning techniques where all device or server data is merged for analysis [2]. FL has garnered significant attention since its introduction by Google as a ML technique for predicting users' input from Gboard (a keypad) on Android devices. This technique has been widely adopted in communication, engineering, and healthcare. However, medical institutes in particular possess a vast amount of patient data that may not be sufficient to train ML or deep learning models, and may even be biased due to a lack of data diversity. FL addresses this issue through its collaborative learning approach, where local models trained in each medical institute share their model weights with a global model stored in a shared server [3]. This maintains data privacy, as the institute's data remains within its premises. The process can be used at the patient level to monitor their health status by predicting vital signs, such as heart rate and breathing, and classifying their physical activities. It enables personalized patient monitoring with enhanced data privacy.

A heterogeneous stacked FL, FedStack, was proposed by Shaik et al. [4] to overcome the problems of the traditional FL approach, while enabling personalized monitoring of patients' physical actions. The authors achieved state-of-the-art performances using different deep learning models as part of local and global clients. The FedStack approach is confined to building the global model by stacking local clients' predictions heterogeneously and allowing local clients to have

* Corresponding author.
E-mail addresses: Thanveer.Shaik@usq.edu.au (T. Shaik), Xiaohui.Tao@usq.edu.au (X. Tao), cathylilin@whut.edu.cn (L. Li), Niall.Higgins@health.qld.gov.au (N. Higgins), Raj.Gururajan@usq.edu.au (R. Gururajan), Xujuan.Zhou@usq.edu.au (X. Zhou), Jianming.Yong@usq.edu.au (J. Yong).

https://doi.org/10.1016/j.patrec.2023.12.004
Received 16 March 2023; Received in revised form 21 October 2023; Accepted 11 December 2023
Available online 14 December 2023

different architectural models. However, it has a limitation of non-identically and independently distributed (non-IID) data, where the local clients' data distributions may be different. This can be addressed by allowing the global model to group the local clients based on their deep learning model output weights. To avoid any bias in grouping the local clients, unsupervised clustering methods can be adopted.

This study proposes a novel Clustered-FedStack framework to overcome FL's non-IID data challenge [5]. All models trained on local clients pass their predictions and output layer weights to the server, which builds a global server model based on the predictions received from the local models. Later, the global server model clusters local client models with output layer weights received and creates intermediate clustered models between local clients and the server. In this unsupervised process, the server model computes the cosine distance matrix among the local model output layer weights. To determine the number of clusters in this process, the BIC technique is adopted and maximum likelihood estimation is applied to the local model weights in the server. Three types of clustering techniques: centroid-based (k-Means), hierarchical (Agglomerative), and distribution-based (Gaussian Mixture Model) techniques are deployed. Cyclical learning rates are applied to estimate the convergence of the clustered models.

The proposed framework is evaluated with a human activity recognition (HAR) task using the publicly available sensor-based PPG-DALiA dataset [6]. The results show that clustered models have state-of-the-art performance in classifying human activities with the sensor data of 15 subjects. The performance of the clustered FedStack model is compared with four clustered FL baseline models, and the proposed model has outperformed the baseline models in all classification metrics. Moreover, the proposed framework can be scalable to Natural Language Processing (NLP) tasks. This has been evaluated on the drug review dataset [7], where the intermediate clustered models performed better and could handle a huge number of local clients with non-IID data to achieve superconvergence. Thus, the proposed clustered FedStack framework can group local clients and overcome the non-IID challenge in FL. The contributions of the present study include the following:

- A novel Clustered-FedStack framework is proposed to group local clients in an unsupervised approach and overcome the non-IID challenge in FL.
- Improved personalized modeling in FL by building intermediate clustered models between the global server model and local clients.
- Achievement of superconvergence of all clustered-FedStack models using Cyclical learning rates.
- A Clustered-FedStack approach that proves scalable for Natural Language Processing (NLP) tasks, effectively handling a high number of local clients with non-IID data.

Section 2 presents related works on FL and different aggregating techniques developed. Section 3 presents the formulation of the research problem and the proposed Clustered-FedStack framework. In Section 4, the proposed framework is evaluated in HAR and the results are discussed. The framework optimization with Cyclical learning rates is also presented in Section 4. In Section 5, we evaluate the scalability of the proposed framework using a NLP dataset. Section 6 concludes the paper.

2. Related works

Numerous studies have explored the aggregation of local model parameters in FL and passed them to the global model on the server. One of the first proposed aggregation techniques in FL is the Federated Averaging (FedAvg) algorithm, which uses the average function to aggregate local model weights and generate new weights to feed to the global model [8]. However, the FedAvg technique cannot optimize models if a client has a heterogeneous data distribution. To combat this, Arivazhagan et al. [9] proposed FedPer, which has two layers: a base layer and a personalization layer. FedAvg trains the base layers, while the personalization layers are trained with stochastic gradient descent, helping to mitigate the ill effects of statistical heterogeneity. Wang et al. [10] proposed Federated Matched Averaging (FedMA), which is a layer-wise approach that matches and merges nodes with the same weights, trains them independently, and communicates the layers to the global model.

Osmani et al. [11] proposed a multi-level FL system for HAR, which includes a reconciliation step based on FL aggregation techniques such as FedAvg or Federated Normalized Averaging. Xiao et al. [12] proposed another FL system for HAR with enhanced feature extractions. They designed a Perceptive Extraction Network (PEN) with two networks: a featured network based on the convolutional block to extract local features, and a relation network based on Long Short-Term Memory (LSTM) and an attention mechanism to mine global relationships. Pang et al. [13] proposed a rule-based collaborative framework (CloREF) that allows local clients to use heterogeneous local models. Tian et al. [14] discussed the limitations of traditional FL methods in heterogeneous IoT systems and proposed a novel Weight Similarity-based Client Clustering (WSCC) approach to address the non-IID challenge in FL. The WSCC approach involves splitting clients into different groups based on their data set distributions using an affinity-propagation-based method. Their proposed approach outperformed existing FL schemes under different non-IID settings, achieving up to 20% improvements in accuracy without requiring extra data transmissions or additional models.

Federated Learning in HAR The increasing use of electronic assistive health applications such as smartwatches and activity trackers has led to the emergence of pervasive or ubiquitous computing, where devices can seamlessly exchange data with each other [15]. Although this has the advantage of real-time tracking of human health changes, it is vulnerable to security breaches that compromise data privacy [16]. The advancement of AI techniques as a whole is contributing to the massive amount of human data being generated worldwide every second. To handle such enormous data, Google introduced FL, which trains a ML algorithm across decentralized devices or servers without exchanging their local data samples. FL overcomes the data privacy issues associated with traditional centralized learning techniques, where all device and/or server data is merged for analysis [1]. Sannara et al. [17] evaluated the performance of FL aggregation techniques like Federated Averaging (FedAvg), Federated Learning with Matched Averaging (FedMa), and Federated Personalization Layer (FedPer) against centralized training techniques. They used the CNN model to classify eight physical activities. Zhao et al. [18] designed an activity recognition system based on semi-supervised FL. Ouyang et al. [19] proposed the ClusterFL approach, which exploits the similarity of users' data to minimize the empirical loss of trained models. This improved Federated model accuracy and communication efficiency between local models and global models.

Local clients may have different data distributions, demographics, and model architectures. Passing all the local clients' parameters to build a robust global server model poses challenges such as label imbalance and non-IID. To identify hidden patterns or relationships among the local clients and overcome these challenges, unsupervised clustering techniques can be adopted to improve personalized learning in FL. This study proposes a clustered FL framework to overcome these identified challenges.

3. Methodology

To accommodate heterogeneous architectural models for local clients, we adopt the previously published FedStack framework by Shaik et al. [4]. This study extends the FedStack framework to the clustered-FedStack framework, facilitating the creation of heterogeneous multi-global FL models by clustering individual subjects with local models.

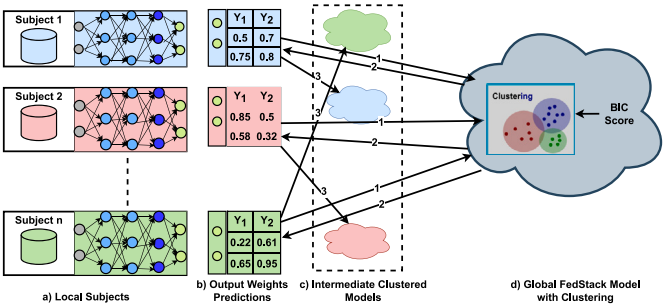

Fig. 1. Clustered FedStack model.

3.1. Research problem

In this study, the research problem is to overcome the non-IID data challenge in a FL environment. Let $S = \{s_1, s_2, \ldots, s_N\}$ be the set of subjects, where the data is non-IID. The objective is to divide subjects S into M clusters $C = \{c_1, c_2, \ldots, c_M\}$, where each cluster c_m is a subset of subjects S, $c_m \subseteq S$. For each cluster c_m, there exists a local model l_m that can be heterogeneous according to the subject's convenience. The predictions p_m from local models and their corresponding output layer weights are passed to a global model server g. The training process for the global model g is shown in Eq. (1).

$$\text{train}(g) \leftarrow \sum_{m=1}^{M} c_m \leftarrow \sum_{m=1}^{M} l_m(p_m) \tag{1}$$

where: train(g) refers to the training process for the Global Model g using local model predictions of cluster c_m, and $l_m(p_m)$ represents the local model l_m and its predictions p_m for each subject in the cluster c_m.

3.2. Clustered-FedStack framework

In the Clustered-FedStack framework, local clients train their models on private data and then forward their model predictions p and output layer weights Q to the global server model g for training, as shown in Fig. 1. The figure's arrow numbers indicate the framework execution order. After receiving the local model predictions and output layer weights, the global server model determines the number of clusters using the BIC score. It then clusters the local clients based on their output layer weights. For each label i in local model training, an output neuron without a successor is configured to gather the computed and accumulated values from the local model's input and hidden layers. The output neuron value q_i is calculated using Eq. (2), with inputs x_i, weights w_i, and bias b for a local model l_n. By computing all output neuron values, the local model l_n predictions p can be estimated using Eq. (3).

$$q_i = l_n(b, x_i, w_i) \tag{2}$$

$$p = l_n(b + \sum_{i=1}^{n} x_i \cdot w_i) \tag{3}$$

Output neuron values for each local model l_n are consolidated into a single set Q using Eq. (4). This procedure is repeated for all local models based on their output layer values, forming a large set Q as defined in Eq. (5).

$$Q = \{q_1, q_2, q_3, \ldots, q_n\} \tag{4}$$

$$Q = \{Q_{l_1}, Q_{l_2}, \ldots, Q_{l_n}\} \tag{5}$$

3.2.1. Clustering technique

Given the set Q from Eq. (5), where each element of the set represents the values of a local model's l_n output layer, the goal is to divide Q into k clusters, where $k \leq n$, represented by $C = \{c_1, c_2, \ldots, c_k\}$. There are various techniques that can be applied to clustering, including centroid-based, hierarchical, and distribution-based methods. The general objective of these methods is to minimize the within-cluster sum of squared differences or a related measure of dissimilarity, as described in Eq. (6). The notation $\arg\min_C$ refers to finding the set of clusters C that minimizes the following expression, where the "arg min" stands for the argument of the minimum, i.e., the specific value of the variable that results in the lowest possible value of the given function.

$$\arg\min_{C} \left(\sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2 \right) \tag{6}$$

Here, C represents the set of clusters, x is a data point, and c_i is the representative point, such as a centroid. The term $\|\cdot\|$ represents a distance measure.

Cosine similarity is utilized to assign each local model's output neuron set to a specific cluster, considering the angle between output neuron sets of two local model l_n as Q_{l_1} and Q_{l_2}, the cosine similarity can be estimated using Eq. (7).

$$S_C(Q_{l_1}, Q_{l_2}) = \frac{Q_{l_1} \cdot Q_{l_2}}{\|Q_{l_1}\| \, \|Q_{l_2}\|} \tag{7}$$

3.2.2. Bayesian information criterion

The proposed Clustered-FedStack technique enables the global server model to access local models' predictions and layer weights. However, using an unsupervised method to determine the number of clusters in local models is challenging. The BIC technique is utilized to overcome this. BIC calculates its value based on a clustering model $\mathcal{M}$'s maximum likelihood function M_L, representing the probability that the layer weights data fits the clustering model [20]. This is shown in Eq. (8). BIC values balance the maximum likelihood estimation against the number of model parameters m_p, seeking a model with the fewest parameters that can accurately explain the data clusters, as in Eq. (9).

$$M_L(\mathcal{M}) = -2\ln(\mathcal{L}) + m_p \ln(n) \tag{8}$$

$$BIC = -2\ln(\mathcal{L}) + m_p \ln(n) = M_L(\mathcal{M}) \tag{9}$$

The BIC values for each clustering model are compared with the minimum BIC value indicating the optimal clustering model. This process ensures that the global model converges by configuring a suitable number of clusters for local models, resulting in a consolidated global model that represents heterogeneous subject models.

3.3. Clustered-FedStack algorithm

Algorithm 1 presents the proposed Clustered-FedStack process in detail. Line 1 initializes empty sets to collect output layer weights and clustered models, and datasets D and D' for evaluating the global server model. Lines 2–7 detail the FedStack process, where local client model predictions and weights are passed to the global server model g for training and testing. Lines 8–10 present the iteration through all local model weights in g to collect their output layer weights. Lines 11–12 detail the determination of the number of clusters to be formed from the weights W set. Line 13 computes the cosine distance among all the local model weights collected. Lines 14–19 explain the clustering process for all the local models, based on Lines 11–13.

Algorithm 1 Proposed Clustered-FedStack Algorithm

Require:
Subjects set $S = \{s_1, s_2, \ldots, s_n\}$
Local AI models $M = \{m_1, m_2, \ldots, m_m\}$
Labels set $K = \{1, 2, \ldots, k\}$
Global Server Model g
Ensure: Classification probabilities of labels K for each intermediate cluster model C
Initialization:
D: Dataset for training
D': Unseen Dataset for testing
$W = \emptyset$: Set to collect weights
$CM = \emptyset$: Set for clustered models
$stack = \left\{ \{m_i^K, m_i^w\}, \{m_j^K, m_j^w\}, \{m_k^K, m_k^w\} \right\}$; ▷ Predictions and weights of local AI models
for $m \in M$ **do**
  $g^{train} \leftarrow stack$;
  $g^{test} \leftarrow D'$;
end for
for $m \in G(M)$ **do**
  Collect weights of m: $W \leftarrow \{m, w\}$;
end for
Determine Clusters:
Compute BIC scores of $CM \geq M$;
$CM \leftarrow \min(\text{BIC})$;
Compute cosine distance among $\{\{m_1, w_1\}, \{m_2, w_2\}, \ldots, \{m_m, w_m\}\}$;
Assignment:
for c in C **do**
  $c \leftarrow \arg\min_C \left(\sum_{i=1}^{k} \sum_{x \in C_i} ||x - c_i||^2 \right)$;
  $CM \leftarrow c$;
end for
Return CM;

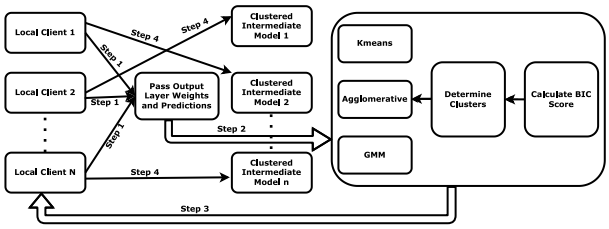

Fig. 2. Experimental design of the proposed framework.

4. Experiments on clustered FedStack in HAR

Conventional FL methods assume that the data distribution is consistent among all clients [21]. However, this assumption may not be valid in FL, as data heterogeneity can be present [22]. This limitation forces clients to have identical data distribution and architectural models to build global models. FedStack [4] addressed the issue of identical architectural models in FL. The goal of this study is to extend the FedStack framework by introducing intermediate clustered models to address the non-IID challenge in FL.

In this study, the objective is to overcome the non-IID challenge in FL. To achieve this, the proposed Clustered-FedStack algorithm is applied to the domain of human activity recognition, where patients' physical activity is classified. The non-IID data distribution of the dataset used in the experiment is presented. The proposed methodology involves passing the output layer weights and predictions of local clients to the global model, which then calculates unsupervised clustering of the local model layer weights to group the local clients and establish clustered intermediate models. The experimental design is presented in Fig. 2. The evaluation results compare the performance of the proposed framework to the baseline models and show clustering results leading to clustered FedStack models. Furthermore, the convergence of the clustered FedStack models is analyzed using Cyclical learning rates.

4.1. Dataset

The proposed Clustered-FedStack algorithm was evaluated on the HAR problem, which involves classifying patients' physical activity.

Table 1
Non-IID data.

Local clients	Distribution	1	2	3	4	5	6	7	8
Subject 1	27 724	2800	1148	1380	1648	3556	9420	3016	4756
Subject 2	22 712	2400	1068	1216	1548	3680	4880	2756	5164
Subject 3	26 900	2400	1740	1172	1516	3640	8640	2952	4840
Subject 4	26 528	2280	2092	1312	1900	4028	7580	2376	4960
Subject 5	26 924	2400	1860	1160	1728	3320	9020	2356	5080
Subject 6	11 812	2532	1720	1236	2132	4192	9020	0	0
Subject 7	28 580	2472	1624	1096	2012	4140	9700	2836	4700
Subject 8	23 992	2400	1648	1292	1680	3080	7200	1924	4768
Subject 9	26 212	2400	1932	1140	2216	3820	7368	2356	4980
Subject 10	28 424	2392	1868	1220	1952	3748	8336	4328	4580
Subject 11	28 052	2400	1828	1296	1960	3440	9632	2616	4880
Subject 12	23 680	2408	1936	1120	1920	3560	5840	2116	4780
Subject 13	26 996	2420	1988	1160	1992	3588	8112	2836	4900
Subject 14	25 584	2432	1824	1300	2008	3816	6924	2460	4820
Subject 15	23 504	2444	1676	1416	1620	3140	5760	2636	4812

The PPG-DALiA [6] dataset, which is publicly accessible and cited in [6], was utilized for this study. This dataset includes physiological and motion data gathered from 15 participants as they engaged in a diverse array of activities, closely mirroring real-life conditions. The data was collected from both a wrist-worn (Empatica E4) and a chest-worn (RespiBAN) device, and includes 11 attributes such as 3-Dimensional (3D) acceleration data, Electrocardiogram (ECG), respiration, Blood Volume Pulse (BVP), Electrothermal Activity (EDA), and body temperature. The 3D acceleration data was labeled with eight different physical activities.

4.2. Non-IID data

Table 1 shows the distribution and activity of local clients in a FL scenario with non-IID data. Each row represents a client, and each column represents a feature. The "Distribution" column shows the number of data points available at each client, which varies across clients, indicating non-IID in the dataset. The remaining columns represent different activities that are each related to the type of data collected or the task being performed. For instance, "Activity 1" to "Activity 8" could be different types of sensor readings or behavioral data collected from different sources. The non-IID nature of this data could potentially impact the performance of the FL algorithm since the data distribution across clients is not uniform, and the model may not generalize well to all clients. Therefore, special attention must be given to handling the non-IID data in FL, by using the technique of personalized FL to improve model performance for each client's unique data distribution.

4.3. Data modeling

In data modeling, three AI models were chosen: Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Bidirectional Long Short-Term Memory (BiLSTM) models, due to their state-of-the-art performances in FL works [12] and activity classification [23]. Each subject trained with one of the chosen models locally and passed their predictions and local model output layer weights to the global server model. The proposed framework clustered the global model without any private information about local clients, based on the output layer weights.

4.4. Baseline models

- **ClusterFL** [19]**:** A clustering-based FL system for the HAR application. The ClusterFL approach captures the intrinsic clustering relation among local clients and minimizes the training loss.
- **FL+HC** [24]**:** A hierarchical clustered FL system to separate clusters of clients based on the similarity of their local updates to the global server model.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
S1	0	0.18	0.21	0.29	0.28	0.35	0.21	0.39	0.2	0.2	0.38	0.52	0.15	0.27	0.14
S2	0.18	0	0.02	0.06	0.1	0.11	0.26	0.11	0.06	0.19	0.21	0.25	0.12	0.06	0.05
S3	0.21	0.02	0	0.06	0.09	0.19	0.28	0.13	0.04	0.24	0.11	0.29	0.08	0.03	0.04
S4	0.29	0.06	0.06	0	0.06	0.15	0.23	0.12	0.06	0.18	0.2	0.26	0.11	0.02	0.13
S5	0.28	0.1	0.09	0.06	0	0.16	0.11	0.15	0.06	0.28	0.16	0.23	0.08	0.05	0.15
S6	0.35	0.11	0.19	0.15	0.16	0	0.33	0.21	0.23	0.37	0.43	0.26	0.31	0.17	0.2
S7	0.21	0.26	0.28	0.23	0.11	0.33	0	0.28	0.18	0.25	0.34	0.33	0.15	0.24	0.3
S8	0.39	0.11	0.13	0.12	0.15	0.21	0.28	0	0.09	0.2	0.26	0.07	0.25	0.15	0.22
S9	0.2	0.06	0.04	0.06	0.06	0.23	0.18	0.09	0	0.16	0.14	0.2	0.06	0.05	0.07
S10	0.2	0.19	0.24	0.18	0.28	0.37	0.25	0.2	0.16	0	0.45	0.32	0.24	0.25	0.25
S11	0.38	0.21	0.11	0.2	0.16	0.43	0.34	0.26	0.14	0.45	0	0.47	0.15	0.11	0.21
S12	0.52	0.25	0.29	0.26	0.23	0.26	0.33	0.07	0.2	0.32	0.47	0	0.39	0.32	0.34
S13	0.15	0.12	0.08	0.11	0.08	0.31	0.15	0.25	0.06	0.24	0.15	0.39	0	0.08	0.08
S14	0.27	0.06	0.03	0.02	0.05	0.17	0.24	0.15	0.05	0.25	0.11	0.32	0.08	0	0.09
S15	0.14	0.05	0.04	0.13	0.15	0.2	0.3	0.22	0.07	0.25	0.21	0.34	0.08	0.09	0

Fig. 3. Cosine distance among local clients.

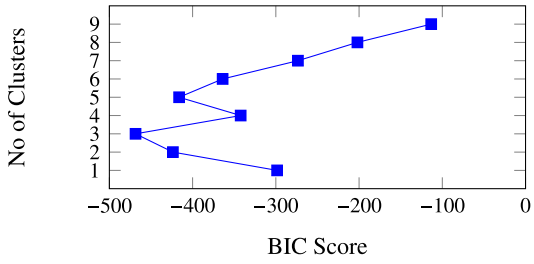

Fig. 4. BIC score to determine the number of clusters.

- **HypCluster** [25]**:** A hypothesis-based clustering with a stochastic Expectation-Maximization (EM) algorithm adopted for the FL approach, where local clients partition into a certain number of clusters and then the model finds the best hypothesis for each cluster.
- **Dynamic Clustering** [26]**:** A three-phased data clustering algorithm, namely, generative adversarial network-based clustering, cluster calibration, and cluster division, designed to overcome the fixed shape of clusters, data privacy breaches, and non-adaptive numbers of clusters.

4.5. Results analysis

4.5.1. Clustering results

Before clustering, the cosine distance among all 15 local models trained on clients is calculated to check their similarity in terms of the models' output layer weights, as shown in Fig. 3. The matrix heatmap ranges on a scale from 0 to 0.6 where 0 shows no cosine distance between the client output layer values, and 0.6 shows the maximum cosine distance.

The proposed Clustered-FedStack algorithm employed the BIC approach to calculate the maximum likelihood function on the output layer weights received from the local client models by the global server model, as shown in Fig. 1. This process determines the number of clusters among the 15 local clients. Fig. 4 shows that the lowest BIC score corresponds to three clusters in the global server model. After determining the clusters, three clustering techniques were applied: centroid-based clustering (K-Means) [27], hierarchical clustering (Agglomerative) [28], and distribution-based clustering (Gaussian Mixture Model (GMM)) [29]. Fig. 5 shows that K-Means and Agglomerative clustering produced similar groups of local client models, while GMM clustering grouped all CNN models into the second cluster and distributed other ANN and BiLSTM models in the first and third clusters.

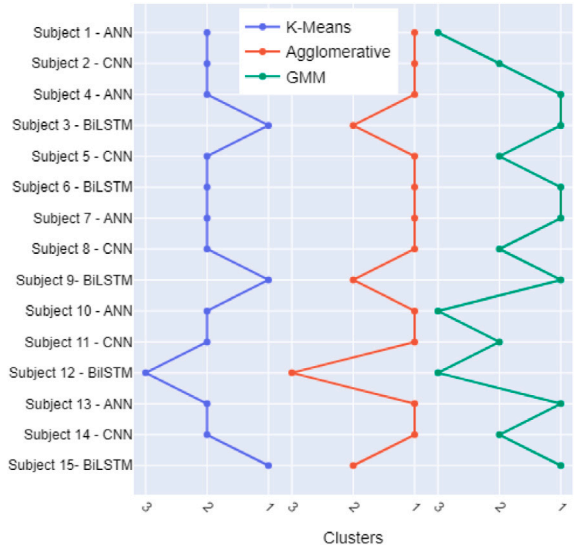

Fig. 5. Clustering results.

4.5.2. Clustered FedStack model performances

After determining the clusters, each cluster of local clients passes their output layer weights to an intermediate Clustered-FedStack model, situated between the local clients and the global server model, as shown in Fig. 1. This approach reduces the load on the global server model and groups similar local models for more efficient AI results. The three clustering techniques generate three Clustered-FedStack models each, and their performance in HAR is shown in Table 2. All nine Clustered-FedStack intermediate global models generated from the clustering techniques have performed well in the HAR task. K-Means and agglomerative clustering, having similar clustering results, showed similar classification accuracy in HAR. While comparing the results, the GMM Clustered FedStack models, which are distribution-based, exhibited slightly better accuracy than the other two clustered models.

4.5.3. Baseline models comparison

The proposed framework was compared against four other baseline models in FL approaches with clustering. All models were trained using 3D acceleration data for HAR tasks, and their evaluation results are presented in Table 3. As K-Means and hierarchical clustering techniques generate similar clusters from the 15 local client models, the table shows three clustered models (Clustered FedStack 1, Clustered FedStack 2, Clustered FedStack 3) built based on K-Means and hierarchical clustering, and three clustered models (Clustered FedStack 7, Clustered FedStack 8, Clustered FedStack 9) built based on the GMM model. The Table presents the mean of four metrics: balanced accuracy, precision, recall, and F1-score in classifying eight activities for six intermediate clustered models. The proposed approach outperformed all other baseline models in terms of all the metrics.

4.6. Convergence analysis

The optimization of the proposed Clustered FedStack framework is estimated using Cyclical learning rates [30] for convergence. The performance of the intermediate Clustered FedStack models shown in Fig. 1 is optimized using the Learning Rate (α) of the deep learning models. In the Cyclical learning rates process, the α values are cycled with an initial learning rate of 0.00001 and a maximum learning rate of 0.001, and stochastic gradient descent is performed. A scale function is defined to control the change from the initial learning rate to the maximal learning rate and back to the initial learning rate. The

Table 2
Clustered FedStack model accuracy in HAR.

Activity	K-Means clusters			Agglomerative clusters			GMM clusters		
	Cluster 1	Cluster 2	Cluster 3	Cluster 1	Cluster 2	Cluster 3	Cluster 1	Cluster 2	Cluster 3
Sitting	0.99	0.96	0.95	0.97	0.99	0.95	0.99	0.99	0.99
Ascending and descending stairs	0.92	0.96	0.94	0.96	0.92	0.94	0.92	0.92	0.92
Table soccer	0.96	0.95	0.95	0.97	0.95	0.95	0.95	0.96	0.97
Cycling	0.94	0.97	0.98	0.95	0.96	0.98	0.96	0.93	0.96
Driving a car	0.89	0.93	0.99	0.89	0.95	0.99	0.95	0.93	0.97
Lunch break	0.87	0.86	0.92	0.87	0.89	0.92	0.9	0.9	0.91
Walking	0.91	0.90	0.89	0.90	0.92	0.89	0.91	0.91	0.92
Working	0.92	0.96	0.95	0.97	0.97	0.95	0.96	0.92	0.97

Table 3
Baseline models comparison.

Model	Balanced accuracy	Precision	Recall	F1-Score
ClusterFL [19]	0.93	0.78	0.86	0.82
FL+HC [24]	0.94	0.85	0.89	0.83
HypCluster [25]	0.9	0.65	0.56	0.65
Dynamic clustering [26]	0.92	0.86	0.75	0.76
Clustered FedStack 1	**0.98**	**0.95**	**0.91**	**0.93**
Clustered FedStack 2	**0.96**	**0.89**	**0.9**	**0.89**
Clustered FedStack 3	**0.94**	**0.91**	**0.92**	**0.91**
Clustered FedStack 7	**0.95**	**0.92**	**0.91**	**0.91**
Clustered FedStack 8	**0.98**	**0.94**	**0.93**	**0.93**
Clustered FedStack 9	**0.97**	**0.96**	**0.95**	**0.95**

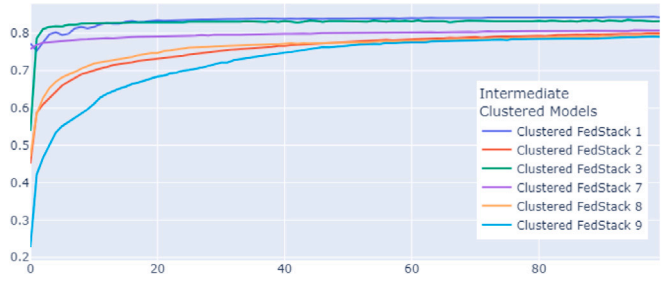

Fig. 6. Convergence of intermediate Clustered FedStack models on PPG-DALiA under the Cyclical learning rates.

scale function, a lambda function shown in Eq. (10), scales the initial amplitude by half with each cycle.

$$lambda\ \ x : \frac{1}{(2^{(x-1)})} \tag{10}$$

Fig. 6 presents the convergence curves of six intermediate Clustered FedStack models from the three clustering techniques proposed in this study. The intermediate clustered models built based on K-Means and Agglomerative clustering converge faster than the clustered models built based on GMM clustering. There is not much difference in the number of epochs required for each clustered model to converge. All six models converge in less than 50 epochs. The results show that the proposed Clustered FedStack framework can be implemented with centroid-based, hierarchical or distribution-based clustering. The Clustered FedStack models built based on any of these clustering techniques converge quickly in 50 epochs.

5. Experiments on clustered FedStack scalability in NLP tasks

The scalability of the proposed Clustered FedStack model was rigorously assessed through a targeted evaluation. For this purpose, the drug review dataset [7], containing reviews and ratings, was utilized. This comprehensive dataset encompasses 3677 distinct drugs and 916 different medical conditions. The aim of this experiment is to classify drug ratings (1–10) based on input data such as medical conditions. In alignment with the clustering methodologies proposed in the Clustered FedStack framework, the GMM clustering was employed to perform

Table 4
Clustered FedStack performance in classification of drug ratings.

Model	Accuracy	Precision	Recall	F1-Score
ClusterFL	0.92	0.8	0.88	0.82
FL+HC	0.93	0.87	0.91	0.83
HypCluster	0.89	0.66	0.57	0.65
Dynamic clustering	0.91	0.88	0.77	0.76
Clustered FedStack 1	0.99	0.92	0.93	0.91
Clustered FedStack 2	0.98	0.91	0.92	0.91
Clustered FedStack 3	1	0.96	0.95	0.95
Clustered FedStack 4	0.97	0.94	0.93	0.93
Clustered FedStack 5	0.98	0.97	0.94	0.96
Clustered FedStack 6	1	0.97	0.93	0.95
Clustered FedStack 7	0.99	0.98	0.97	0.97
Clustered FedStack 8	0.98	0.97	0.94	0.96
Clustered FedStack 9	0.96	0.93	0.94	0.93
Clustered FedStack 10	0.94	0.93	0.92	0.91

the clustering of 2191 drugs, resulting in 78 unique clusters as shown in Supplementary Material. The Supplementary Material also includes information on the cosine distance for 200 local clients (drugs).

The performance comparisons of different clustering models, including the top 10 variations of the Clustered FedStack model, are presented in Table 4. The metrics evaluated include accuracy, precision, recall, and F1-score for classifying drug ratings. Four baseline models are included: ClusterFL, FL+HC, HypCluster, and Dynamic Clustering. Their performances are relatively consistent, with accuracy ranging from 0.89 to 0.93. The Clustered FedStack models demonstrated superior performance, with notable improvements in all evaluated metrics. The accuracy for these variations ranged from 0.94 to a perfect 1, highlighting the efficiency and robustness of the model. The first five Clustered FedStack models exhibited particularly impressive results, achieving almost perfect or perfect accuracy. The precision, recall, and F1-score also showcased strong consistency and harmony, reflecting the model's ability to balance both false positives and false negatives.

These results underscore the scalability and effectiveness of the Clustered FedStack model across local clients with non-IID data. The model's scalability and adaptability are evident, maintaining high levels of accuracy and F1-scores regardless of the local clients' variation. This highlights the Clustered FedStack model's potential in managing large and intricate datasets like drug reviews and ratings, validating both its resilience and relevance to real-world applications.

The convergence of the proposed Clustered FedStack on the drug review dataset has been assessed, as shown in Fig. 7. The line chart presents a convergence pattern that denotes accuracy in the y-axis across 100 epochs in the x-axis. The values for Clustered FedStack 1 exhibited consistent growth, starting at 0.7882 and reaching 0.8556 by epoch 40. Similarly, other clustered FedStacks demonstrated a progressive increase in values across epochs, such as Clustered FedStack 2, which advanced from 0.5188 to 0.8811, signifying a gradual strengthening of the model. These convergence trends shed light on the efficiency and efficacy of the iterative learning process. Variations in convergence rates among different stacks were observed, reflecting the distinct characteristics of each clustered FedStack. These findings suggest a general trend of convergence towards higher values,

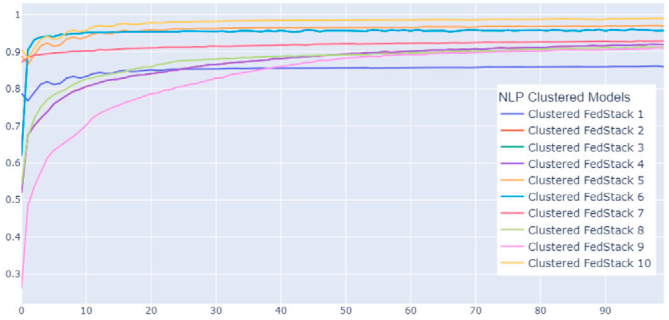

Fig. 7. Convergence of intermediate Clustered FedStack models on drug review dataset under Cyclical learning rates.

though occasional oscillations and fluctuations were detected in specific iterations. This in-depth analysis offers valuable insights into the behavior of clustered FL systems, potentially opening new avenues for enhanced optimization strategies and a more profound understanding of convergence mechanisms within distributed ML frameworks.

6. Conclusion

In the present study, a novel framework named Clustered-FedStack was introduced, designed to cluster local clients within the FL paradigm based on the weights of their output layers. This methodology was devised to address the non-IID challenge inherent to FL. It is important to acknowledge certain limitations of the proposed framework, notably its incompatibility with the application on local clients utilizing conventional Machine Learning models for the training of private data. Moreover, the global server model's process of clustering local clients operates on an unsupervised basis, without access to specific information about local clients, depending solely on the local model rather than client demographics. In light of these considerations, future investigations should aim to develop strategies for the dynamic clustering of local clients, taking into account meta-information that pertains to client similarities.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Here is the link to publicly available dataset: https://archive.ics.uci.edu/ml/datasets/PPG-DaLiA.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.patrec.2023.12.004.

References

[1] K. Bonawitz, P. Kairouz, B. McMahan, D. Ramage, Federated learning and privacy: Building privacy-preserving systems for machine learning and data science on decentralized data, Queue 19 (5) (2021) 87–114.
[2] Y. Shi, Y. Zhang, P. Zhang, Y. Xiao, L. Niu, Federated learning with l1 regularization, Pattern Recognit. Lett. (2023).
[3] L. Peng, G. Luo, A. Walker, Z. Zaiman, E.K. Jones, H. Gupta, K. Kersten, J.L. Burns, C.A. Harle, T. Magoc, et al., Evaluation of federated learning variations for COVID-19 diagnosis using chest radiographs from 42 US and European hospitals, J. Amer. Med. Inform. Assoc. (2022).
[4] T. Shaik, X. Tao, N. Higgins, R. Gururajan, Y. Li, X. Zhou, U.R. Acharya, FedStack: Personalized activity monitoring using stacked federated learning, Knowl.-Based Syst. 257 (2022) 109929.
[5] M. Arafeh, H. Ould-Slimane, H. Otrok, A. Mourad, C. Talhi, E. Damiani, Data independent warmup scheme for non-IID federated learning, Inform. Sci. 623 (2023) 342–360.
[6] A. Reiss, I. Indlekofer, P. Schmidt, K. Van Laerhoven, Deep PPG: large-scale heart rate estimation with convolutional neural networks, Sensors 19 (14) (2019) 3079.
[7] S. Kallumadi, F. Grer, Drug Review Dataset (Drugs.com), UCI Machine Learning Repository, 2018, http://dx.doi.org/10.24432/C5SK5S.
[8] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.
[9] M.G. Arivazhagan, V. Aggarwal, A.K. Singh, S. Choudhary, Federated learning with personalization layers, 2019, arXiv preprint arXiv:1912.00818.
[10] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, Y. Khazaeni, Federated learning with matched averaging, in: International Conference on Learning Representations, 2020.
[11] A. Osmani, M. Hamidi, Reduction of the position bias via multi-level learning for activity recognition, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2022, pp. 289–302.
[12] Z. Xiao, X. Xu, H. Xing, F. Song, X. Wang, B. Zhao, A federated learning system with enhanced feature extraction for human activity recognition, Knowl.-Based Syst. 229 (2021) 107338.
[13] Y. Pang, H. Zhang, J.D. Deng, L. Peng, F. Teng, Rule-based collaborative learning with heterogeneous local learning models, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2022, pp. 639–651.
[14] P. Tian, W. Liao, W. Yu, E. Blasch, WSCC: A weight-similarity-based client clustering approach for Non-IID federated learning, IEEE Internet Things J. 9 (20) (2022) 20243–20256.
[15] A. Alam, S. Qazi, N. Iqbal, K. Raza, Fog, edge and pervasive computing in intelligent internet of things driven applications in healthcare: Challenges, limitations and future use, in: Fog, Edge, and Pervasive Computing in Intelligent IoT Driven Applications, Wiley Online Library, 2020, pp. 1–26.
[16] L.M. Dang, M. Piran, D. Han, K. Min, H. Moon, et al., A survey on internet of things and cloud computing for healthcare, Electronics 8 (7) (2019) 768.
[17] S. Ek, F. Portet, P. Lalanda, G. Vega, Evaluation of federated learning aggregation algorithms: application to human activity recognition, in: Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers, 2020, pp. 638–643.
[18] Y. Zhao, H. Liu, H. Li, P. Barnaghi, Semi-supervised federated learning for activity recognition, ACM Trans. Intell. Syst. Technol. 1 (1) (2021).
[19] X. Ouyang, Z. Xie, J. Zhou, J. Huang, G. Xing, Clusterfl: a similarity-aware federated learning system for human activity recognition, in: Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services, 2021, pp. 54–66.
[20] C. Xiang, P.C. Yong, L.S. Meng, Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees, Pattern Recognit. Lett. 29 (7) (2008) 918–924.
[21] L. Yang, J. Huang, W. Lin, J. Cao, Personalized federated learning on non-IID data via group-based meta-learning, ACM Trans. Knowl. Discov. Data (TKDD) (2022).
[22] X. Shang, Y. Lu, Y.-m. Cheung, H. Wang, FEDIC: Federated learning on non-IID and long-tailed data via calibrated distillation, in: 2022 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2022, pp. 1–6.
[23] J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, Deep learning for sensor-based activity recognition: A survey, Pattern Recognit. Lett. 119 (2019) 3–11.
[24] C. Briggs, Z. Fan, P. Andras, Federated learning with hierarchical clustering of local updates to improve training on non-IID data, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–9.
[25] Y. Mansour, M. Mohri, J. Ro, A.T. Suresh, Three approaches for personalization with applications to federated learning, 2020, arXiv preprint arXiv:2002.10619.
[26] Y. Kim, E. Al Hakim, J. Haraldson, H. Eriksson, J.M.B. da Silva, C. Fischione, Dynamic clustering in federated learning, in: ICC 2021-IEEE International Conference on Communications, IEEE, 2021, pp. 1–6.
[27] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.
[28] A. Sellami, A.B. Abbes, V. Barra, I.R. Farah, Fused 3-D spectral-spatial deep neural networks and spectral clustering for hyperspectral image classification, Pattern Recognit. Lett. 138 (2020) 594–600.
[29] J. Lücke, D. Forster, k-means as a variational EM approximation of Gaussian mixture models, Pattern Recognit. Lett. 125 (2019) 349–356.
[30] L.N. Smith, Cyclical learning rates for training neural networks, in: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2017, pp. 464–472.