2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops) | 978-1-6654-0424-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/PERCOMWORKSHOPS51409.2021.9431011

# MLMG: Multi-Local and Multi-Global Model Aggregation for Federated Learning

Yang Qin
*The University of Tokyo*
Tokyo, Japan
shin@hal.ipc.i.u-tokyo.ac.jp

Masaaki Kondo
*The University of Tokyo*
Tokyo, Japan
kondo@hal.ipc.i.u-tokyo.ac.jp

***Abstract*—Federated learning has attracted much interest and attention as a solution to collaboratively learn a prediction model without sharing the training data of users. Existing federated learning approaches usually develop a single independent local model for each client to train their privacy-sensitive data, afterward adopt a single centralized global model to exchange the trained parameters of clients that participate in federated training. However, given the diverse characteristics of local data and the heterogeneity across participating clients, the conventional federated learning paradigm may not achieve uniformly good performance over all users.**

**In this work, we propose a novel federated learning mechanism which suggests using a Multi-Local and Multi-Global (MLMG) model aggregation to train the non-IID user data with clustering methods. Then a Matching algorithm is introduced to derive the appropriate exchanges between local models and global models. The new federated learning mechanism helps separate the data and user with different characteristics, thus makes it easier to capture the heterogeneity of data distributions across the users. We choose the latest on-device neural network for anomaly detection to evaluate the proposal, and experimental results based on several benchmark datasets demonstrate better detection accuracy (up to 2.83% accuracy improvement) of the novel paradigm compared with a conventional federated learning approach.**

***Index Terms*—federated learning, multi-local and multi-global, aggregation mechanism, on-device neural network, anomaly detection**

## I. Introduction

The primary goal of machine learning is to develop a general model that achieves uniform good performance for all the users. However, many machine learning algorithms suffer from insufficient training data, as the data dispersed over different parties are often under the protection of personal information in reality. The federated learning (FL), first proposed by Google in [1], is a new decentralized machine learning framework that learns a shared model by aggregating the locally-computed model parameters from remote clients. In the last few years, the federated learning framework has received much attention for its capability to develop a prediction model collaboratively while protecting the privacy-sensitive data among users [2], [3].

The vanilla federated learning framework considers a very practical scenario, where the distributed data among clients are non-I.I.D (identically and independently distributed). The ultimate goal of federated learning is to train a global model that achieves uniformly good performance over all participating clients. Instinctively, if the distribution of user data does not deviate too much from each other, collaboratively training a prediction model is supposed to be beneficial to reduce the generalization error of local clients. By contrast, if the distribution of data varies significantly among participating clients, it becomes difficult to calculate a global model with high generalization ability. Moreover, the difficulty is exacerbated even further as the diversity among local data increases. The generalization error of local models motivates many researchers to explore the data heterogeneity across participating clients, aiming at deploying more specialized models on local data. Recently, several new federated learning frameworks have been investigated with the goal of grouping clients trained on the same or similar datasets into the same cluster [4]–[6].

However, existing federated learning approaches process the non-I.I.D data by clustering the clients, but keep the assumption of a single prediction model for each user as the vanilla federated learning. There arises a problem that sufficient accuracy cannot be obtained through one prediction model because of the high complexity of local data, take anomaly detection for an example, the normal patterns are not always simple and may vary depending on a given environment. Suppose we train a single abnormal detection model for each user on their local data, the detection accuracy may decrease when there is a large diversity in normal data (i.e., number of patterns, variance, etc.). In [7], the authors suggested a multi-instance design in order to clearly distinguish anomaly patterns from multiple normal patterns. More concretely, a detection model is specialized for each multiple normal pattern, so that the abnormal behaviors can be detected more accurately through the comparison between multiple detection models. This observation illustrates that we should not only specialize models for non-I.I.D clients, but also customize models for the diversity of training data.

To address the issues above, we study a novel federated learning framework, which suggests a multi-local and multi-global (MLMG) model aggregation to train non-I.I.D data. Specifically, given the data complexity in a client's dataset, e.g., multiple normal patterns, we suggest clustering the data into several groups and train a detection model for each group. Thus, the data with the same/similar characteristics are

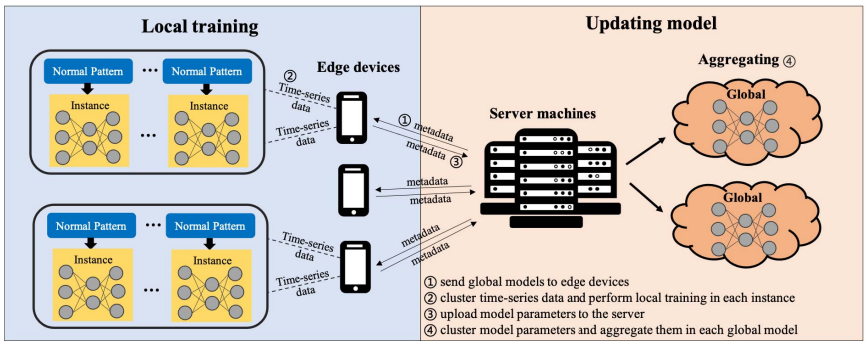



Fig. 1. Framework of a multi-local and multi-global federated learning system for anomaly detection.

trained to learn a prediction model, to reduce generalization errors caused by data diversity. We refer to this as *multi-local* scheme. Furthermore, due to the data heterogeneity across participating clients, we cluster the local models into several groups and adopt a global model for each group. Each local model is assigned to its nearest global model to reduce the distances between local models and global models. We refer to this as *multi-global* scheme. Finally, we formulate the exchange between local and global models as a joint optimization problem, and propose an efficient aggregation approach to solve it. We refer to this as *Matching* algorithm.

Particularly, we evaluate it with the latest sequential learning neural network proposed for anomaly detection, which is similar to [8]. The framework is shown in Fig.1. We simply summarize the procedures as four steps, and leave the details of detection model and algorithmic framework in Section III: (1) the server sends the metadata (i.e., model parameters) to edge devices; (2) each edge device collects time-series data and clusters them into multiple detection models; (3) each model performs local training and uploads their model parameters to server; (4) the server clusters the received parameters and aggregates them to multiple global models, and the process repeats.

The novel contributions of this work are summarized in the following:

- We propose a novel Multi-Local and Multi-Global (MLMG) federated learning paradigm, to address the challenges of client heterogeneity and data diversity.
- We propose an optimized Matching algorithm to solve the joint optimization problem between multiple local and global models.
- We evaluate the novel paradigm with a sequential learning neural network for semi-supervised anomaly detection, and experiments of several benchmark datasets demonstrate better detection accuracy of our approach.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the detailed information of the novel federated learning paradigm. Section IV verifies the proposed method by several benchmark datasets. Finally, Section V makes a conclude of this paper.

## II. Related Work

The research of federated learning is proliferating in the last few years which enables clients to collaboratively train a machine learning model while reducing the systemic privacy risks. Due to the highly non-I.I.D distribution of clients, deploying more specialized models for specific scenarios has emerged as popular research. In this section, we summarize the studies concerned with specialized models to optimize the generation in federate training.

The natural framework for dealing with incongruent data is Multi-Task Learning [9], [10]. However, the techniques are applied in a centralized setting in which all data resides at one location and the server has full control over and knowledge about the optimization process. Recently, a distributed multi-task learning framework in a federated setting is discussed in [11]. Additionally, to solve the problems caused by data heterogeneity among clients, clustering algorithms with the purpose of grouping clients as their data similarity are researched. In the presence of adversarial data points, [4] studies a clustering algorithm running on center server leveraging the statistical model. [5] introduces a Clustered Federated Learning (CFL) framework, which aims at dividing the clients into clusters via clustering. [6] proposes a multi-center federated learning framework that generates multiple global models by stochastic expectation maximization (EM) algorithm. Also, [12] advocates an adaptive personalized federated learning algorithm (APFL), which aims to learn a personalized model for each device that is a mixture of optimal local and global models. [13] proposes a heterogeneity-aware federated optimization method, to provide convergence guarantees in federated learning. [14] adds a personalized layer for each local model to tackle heterogeneous. Moreover, the robustness and security issues have also received much attention. [15] studies the Byzantine-robust distributed learning from heterogeneous datasets. In addition, some researches pursue to improve the robustness of the global model against abnormal/adversarial clients through outlier-robust clustering [16], [17].

## III. Multi-Local and Multi-Global Model Aggregation

In this section, we introduce the construction of novel federated learning approach. We first provide a brief introduc-

tion of anomaly detection model on edge devices. Then, we present the federated learning algorithms utilized in this work. Afterwards, we give detailed information about the Multi-Local and Multi-Global (MLMG) model aggregation approach in federated learning.

### A. On-device Anomaly Detection Model

We take advantage of a single layer feedback neural network [18], which suggests a combination of Online Sequential Extreme Learning Machine (OS-ELM) [19] and Autoencoder [20] to perform sequential learning for anomaly detection. Below, we give a brief introduction to the model.

*1)* ***OS-ELM****:* OS-ELM is a sequential implementation of batch learning ELM [21], which is originally developed from a single hidden layer feedforward network (SLFN). Assuming an $n$-dimensional input chunk $x \in R^{k\times n}$ is given, where $k$ denotes the batch size; then the $m$-dimensional output chunk $y \in R^{k\times m}$ with respect to input $x$ can be computed as

$$y = G(x \cdot \alpha + b)\beta, \tag{1}$$

where $G(\cdot)$ is an activate function, $\alpha$ is an input weight, $b$ is a bias of the hidden node, and $\beta$ is an output weight.

Let $H \in R^{k\times \tilde{N}}$ be the hidden layer output matrix of the network, i.e., $H = G(x \cdot \alpha + b)$. If SLFN can approximate an $m$-dimensional target chunk $t \in R^{k\times m}$ with zero error, the optimal output weight $\hat{\beta}$ can be formulated as $\hat{\beta} = H^{\dagger}t$, where $H^{\dagger}$ is Moore-Penrose generalized inverse of matrix.

Moreover, OS-ELM uses sequential learning to update the output weights instead of batch learning is implemented. In the sequential learning phase, suppose the $i$-th training chunk $\{x_i \in R^{k_i\times n}, t_i \in R^{k_i\times m}\}$ is given, the output weights update equations are computed as follows

$$\begin{aligned} P_i &= P_{i-1} - P_{i-1}{H_i}^T(I + H_i P_{i-1}{H_i}^T)^{-1}H_i P_{i-1}, \\ \beta_i &= \beta_{i-1} + P_i{H_i}^T(t_i - H_i\beta_{i-1}). \end{aligned} \tag{2}$$

Specially, the output weights for an initial training set are determined by

$$\begin{aligned} P_0 &= (H_0^T H_0)^{-1}, \\ \beta_0 &= P_0 H_0^T t_0. \end{aligned} \tag{3}$$

*2)* ***Autoencoder****:* Autoencoder is a type of artificial neural network which leverages unsupervised learning algorithms to learn a representation (encoding) for an input. On the basis, it consists of two parts, the encoder and decoder. The encoder tends to compress the input $x \in R^{k\times n}$ into a latent-space representation $\tilde{x} \in R^{k\times \tilde{N}}$, while the decoder takes in the reduced encodings to produce the output $y \in R^{k\times n}$. Generally, the hidden nodes $\tilde{N}$ is less than input nodes $n$, i.e., $\tilde{N} < n$, referred to as undercomplete autoencoders. In the training phase, Autoencoder sets the target values to be equal to the inputs, i.e., $t = x$. The goal of Autoencoder is to generate a well-characterized dimensionality-reduced form so that it can reconstruct it as close as possible to its original input.

*3)* ***Anomaly Detection Model*** *:* Autoencoder is implemented with OS-ELM to perform anomaly detection. When it is combined with OS-ELM, the encoding result for input $x$ is represented as $H = G(x\cdot\alpha+b)$, and the decoding result that for decompressing the reduced form is $y = H \cdot \beta$. Thus, when the model is trained with a normal input pattern, the output tends to have a relatively large difference with the input in the case of anomaly. [18] can be referred to for more details.

### B. Federated Learning Algorithms

In this subsection, we highlight two related federated learning algorithms for a sequential learning neural network. Assume that $m$ clients participate in a federated setting. Particularly, each client $i$ has a private training dataset $D_i = \{X_i, Y_i\}$, where $X_i$ and $Y_i$ denote input features and corresponding labels for a sequential learning task, for example, $Y_i = \{0, 1\}$ in anomaly detection. We use $W_i$ to represent the model parameter of client $i$. After each client completes the sequential learning, all $W_i$ will be sent to the server. The server collects the received parameters and performs model aggregation to compute the weight $\widetilde{W}$ for global model.

*1)* ***Federated Averaging*** *($FedAvg$):* Most notably, $FedAvg$ takes the average of either model weight or gradient of the local models to train a global model [22]. By applying $FedAvg$ algorithm, the global model in an established round should be updated as $\widetilde{W} = \sum_{i=1}^{m} \frac{|D_i|}{|D|} W_i$. Particularly, $W_i = \{\beta_i, P_i\}$ in the sequential-learning neural network we examnied (see Equations (2)), hence, the global model should be updated as following formulas:

$$\begin{aligned} \beta &= \sum_{i=1}^{m} \frac{|D_i|}{|D|}\beta_i, \\ P &= \sum_{i=1}^{m} \frac{|D_i|}{|D|} P_i, \end{aligned} \tag{4}$$

where $\beta_i$ and $P_i$ represent the local model parameters, $\beta$ and $P$ represent the global parameters after aggregation, $|D_i|$ represents the number of training data points on client $i$, and $|D|$ represents the total number of data points on all $m$ participating clients.

*2)* ***Elastic Extreme Learning Machine*** *($E^2LM$):* $E^2LM$ model was originally proposed to cover the shortage of ELM, which has a weakness in learning large-scale training dataset. According to [23], the model calculates the most computation-expensive matrix multiplication with the incremental learning. [24] first employs $E^2LM$ to OS-ELM-based sequential training. In this work, we briefly extend the formulas to our federated learning framework. Suppose the model aggregation is written as a function of local parameters, that is $\widetilde{W} = \sum_{i=1}^{m} f(W_i)$. After completing the sequential learning on edge devices, the local parameters can be derived as $U_i = P_i^{-1}$, and $V_i = U_i\beta_i$ by using $\{\beta_i, P_i\}$. Then, the server performs aggregation according to $E^2LM$: $U = \sum_{i=1}^{m} U_i$,

$V = \sum_{i=1}^{m} V_i$. Finally, the weights of global model can be updated as:

$$\begin{aligned} \beta &= U^{-1}V, \\ P &= U^{-1}. \end{aligned} \tag{5}$$

### C. Multi-Local and Multi-Global Federated Learning

We conclude our MLMG framework as three steps: initialized clustering, distance-based sequential clustering, and matching. Below, we illustrate the procedures regarding three algorithms.

*1) Initialized clustering (Algorithm 1) :* In general, clustering algorithm incurs a high computational cost, therefore, an initialized clustering should be added in the first round of federated learning as initialization. Since the objective of MLMG is to separate data and clients as their diverse characteristics and non-I.I.D distribution, we adopt two clustering procedures, that is data clustering and user clustering. We take data clustering for example. In the initialized process, a clustering algorithm (e.g., K-Means) and validation function (e.g., Silhouette) cooperate to calculate the optimal number of clusters $K$ and their centroids. As illustrated in Algorithm 1, lines 3-4 compute clustering scores for each initialized $K$, and line 10 selects the optimal $K$ that achieves the best clustering result. Additionally, we carefully discuss whether clustering is necessary or not via thresholding (lines 7-8). If the best calculated score is above a threshold, the dataset should better be separated into several groups; otherwise, it will not be separated.

*2) Distance-based sequential clustering (Algorithm 2) :* Given the high cost of clustering, we cannot perform separation in sequential learning. Particularly, the training data is changing over time and further leads to dynamic changes in model parameters. To tackle this problem, we suggest a distance-based sequential clustering. For example, for each newly arrived data or updated local parameter, we compute its distance to existing cluster centroids (lines 3-5). $Dist$ denotes a function to measure the distance between new data and centroid, for example, L2 distance is used in this work. As long as all the distances are computed, the data can be assigned to the nearest cluster (lines 7-8).

*3) Matching algorithm (Algorithm 3):* Generally, all model parameters are fed into a shared global model, and receive the same results after model aggregation. To suit a multi-local and multi-global framework, we suggest an optiming matching algorithm. In particular, the matching algorithm consists of two procedures: Forward-propagation is used to update global models, and Back-propagation is used to update local models. In the Forward matching process, the local parameters are assigned to global models as their clustering results (lines 4-5); while in the Back matching process, we should first compare the number of clusters. If a client has more detection models than global models, all aggregated parameters will be back to the client to exert the effectiveness of multiple detection models (lines 12-13). This is because more local models imply more choices in the test phase, thus it is more likely to choose a suitable prediction model for test data to achieve higher detection accuracy. Conversely, if a client has fewer detection models than global models, the global parameters will be sent back as the clustering results (lines 16-17). After computing model parameters with MLMG federated learing, the prediction process is performed at all generated local models, as presented in [7]. More specially, the prediction result with the lowest value is used to determine whether the test data is normal or not.

**Algorithm 1** Initialized Clustering

1: # **Data clustering:**
2: **for** each participting client $i$ **do**
3: **for** each initialized cluster number $k = 2, 3, ..., n$ **do**
4: compute clustering score $s_{l,i}^{(k)}$ for dataset $D_i$
5: **end for**
6: $S_{l,i} = max\{s_{l,i}^{(2)}, s_{l,i}^{(3)}, ..., s_{l,i}^{(n)}\}$
7: **if** $S_{l,i} < \lambda$ **then**
8: $K_{l,i} = 1$
9: **else**
10: $K_{l,i} = \arg\max_{\theta \in [2,n]} s_{l,i}^{(\theta)}$
11: **end if**
12: group dataset $D_i$ into $K_{l,i}$ clusters
13: compute cluster centers $C_{l,i} = \{C_{l,i}^{(1)}, C_{l,i}^{(2)}, ..., C_{l,i}^{(K_{l,i})}\}$
14: **end for**
15: # **User clustering:**
16: **for** each initialized cluser number $k = 2, 3, ..., n'$ **do**
17: compute clustering score $s_g^{(k)}$ for all received model parameters $W$
18: **end for**
19: $S_g = max\{s_g^{(2)}, s_g^{(3)}, ..., s_g^{(n')}\}$
20: $K_g = \arg\max_{\theta \in [2,n']} s_g^{(\theta)}$
21: group local model parameters into $K_g$ clusters
22: compute cluster centers $C_g = \{C_g^{(1)}, C_g^{(2)}, ..., C_g^{(K_g)}\}$

## IV. Evaluation Results

In this section, we evaluate the proof-of-concept implementation of the proposed paradigm in federated learning. We first introduce the experimental setup, and then present the procedures and evaluation results.

We use anomaly detection as our research target, and employ the sequential learning neural network introduced in Section III to detect the anomalous data. Experiments are conducted with MNIST dataset [25], Federated CelebA (FedcelebA) dataset [26] and Audio dataset [27]. A statistical description of employed datasets and experimental setup is shown in Table 1.

**MNIST:** The MNIST dataset is a large dataset of handwritten digits from different people. In MNIST dataset, we assume five clients in a federated setting and distribute the data in a non-I.I.D way, e.g., the dataset partitioned in each client are with a limited number of classes. Particularly, we consider two

**Algorithm 2** Distance-based Sequential Clustering

**# Data clustering:**
**for** each participting client $i$ **do**
  **for** each newly arrived data/chunk $D_j$ **do**
    **for** each cluster $k = 1, 2, ..., K_{l,i}$ **do**
      $Dist_j^k = Dist(D_j, C_{l,i}^{(k)})$
    **end for**
    $r_j = \arg\min_{\theta \in [1, K_{l,i}]} Dist_j^\theta$
    update cluster centroid $C_{l,i}^{(r_j)}$
  **end for**
**end for**
**# User clustering:**
**for** each updated local parameter $W_t$ **do**
  **for** each cluster $k = 1, 2, ..., K_g$ **do**
    $Dist_t^k = Dist(W_t, C_g^{(k)})$
  **end for**
  $r_t = \arg\min_{\theta \in [1, K_g]} Dist_t^\theta$
  update cluster centroid $C_g^{(r_t)}$
**end for**

**Algorithm 3** Matching Algorithm

**# Forward-propagation Matching:**
**for** each participting client $i$ **do**
  **for** each local model $j$ in client $i$ **do**
    **if** $W_i^j$ is clustered into global cluster $k$ **then**
      $\widetilde{W}^{(k)} \leftarrow W_i^j$
    **end if**
  **end for**
**end for**
**# Back-propagation Matching:**
**for** each global cluster $k = 1, 2, ..., K_g$ **do**
  **for** each participting client $i$ **do**
    **if** $K_{l,i} \geq K_g$ **then**
      $W_i^k \leftarrow \widetilde{W}^{(k)}$
    **else**
      **for** each local model $j$ in client $i$ **do**
        **if** $W_i^j$ is clustered into global cluster $k$ **then**
          $W_i^j \leftarrow \widetilde{W}^{(k)}$
        **end if**
      **end for**
    **end if**
  **end for**
**end for**

cases, 2 classes per-client in case#1 and 5 classes per-client in case#2. In order to eliminated the impact of data points, all training and test data for the selected classes is used. Suppose that digits labeled as 0-8 are normal data, while the digits labeled as 9 are abnormal.

**CelebA:** The CelebA dataset is a large-scale face attributes dataset with more than 200K celebrity images. In this dataset, we perform federated learning in the non-I.I.D. sampling scenario with 100 randomly selected clients. We treat smiling images as normal and not smiling images as abnormal.

**Audio:** The audio dataset comprises of operating sounds of six types of toy/real machines. Each machine type has three or four machine IDs, which contributes to 23 machine IDs in total. In order to simulate a federated learning scenario, we simply identify each machine ID as a client, that is, 23 clients in Audio dataset. Normal and anomalous data corresponds to the real recordings collected from normal working machines and deliberately damaging target machines, respectively.

### *A. Experimental Results*

In this part, we discuss the evaluation results of different aggregation algorithms and different federated learning paradigms. Due to limited space of paper, we introduce the results in the first round of federated learning.

*1) **Comparison of federated algorithms**:* In the first experiment, we compare the $E^2LM$-based federated learning with the popular $FedAvg$ method. Also, the conventional centralized machine learning approach, which combines all data at one location to learn a prediction model is presented for comparison. The detailed results can be seen in Table 2. Note that the number of anomaly samples in the test dataset is limited up to 10% of that of normal samples to simulate a practical situation, that is, anomaly data are much rarer than normal data in most cases. It can be observed that the centralized approach is always superior or at least consistent with the federated learning algorithms, which proves that more training data usually results in a higher-quality detection model. Besides, we find federated algorithm of $E^2LM$ outperforms $FedAvg$ in all three datasets. As stated in [12], $FedAvg$ is sensitive to its hyperparameters due to the data diversity among clients, thus may not benefit from a favorable convergence guarantee. On the other hand, $E^2LM$ takes advantage of an incremental learning approach to achieve a higher generalization of a global model. Additionally, in terms of detection accuracy, we found case#1 (2 classes per-client) in MINST dataset achieves better results than case#2 (5 classes per-client), which implies less normal patterns makes it easier to develop a high-quality detection model.

*2) **Effectness of Multi-Local and Multi-Global federated paradigm**:* In this experiment, we compare our MLMG framework with the conventional single [2] and the latest multi-global federated learning systems [9]. Also, both the federated learning algorithms are evaluated, and Fig.3 proves better results of $E^2LM$, in which yellow bars are higher than grey bars. Moreover, we notice the "multi-global" framework in the medium does not always improve the detection accuracy than "single paradigm" on the left side due to the data diversity, especially in the case that the number of normal classes (Fig.3-(a)) and data points (Fig.3-(c)) is not sufficient. On the other hand, our "multi-local and multi-global" mechanism on the right side always achieves better results than "single paradigm" and "multi-global" designs, as it provides more flexibility in exploring data heterogeneity. The results demonstrate the effectiveness of our proposal.

*3) **Effect of the number of clusters**:* Besides the clustering algorithms, we also discuss the number of clusters in this

TABLE I
STATISTICS OF DATASETS AND EXPERIMENTAL SETUP FOR ANOMALY DETECTION

| | ***MNIST*** | ***CelebA*** | ***Audio*** |
|---|---|---|---|
| Content type | Handwritten digits | Facial images | Sound emitted from machines |
| Number of clients | 5 clients (case#1: 2 classes per-client; case#2: 5 classes per-client) | 100 clients (randomly chosen from 9343 users) | 23 clients (composed of 6 machine types) |
| Normal (trained class) | Digits labeled as 0-8 | Smiling images | Sound from normal machines |
| Abnormal | Digits labeled as 9 | Not smiling images | Sound from damaged machines |

TABLE II
ANOMALY DETECTION RESULTS OF THE CONVENTIONAL CENTRALIZED MACHINE LEARNING APPROACH AND TWO EXTENDED FEDERATED LEARNING APPROACHES

| | ***Training data*** | ***Test data*** (Normal) | ***Test data*** (Abnormal) | **FL Approach** | ***Precision*** | ***Recall*** | ***F-score*** | ***Accuracy*** |
|---|---|---|---|---|---|---|---|---|
| ***MNIST*** (case#1) | 60968 | 10118 | 1125 | $Centralized$ | 0.901 | 0.991 | 0.944 | 0.894 |
| | | | | $FedAvg$ | 0.9 | 0.984 | 0.94 | 0.887 |
| | | | | $E^2LM$ | 0.901 | 0.991 | **0.944** | 0.894 |
| ***MNIST*** (case#2) | 149543 | 24683 | 2750 | $Centralized$ | 0.9 | 0.981 | 0.939 | 0.885 |
| | | | | $FedAvg$ | 0.899 | 0.976 | 0.936 | 0.88 |
| | | | | $E^2LM$ | 0.9 | 0.981 | **0.939** | 0.884 |
| ***CelebA*** | 928 | 121 | 14 | $Centralized$ | 0.929 | 0.86 | 0.893 | 0.815 |
| | | | | $FedAvg$ | 0.913 | 0.777 | 0.839 | 0.733 |
| | | | | $E^2LM$ | 0.917 | 0.826 | **0.87** | 0.778 |
| ***Audio*** | 20119 | 5399 | 598 | $Centralized$ | 0.901 | 0.886 | 0.894 | 0.811 |
| | | | | $FedAvg$ | 0.889 | 0.698 | 0.782 | 0.65 |
| | | | | $E^2LM$ | 0.901 | 0.882 | **0.892** | 0.807 |

work. We employ Sihouette Analysis to determine the optimal number of clusters, for both multi-local scheme and multi-global scheme. Since the Silhouette Analysis validates the consistency of data for different $K$ values, starting from 2, we carefully discuss if local clustering is necessary or not (i.e., $K = 1$) for different clients via thresholding. For example, we notice only the machine type of $pump$ in Audio dataset obtains a high silhouette score with data clustering, thus the datasets of other machine types are not separated through data clustering. The optimal designs relative to the highest clustering score are marked as blue and red bars in Fig.3. It is observed that the clustering algorithm and Silhouette Analysis help compute the best design in most cases. In Fig.3-(a), the best design failes to achieve the highest detectin accuracy, because the clustering and proposed matching algorithm are heuristic approaches. Even so, we find that the computed "multi-local and multi-global" always achieves the best detection accuracy among all three federated learing frameworks. Also, the results enlighten us that the detection accuracy is not absolutely positively correlated with the number of global models.

## V. CONCLUSIONS

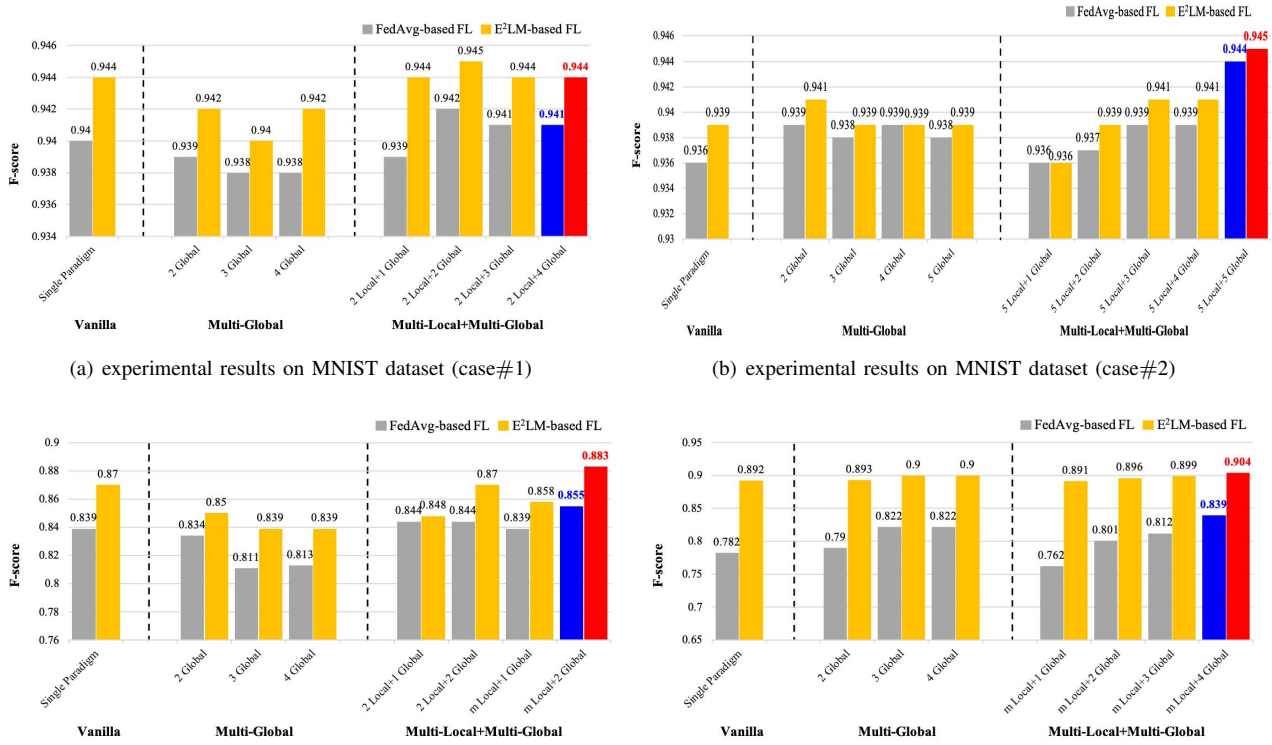
The ordinary federated learning approach adopts one single prediction model to train local data, and adopts one single global model to aggregate the information of clients. However, the vanilla federated learning framework is not able to deal with the challenges of incongruent data distributions and diverse data characteristics. To address the issues above, in this work, we propose a novel Multi-Local and Multi-Global (MLMG) federated learning paradigm to deploy specified models, and introduce a matching algorithm to derive appropriate exchanges between them. Evaluation results based on several benchmark datasets prove better performance of the novel paradigm compared with the existing federated learning frameworks.

## VI. ACKNOWLEDGMENTS

This work was supported by JST CREST Grant Number JPMJCR20F2, Japan.

## REFERENCES

[1] F. Learning, "Collaborative machine learning without centralized training data," 2016.
[2] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, and B. He, "A survey on federated learning systems: vision, hype and reality for data privacy and protection," *arXiv preprint arXiv:1907.09693*, 2019.
[3] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, 2020.
[4] A. Ghosh, J. Hong, D. Yin, and K. Ramchandran, "Robust federated learning in a heterogeneous environment," *arXiv preprint arXiv:1906.06629*, 2019.
[5] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
[6] M. Xie, G. Long, T. Shen, T. Zhou, X. Wang, and J. Jiang, "Multi-center federated learning," *arXiv preprint arXiv:2005.01026*, 2020.
[7] R. Ito, M. Tsukada, M. Kondo, and H. Matsutani, "An adaptive abnormal behavior detection using online sequential learning," in *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*. IEEE, 2019, pp. 436–440.




(a) experimental results on MNIST dataset (case#1)

(b) experimental results on MNIST dataset (case#2)

(c) experimental results on CelebA datase ( $1 \leq m \leq 5$, $m$ is integer)

(d) experimental results on Audio dataset ( $1 \leq m \leq 2$, $m$ is integer)

Fig. 2. Anomaly detection results of different federated algorithms in different federated learning frameworks. (Blue and red bars represent detection results under the best calcuted design using K-Means and Silhouette Analysis.)


[8] Y. Qin, H. Matsutani, and M. Kondo, "A selective model aggregation approach in federated learning for online anomaly (in press)," in *the 13th IEEE International Conference on Cyber, Physical and Social Computing (CPSCOM)*. IEEE, 2020.

[9] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[10] L. Jacob, J.-p. Vert, and F. R. Bach, "Clustered multi-task learning: A convex formulation," in *Advances in neural information processing systems*, 2009, pp. 745–752.

[11] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4424–4434.

[12] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," *arXiv preprint arXiv:2003.13461*, 2020.

[13] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smithy, "Feddane: A federated newton-type method," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019, pp. 1227–1231.

[14] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.

[15] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1544–1551.

[16] S. Gupta, R. Kumar, K. Lu, B. Moseley, and S. Vassilvitskii, "Local search methods for k-means with outliers," *Proceedings of the VLDB Endowment*, vol. 10, no. 7, pp. 757–768, 2017.

[17] R. Krishnaswamy, S. Li, and S. Sandeep, "Constant approximation for k-median and k-means with outliers via iterative rounding," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 2018, pp. 646–659.

[18] M. Tsukada, M. Kondo, and H. Matsutani, "A neural network-based on-device learning anomaly detector for edge devices," *IEEE Transactions on Computers*, 2020.

[19] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on neural networks*, vol. 17, no. 6, pp. 1411–1423, 2006.

[20] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[21] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, vol. 2. IEEE, 2004, pp. 985–990.

[22] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[23] J. Xin, Z. Wang, L. Qu, and G. Wang, "Elastic extreme learning machine for big data classification," *Neurocomputing*, vol. 149, pp. 464–471, 2015.

[24] R. Ito, M. Tsukada, and H. Matsutani, "An on-device federated learning approach for cooperative anomaly detection," *arXiv preprint arXiv:2002.12301*, 2020.

[25] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," 2010.

[26] S. Caldas, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.

[27] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda *et al.*, "Description and discussion on dcase2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring," *arXiv preprint arXiv:2006.05822*, 2020.