A Semi-Decentralized Collaborative Learning Framework Across Heterogeneous Devices for Personalized Predictive Analytics

Guanhua Ye, Hongzhi Yin,∗ and Tong Chen

organization=The University of Queensland, city=Brisbane, postcode=4072, country=Australia

ARTICLE INFO

Keywords:
On-Device Machine Learning
Personalized Predictive Analytics
Distributed Optimization
Similarity-based Aggregation

ABSTRACT

Due to the inevitable data sparsity on a single edge device, collaborative learning is regarded as an integral approach for learning on-device personalized deep neural networks (DNNs). However, traditional decentralized learning protocols communicate the knowledge by sharing the model parameters (i.e., weights or gradients). Consequently, such methods strictly require homogeneity of all participant models, thus being unable to handle applications where heterogeneous deep models are deployed. Besides, to support model personalization, a conventional practice is to pick a curated subset of similar users (i.e., neighbors) for knowledge aggregation, which potentially threatens user privacy with the need for exchanging sensitive personal information or model parameters. In this paper, we propose a Similarity-based Semi-Decentralized Knowledge Distillation (SD-Dist) framework for heterogeneous collaboratively distributed learning. By introducing a preloaded reference dataset and a cloud server, SD-Dist enables all participant devices to identify similar users and distil knowledge from them without any assumptions on a fixed model architecture. None of these operations will reveal any sensitive information like personal data and model parameters. Extensive experimental results on three real-life datasets show that SD-Dist can achieve 0.90% average accuracy improvement compared with the random decentralized distillation baseline. Furthermore, it saves 83.62% model parameters with negligible performance drop and enhances the resultant models’ robustness when users’ data is sparse and diverse.

1. Introduction

The emergence of wearable wireless sensors and the popularity of smart terminals have greatly improved the popularity of many on-device applications such as sports analytics (Gowda et al., 2017), affective interaction (Chen et al., 2015), and chronic disease monitoring (Allet et al., 2010). Among these researches, machine learning (Jordan and Mitchell, 2015) (ML) models, especially deep neural networks (DNNs) get established by exhibiting strong accuracy when accessing an enormous amount of personal data. However, it is usually unrealistic to train a generalized model based on a single user’s data (Zhang et al., 2021a). The main reason is that the samples collected from one individual are inadequate and skewed, leading to limited expressiveness and over-fitting of those deep models.

To address the data sparsity issue, a common practice is to collect as many users’ data as possible in an on-cloud global model (Baker et al., 2017). In this centralized setting, the user devices are only used for transmitting locally collected data to the server and displaying could-generated results. Consequently, the only way to obtain analytic results is to wait for the output from the global model, which can be devastating when monitoring fatal diseases under network delay or connection interruptions (Ye et al., 2022). Additionally, since the central server stores overall personal user data, the security of sensitive information and the reliability of the learned models are highly vulnerable to adversarial attacks (e.g., attribute inference attacks (Mosallanezhad et al., 2019; Zhang et al., 2022) and false data injection (Koh and Liang, 2017)). In this regard, on-device decentralized learning (Imteaj et al., 2021) was proposed to remove these bottlenecks of the classic paradigm, where the core is retaining both the private data and the model training process on edge devices (Vanhaesebrouck et al., 2017).

Federated Learning (FL) (Yang et al., 2019) has been a performant and representative decentralized learning paradigm, given its ability to allow collaborations among personal devices. In a general FL setting, each device possess a private dataset and trains machine learning models (e.g., DNNs) via corresponding optimization strategies (e.g., stochastic gradient descent) (Li et al., 2020). Then the edge devices will upload the learned model weights or gradients...
to the central server. A typical approach of collaborative learning is to aggregate the parameters received from all individual participants (e.g., by taking the average) and then transmit the updated values back to all devices (McMahan et al., 2017). In this way, FL ensures that each device maintains a local model for rapid analysis, and learns from all peers without access to their private data. However, there are two major drawbacks of such paradigms. Firstly, though an accurate global model can be learned and shared across devices, it inevitably leans towards the data distribution of the majority user group, and weakens the personalization effect on users sitting at the rear of the long-tail distribution. Despite the efforts in converting global model aggregation to personalized model aggregation with a subset of similar users (He et al., 2018; Koloskova et al., 2019; Zhang et al., 2021c), these frameworks put user privacy at risk as either the sensitive model parameters or user attributes are exchanged for identifying similar users for each device. Secondly, decentralized learning frameworks like FL require all participants to hold a homogeneous model architecture so that the parameters/gradientes from different devices can be shared. This further marks down the practicality of those decentralized learning paradigms, which usually means the model architecture has to be restricted by the device with the least computational capacity and memory (Bistritz et al., 2020), rather than supporting heterogeneity in model architectures for the same task to adapt to varying hardware capacities.

To address these two challenges, we design a Similarity-Based Semi-Decentralized Knowledge Distillation (SD-Dist) framework for heterogeneous model collaborations. In SD-Dist, instead of sharing the sensitive data or parameters, devices firstly send their soft decisions (i.e., predicted label distributions) on a preloaded reference dataset to the cloud server during training. By comparing the received soft decisions, the cloud server can quantify the similarity between arbitrary two devices without revealing any personal information or model parameters. Then all devices can mutually communicate with the neighbors assigned by the cloud server, where the sharing content is also their soft decisions. This privacy-preserving protocol allows each device to only communicate with peers that are most similar to itself, so as to enhance the personalization of all learned models. The soft decision itself, on the other hand, contains implicit knowledge for co-Distillation (Zhang et al., 2021b). In this way, SD-Dist enables classifiers with different model architectures to learn from each other, which is not possible in gradient-based (Geiping et al., 2020; Zhu et al., 2019) collaborative learning frameworks. As such, SD-Dist provides a secure way of learning personalized predictive models without any assumptions on unified network architectures across devices.

The primary contributions of this study are summarized as follows:

- We study a new collaborative learning paradigm driven by distributed knowledge distillation. The users in the new paradigm are allowed to possess a heterogeneous on-device predictive model that can fully utilize the computing resources of the device. These heterogeneous models communicate with its neighbors via soft labels on a public reference dataset to protect their sensitive personal data and model parameters.

- We design a novel SD-Dist framework to reinforce the paradigm in learning heterogeneous deep models. In SD-Dist, we develop a privacy-preserving similarity-based collaborative learning protocol to support selective inter-device communication, which enhances the efficacy of collaborative learning.

- We extensively evaluate our framework on three real-world datasets. The experimental results show that our SD-Dist achieves state-of-the-art classification accuracy with far less communication cost and computing resources. In addition, our SD-Dist is the most robust to the data sparsity that is quite common in real-life applications.

The organization of this paper is as follows. In Section 2 we introduce relevant background of this paper. The details of the proposed SD-Dist framework are provided in Section 4. We conduct a series of experiments and analyze the results in Section 5 and conclude the paper in Section 6.

2. Related works

2.1. Decentralized Learning

Decentralized learning paradigms include federated learning with a central coordinator and fully decentralized learning paradigm (Kempe et al., 2003). The former is also named as Semi-decentralized learning, where a central server will provide a communication graph and guide the connection pathway between devices (Zhang et al., 2021a). The latter does not involve any central coordinator (e.g., parameter server). It spreads the learned knowledge according to the crude communication graph to enlarge the horizon of each device hence generalizing the local model yet maintaining its personality (Boyd et al., 2006; Xiao and Boyd, 2004). Such optimization problem has been thoroughly studied in (Jakovetić et al., 2014; Scaman et al., 2018; Hammou et al., 2020; Urabe et al., 2021; Wang et al., 2021;
Many fully decentralized learning paradigms assume all devices communicate with others via short-range communication (e.g., Bluetooth) and the weights of edges in the communication graph are determined by the communication cost. However, if a device has no neighbor in physical (e.g., the distance to the nearest device exceeds the longest communication range), it is actually discharged from the network since it has to train the local model all alone. This can be a common phenomenon for some unpopular applications like rare chronic disease monitoring.

2.2. Knowledge Distillation

Knowledge distillation is originally introduced to take advantage of the experience from a pre-trained teacher model to improve the performance of student models with fewer model parameters (Phuong and Lampert, 2019). An unlabeled reference dataset will assist the knowledge transmission from the teacher to the student, which is also commonly adopted in semi-supervised learning (Zhu, 2005). Cheng et al. (2020) and Cho and Hariharan (2019) provide the theoretical or empirical analytics of the knowledge distillation. Among these studies, the response-based knowledge distillation only needs to share the soft decisions (i.e., the outputs of the last fully-connected layer) for communications (Chen et al., 2017). This is perfectly compatible with the heterogeneous device network where the classifiers have different model architectures. Guo et al. (2020) jointly trained some student models without any teacher model. Zhang et al. (2021b) further indicates that co-Distillation between two student models may outperform the teacher-student hierarchy. Nevertheless, these methods are concerned with boosting the performance of models over a fully-connected communication graph, which is unsuitable for large-scale decentralized device networks with communication constraints.

2.3. Neighbors in Collaboration

Intuitively, for a personalized model of an individual in a homogeneous subset, the experience from the subset is more valuable than the corpora. Collaborative filtering (CF) (Su and Khoshgoftaar, 2009) is an excellent example. In typical CF methods, a group of users that are similar to the target user are considered as its neighbors. The similarity is calculated based on users’ explicit data (e.g., personal information and ratings on some items) or implicit data (e.g., purchase history, browsing history, and search patterns) (Schafer et al., 2007). Kleinberg and Sandler (2003) show that the experience from neighbors is theoretically helpful for the personalized prediction. Such methods also work for the decentralized on-device learning framework (He et al., 2018; Koloskova et al., 2019; Zhang et al., 2021c). However, these frameworks either reveal the sensitive user data (Ye et al., 2022) or uncover the model parameters (He et al., 2018).

To sum up, our SD-Dist assumes all devices have access to the internet, and the communication costs between any two devices are the same. And the cloud server only distributes connection suggestions to each device but not holds any machine learning models. In the new setting, we determine the communication graph by the dynamic similarity between local models based on the soft-decision upon the public reference dataset. Since inference attack cannot restore local data from the soft-decision on the reference dataset, it preserves the privacy of all participants of the device network.

3. Problem Formulation

Consider a personalized predictive task that involves $N$ user devices. Each user device $n (1 \leq n \leq N)$ possesses a private machine learning model $\phi(\cdot)$ parameterized by $\theta^n \in \mathbb{R}^{p_n}$ where $p_n$ denotes the parameters’ dimensionality that differs among devices. Meanwhile, the $n$-th user device possesses a local dataset $D_n = \{(x^n_i, y^n_i)\}_{i=1}^{M_n}$ that contains $M_n$ personal and sensitive data samples. The $x^n_i$ denotes the input feature vector and $y^n_i$ is an one-hot vector over all classes representing the corresponding ground truth. Our goal is to learn a function that maps every input data point to the correct class. We define the local objective function $L_{loc}$ of a single user device $n$ w.r.t. $D_n$ as:

$$L_{loc}(\theta^n, D_n) = \sum_{i=1}^{M_n} \ell(\phi(\theta^n, x^n_i), y^n_i),$$

where $\ell$ is a loss function (i.e., cross-entropy in our case) that quantifies the classification error.

For a simpler isolated network optimization strategy without communication, the only training resource of some device $n$ is the local dataset. So for each device we wish to minimize the gap between $\phi(\theta^n, x^n_i)$ and $y^n_i$. Apparently,
4. Methodology

This section introduces the design and principle of the SD-Dist framework. We first provide preliminaries for the personalized model, the reference dataset, and the device network. Afterwards, we demonstrate the approach for on-device heterogeneous collaborative learning.

4.1. Preliminaries

Definition 1: Reference Dataset. All devices possess an identical, unlabeled reference dataset $D_r = \{\mathbf{x}_j\}_{j=1}^Q$ and will produce a soft decision of $D_r$, i.e., the probability distribution over all classes $s^n = \{\phi(\theta^n, \mathbf{x}_j)\}_{j=1}^Q$ at each iteration. The reference dataset is easily accessible in most applications, e.g., using desensitized public benchmarks or synthetic data samples. Devices can then exchange knowledge with each other via those soft decisions without revealing any private data or model parameters. But, it is impractical to communicate with all other devices, especially in a large-scale device network. Therefore, we introduce inter-model distance to rectify the communication neighbors of every device. The distance between devices $n$ and $m$ is:

$$d_{nm} = \frac{1}{Q} \sum_{j=1}^Q \|s^n_j - s^m_j\|^2,$$

(3)

where we use $c_{nm} = \frac{1}{d_{nm}}$ to represent the similarity between devices $n$ and $m$. Notice that $c_{nm}$ is non-constant as models in $n$ and $m$ are dynamically updated during training.

Definition 2: The Device Network. The cloud server in semi-decentralized learning will maintain a communication graph to coordinate the connection between devices. Let $G = (\mathcal{A}, \mathcal{E}, \mathcal{C})$ be a communication graph, where $\mathcal{A} = \{1, \cdots, N\}$ is the user devices set, and $\mathcal{E} \subseteq \mathcal{A} \times \mathcal{A}$ indicates the set of edges between user devices. $\mathcal{C} \subseteq \mathbb{R}^{N \times N}$ is a dynamic weight matrix, where the weight of edge $(n, m) \in \mathcal{E}$ is denoted by $c_{nm} \in \mathcal{C}$. Here we specify that $c_{nm} = \frac{1}{d_{nm}}$, which is the similarity between device $n$ and device $m$. The weight matrix $\mathcal{C}$ will be updated as the similarity between devices changes during training. We use $\mathcal{K}^n = \{\theta^n_1, \theta^n_2, \cdots, \theta^n_K\}$ to indicate device $n$’s neighbor set, where $\theta^n_m \in \mathcal{K}^n$ is the personalized model of the $m$-th nearest neighbor (i.e., reaches $m$-th highest $c_{nm}$) of $n$. Each device in the network will receive its $\mathcal{K}^n$ after $G$ is updated.

4.2. Heterogeneous Distributed Learning with Knowledge Distillation

In personalized prediction scenarios, the amount of data on each user device is limited, e.g., most e-health users do not have a long and diverse list of medical records for learning an automated diagnosis model. Meanwhile, if deployed on IoT sensors, a fully isolated model is prone to the inaccuracy caused by unreliable signals or noises collected from a signal device. These two main obstacles hold the isolated paradigm back from a practicality perspective. On the contrary, centralized learning paradigms can leverage data gathered from all users to learn a global model, thus minimizing the negative impact of data insufficiency and noise. But such non-personalized models usually sacrifice the correctness of the minority to obtain better overall accuracy. Furthermore, this amplifies the privacy concerns as well as incurs a high demand on resource-intensive computing facilities.

In light of this, we propose a similarity-based semi-decentralized information exchanging protocol, enabling the collaboration among devices when learning personalized models. Unlike conventional gradient-sharing or weight-sharing methods, devices in SD-Dist merely need to share their soft decisions on a reference dataset, which drastically reduces the risk of privacy breach for both data and models. Besides, considering the communication bandwidth in real personalized applications is restricted, each device only communicates with its neighbors, i.e., a small alterable group of devices determined by the dynamic weight matrix $\mathcal{C}$ in the cloud server. Specifically, neighbors can be formed for

the learning objective of the whole network will be:

$$\min_{\mathbf{\theta}^1, \cdots, \mathbf{\theta}^N} \sum_{n=1}^N \mathcal{L}_{loc}(\mathbf{\theta}^n, D_n).$$

(2)

When the volume of every local dataset $M_n$ is big enough for training an independent and satisfactory on-device model, Eq.2 is the optimum solution for both individuals and the collectivity.
device $n$ by picking the top-$K$ highest scores from the $n$-th row of $C$, denoted by $c_n$. For a given device $n$, the degree of alienation between $n$ and its $K$ neighbors $\mathcal{K}^n$ is defined as:

$$L_{\text{ref}}(\theta^n, D_r) = \sum_{j=1}^{Q} \left\| \phi(\theta^n, \bar{x}_j) - \frac{1}{K} \sum_{m \in \mathcal{K}^n} \phi(\theta^m, \bar{x}_j) \right\|^2,$$

where $\phi(\theta^n, \bar{x}_j) = s^n$ is the soft decisions from the local device $n$ w.r.t the reference sample $\bar{x}_j \in D_r$, and $\frac{1}{K} \sum_{m \in \mathcal{K}^n} \phi(\theta^m, \bar{x}_j)$ is the ensemble of soft decisions from $n$'s neighbors on the same sample.

We hereby propose the overall objective for training all distributed devices in the network:

$$L^* = \min_{\theta_1, \ldots, \theta_N} \frac{1}{N} \sum_{n=1}^{N} \left( L_{\text{loc}}(\theta^n, D_n) + \rho L_{\text{ref}}(\theta^n, D_r) \right),$$

where $\rho$ is a regularization parameter. Essentially, solving this equation means that the algorithm converges to the following set of distillation stationary points:

$$S^* = \left\{ \theta \mid \nabla_{\theta} \left( \sum_{i=1}^{M_n} \ell'(\phi(\theta^n, x_i^n), y_i^n) + \rho \sum_{j=1}^{Q} \left\| \phi(\theta^n, \bar{x}_j) - \frac{1}{K} \sum_{m \in \mathcal{K}^n} \phi(\theta^m, \bar{x}_j) \right\|^2 \right) = 0, \forall n \right\}.$$

In this way, SD-Dist requires all devices to reach for the optimum on their local data and the consensus on the reference data with their neighbors at the same time. Through Eq. 4, SD-Dist allows knowledge from peer models to be passed through, while bypassing any assumptions on homogeneous model architectures.

### 4.3. SD-Dist in The Device Network

The SD-Dist is presented in detail in Algorithm 1. To reach the stationary status described in Eq. 6, SD-Dist iteratively minimizes the local loss as well as the neighborhood alienation in each device. Lines 4—7 are the communication step. The device $n$ first generates its own soft decisions w.r.t the shared reference dataset with its newest model, and send it to the cloud server. After that, the server will update $C$ and assign the corresponding neighbors $\mathcal{K}^n$ to device $n$. Then device $n$ shares the soft decisions with its neighbors and exchange learned knowledge with each other. We provide a device-level view on SD-Dist in Fig. 1, where the core similarity-based communication steps are highlighted in red. After $c_n$ is updated in an iteration, the top-$K$ among them will be adopted (i.e., stored in $\mathcal{K}^n$). Then, line 9 is the model update step where device $n$ computes new parameters with the help of knowledge from its neighbors. Lastly, the local model is pruned with Stripe-Wise Pruning$^1$ (SWP) technique to save memory and computation workload for small wearable devices. This pruning step also ensures the models on different devices are heterogeneous since it customizes each model by trimming the filters.

### 4.4. SD-Dist in The Device Network

#### 5. Experiment

In this section, extensive experiments are conducted on real-world datasets to evaluate the feasibility of our SD-Dist in three personalized predictive classification tasks. Specifically, the following four research questions (RQs) are studied in our experiments.

**RQ1:** How is the performance competitiveness of our method in classification tasks compared with the baselines?

**RQ2:** How robust can SD-Dist be while increasing the sparsity of local training data?

**RQ3:** How efficient SD-Dist is in terms of consumed compute resources?

**RQ4:** How do the hyperparameters influence our SD-Dist when performing real-life personalized analytics?
Heterogeneous Distributed Collaborative Learning

Figure 1: An instance of device $n$. It possesses a local training set $D_n$, a local label set $Y_n$ and a preloaded reference dataset $D_r$. $D_r$ is identical across all $N$ devices. The model will yield $Y_{loc}$ and $Y_{ref}$ separately. During training, the device $n$ will send $Y_{ref}$ to the cloud server and receive $c_n$ (i.e., the $n$-th row of the weight matrix $C$). Then the device $n$ will communicate devices with the $K$-highest similarity to generate $Y_{neighbor}$. The auto model pruning procedure makes the device network heterogeneous.

Algorithm 1 Similarity-Based Distributed Distillation

**Initialization:** Let $t = 0$ be the index of iterations, $I$ be the communication interval. Let $\theta_0^n$ be the initial variables of device $n$, $s_{0}^{n} = \{\phi(\theta_{0}^{n}, \tilde{x}_j)\}_{j=1}^{Q}$ be the initial soft-decision w.r.t reference dataset from device $n$. Let $U^n_t$ be the set of received soft-decision from other devices in the network. Let $\eta_t$ be the learning rate.

1: repeat
2: for unconverged $n = 1, \ldots, N$ do
3: if $t \bmod I = 0$ then
4: $s_{t-1}^{n} \leftarrow \{\phi(\theta_{t-1}^{n}, \tilde{x}_j)\}_{j=1}^{Q}$
5: Send $s_{t-1}^{n}$ to the cloud server
6: Receive updated $c_n$ from the cloud server
7: Communicate with device group $\{m \in A \text{ with top-}K \ c_{nm} \}$ to generate $K^n_t$
8: end if
9: $\theta_t^n \leftarrow \theta_{t-1}^{n} - \eta_t \frac{\partial}{\partial \theta_t^{n}} \sum_{(x,y) \in D_n} \nabla L(\phi(\theta_{t-1}^{n}, x), y) - \frac{2\eta_t}{M} \sum_{x \in D_r} \nabla \phi(\theta_{t-1}^{n}, \tilde{x})^T (\phi(\theta_{t-1}^{n}, \tilde{x}) - \frac{1}{K} \sum_{m=1}^{K} s_{t-1}^{m}))$
10: Stripe-wise model pruning
11: $t = t + 1$
12: end for
13: until termination condition satisfied

5.1. Datasets and Baselines

Three real-life datasets are used in the experiments, and their primary statistics are shown in Table 1.

The first dataset is Sleep Cassette (SC) database (Mourtazaev et al., 1995), which includes 153 overnight polysomnography (PSG) recordings. We extract the Electroencephalogram (EEG) data from PSD recordings for sleep quality rating. Three labels, i.e, Awake, non-rapid eye movement sleep (NREM), and rapid eye movement sleep (REM) are employed to replace the original Rechtschaffen and Kales sleep stage annotations (Rechtschaffen, 1968) for simplifying the classification task.

The second dataset is PhysioNet Apnea-ECG Dataset (PAD) (Ichimaru and Moody, 1999). It is a famous dataset

---

1https://github.com/fxmeng/Pruning-Filter-in-Filter
Heterogeneous Distributed Collaborative Learning

Table 1

Statistics of experimental datasets.

| Dataset | #Individual | #Sample    | #Class |
|---------|-------------|------------|--------|
| SC      | 153         | 142,481    | 3      |
| PAD     | 70          | 33,776     | 2      |
| MNIST   | -           | 70,000     | 10     |

regarding a common sleep-related disease – Obstructive Sleep Apnea (OSA). PAD contains 70 overnight ECG recordings from severe patients, moderate patients and normal people. The ECG signals in PAD are converted into RR interval (pulse to pulse interval) signals via the algorithm provided by Cai and Hu (2020). Specifically, the model will tell whether there is an apnea event based on a 60-dimensional RR-interval (time interval between two R-peak) vectors. Such classification pipelines greatly facilitate OSA patients’ self-monitoring since they can get detection results via wireless wearable devices like smart-watches and smart-bands (Ye et al., 2021).

Besides, we further conduct experiments on Modified National Institute of Standards and Technology (MNIST) database (LeCun et al., 1998). It has been widely adopted to evaluate the model performance in benchmarking distributed machine learning algorithms. Since there is no explicit relationship between samples for grouping (e.g., samples from one individual can be regarded as one device in SC and PAD), we conduct a random and evenly segmentation on the MNIST, which proposed by Bistritz et al. (2020), to make it compatible with decentralized learning scenarios.

We compare our SD-Dist with the following three baseline optimization strategies:

- **FD-Dist**: It is a Fully-Connected Distributed Distillation framework where any device will communicate with all other devices in the network.

- **RD-Dist**: It is a Random Distributed Distillation framework where one device in the network will communicate with a static group of devices at each iteration (Bistritz et al., 2020). The device group is randomly sampled from the network. Notice that when the size of the group is as large as the whole network \( |A| \), RD-dist is equivalent to FD-Dist.

- **I-SGD**: Every device in the network has an independent model and optimizes the parameters without cross-device collaborations.

5.2. Experimental Setting

In SC and PAD, each recording of a patient is regarded as one slice, so they have 153 and 70 slices, respectively. We randomly and evenly divide samples in the MNIST into 50 slices. For each dataset, 10 of the slices are randomly selected and combined as the reference dataset, and the rest of them are regarded as the local datasets of user devices. That is to say, \( N_{SC} = 143, N_{PAD} = 60, N_{MNIST} = 40 \). We then randomly split the samples in each local dataset for training, validation, and test with a ratio of 8:1:1.

We use ResNet56 (He et al., 2016) as a benchmark DNN model to generally evaluate the feasibility of our SD-Dist framework. For the SC and PAD datasets (time series), all the 2D-convolutional layers in the ResNet56 are replaced with 1D-convolutional layers. We adopt SWP with the default pruning criterion provided by Meng et al. (2020) to trim the convolutional filters in a stripe-wise manner. Specifically, each \( 3 \times 3 \) filter in the ResNet56 has 9 stripes and parts of them will be pruned using \( L1 \)-regularization, which ensures model heterogeneity across the network. We randomly select four pruned models on each dataset and visualize the first filter at layer-1 in Fig. 2 to further demonstrate the divergence and heterogeneity among on-device models. Such pruning operations are also friendly to IoT devices with small computational capacity and memory. The local loss \( \ell \) is specified as cross-entropy for all devices as it is commonly used in such classification tasks. An open-source P2P library \(^2\) is employed to support inter-device communication. Specifically, each device holds a unique node in the network. All nodes are connected with each other so they can broadcast or receive soft decisions w.r.t the reference dataset via TCP/IP channel.

Accuracy (Acc) is adopted to evaluate the performance of SD-Dist as well as three baselines on all datasets. The hyperparameter tuning was conducted via grid search. The optimal values and search intervals are listed in Table 2.

\(^2\)https://github.com/macsnoeren/python-p2p-network
Figure 2: The first filter at layer-1 of four randomly selected models on three datasets, where white color indicates the pruned strips. Apparently, pruning operations will introduce divergence among models.

Table 2
Hyperparameter settings.

| Dataset | Hyperparameter | Value | Search Interval |
|---------|----------------|-------|-----------------|
| SC      | $K$            | 10    | $\{2, 4, 6, 8, 10, 12\}$ |
|         | $\rho$         | 0.01  | $\{0.01, 0.02, 0.04, 0.08\}$ |
| PAD     | $K$            | 8     | $\{2, 4, 6, 8, 10, 12\}$ |
|         | $\rho$         | 0.01  | $\{0.01, 0.02, 0.04, 0.08\}$ |
| MNIST   | $K$            | 12    | $\{2, 4, 6, 8, 10, 12\}$ |
|         | $\rho$         | 0.04  | $\{0.01, 0.02, 0.04, 0.08\}$ |

5.3. Effectiveness Discussion (RQ1)

We draw the following observations based on the performance of our SD-Dist and baselines on three datasets listed in Table 3. Note that here we assume all devices possess abundant training data on SC and PAD, which is impractical in real-life applications.

Firstly, SD-Dist consistently outperforms RD-Dist on all datasets, and the average performance gain is 0.83%. In particular, on the PAD dataset with higher user diversity, our method significantly boosts the quality of personalized analytics (1.34% improvement). This validates the efficacy of the similarity-based connection in the proposed SD-Dist framework.

Secondly, though the I-SGD achieves the lowest accuracy on the MNIST, it outperforms all other methods on SC and PAD. In general, introducing inter-device communication will enhance the learning ability of distributed models. The results on the MNIST dataset is in line with this intuition. The unusual results on SC and PAD suggest that the knowledge from neighbors can be deleterious to local models. This assumption is reasonable when the local dataset on each device is sufficient enough for training a personalized model. In this case, the model aggregation step actually brings noises into the local model, thus leading to performance decay. Additional experiments are conducted in Section 5.4 as a further investigation of this observation.

Lastly, the results imply that the SD-Dist is very close to the FD-Dist, and even surpasses it on the PAD dataset. Since the devices in FD-Dist will communicate with all other devices ($K = N - 1$) during training, FD-Dist is the theoretic skyline for both SD-Dist and RD-Dist ($K < N - 1$). In fact, RD-Dist achieves an inferior performance on all datasets. A possible explanation is that some devices are ‘beneficial’ to the local model and some are ‘detrimental’. If devices are randomly selected as neighbors (RD-Dist), the skyline would be FD-Dist. But if only ‘beneficial’ ones are
Table 3
Performance of SD-Dist and baselines on three datasets

| Dataset | SD-Dist | FD-Dist | RD-Dist | I-SGD  |
|---------|---------|---------|---------|--------|
| MNIST   | 0.9701  | 0.9733  | 0.9681  | 0.9337 |
| SC      | 0.8420  | 0.8434  | 0.8305  | 0.8451 |
| PAD     | 0.9206  | 0.9137  | 0.9072  | 0.9239 |

Figure 3: (a) and (b) depict the performance of SD-Dist with different numbers of neighbors (i.e., SD-Dist \( K = 4 \) and SD-Dist \( K = 8 \)) and baselines on SC and PAD when the data sparsity \( r \) goes down. The I-SGD is the worst method on both datasets when \( r \leq 10\% \). Our SD-Dist constantly outperforms RD-dist when having the same number of neighbors.

5.4. Robustness to Data Sparsity (RQ2)

We answer RQ2 by investigating the performance of the proposed heterogeneous semi-decentralized collaborative learning framework via a set of sparse data training simulations. The simulated data sparsity is implemented by randomly extracting \( r\% \) of samples in SC and PAD datasets. Tested network communication strategies include FD-Dist, I-SGD, SD-Dist (with 4 and 8 neighbors), and RD-Dist (with 4 and 8 neighbors). The performance is depicted in Figure 3. When \( r \) decreases from 100 to 0.1, all the methods suffer a distinct performance decline, especially I-SGD. Apparently, as less data is available for the training processes, the over-fitting issue becomes more serious, which significantly hurts the model’s generalization. We also notice that increasing the number of neighbors \( K \) will remarkably stimulate the model robustness to different levels of data sparsity. The largest gains on SC and PAD are 64.15\% and 24.78\%, respectively. In general, collaborating with more neighbors can achieve higher accuracy, except in one special case. SD-Dist \( (K = 4) \) outperforms RD-Dist \( (K = 8) \) on PAD when \( r = 1 \). It implies that the divergence between selective collaboration (SD-Dist) and random collaboration (RD-Dist) is amplified when the likelihood of model over-fitting rises (fewer available training data). In other words, the local model is more fragile confronting ‘detrimental’ neighbors. To be specific, the mean improvements resulted from replacing random connection with similarity-based connection when \( r = 10, 1 \) and 0.1 are 1.08\%, 3.23\% and 3.49\%, respectively.

In a nutshell, inter-device communication allows each device to get more information from each other when local data is limited, which empowers distributed distillation methods to resist data sparsity. Moreover, distributed distillation methods with similarity-based neighbor filtration, our SD-Dist, can further strengthen the robustness to the data sparsity.
Table 4
Performance of SD-Dist and unpruned SD-Dist on all three datasets.

| Dataset | Metric | SD-Dist* | SD-Dist |
|---------|--------|----------|---------|
| MNIST   | Acc    | 0.9941   | 0.9701  |
|         | Params(M) | 0.87     | 0.01    |
|         | FLOPs(M) | 252.00   | 1.35    |
| SC      | Acc    | 0.8714   | 0.8420  |
|         | Params(M) | 0.87     | 0.19    |
|         | FLOPs(M) | 251.10   | 52.76   |
| PAD     | Acc    | 0.9433   | 0.9206  |
|         | Params(M) | 0.87     | 0.23    |
|         | FLOPs(M) | 251.10   | 64.25   |

5.5. Resource Efficiency (RQ3)

We then evaluate the resource efficiency of SD-Dist by comparing it with SD-Dist*, where the unpruned ResNet56 is deployed on all devices. The experimental results are reported in Table 4. The average volume of model parameters (Params) and floating point operations (FLOPs) are the metrics to measure the efficiency of computing resources that are vital to IoT devices with constrained resources.

On average, our SD-Dist saves 83.52% model parameters and 84.31% FLOPs with only 2.54% accuracy loss. Obviously, the model pruning significantly augments the resource efficiency with a negligible accuracy drop.

5.6. Hyperparameter Sensitivity (RQ4)

In this section, we showcase the effect of two hyperparameters, namely $K$ and $\rho$, on the performance of SD-Dist.

Firstly, we enlarge $K$ from 2 to 12 while $\rho$ follows the optimal setting in this Section 5.3. We illustrate the outcomes in Fig. 4(a)-4(c). The accuracy figures of I-SGD and FD-Dist are also presented as a reference line for $K = 0$ and $K = N - 1$. The classification performance of RD-Dist and SD-Dist on MNIST constantly goes up as $K$ increases and converges to the green dotted line (FD-Dist). This validates that distributed distillation can transfer knowledge across devices. On the other hand, the average gap between RD-Dist and SD-Dist on MNIST is 0.36%, which is significantly smaller than on SC (1.35%) and PAD (1.49%). This indicates that selective collaboration shows little superiority over random collaboration on MNIST, where the local training sets on all devices obey identical distribution (the local datasets of MNIST are generated via random sampling). For the SC and PAD datasets with the ‘BYO’ dataset partition (i.e., taking the record of each patient as a local dataset), selective collaboration persistently excels the random collaboration. Additionally, there is a counter-intuitive observation in Fig 4(c) – the SD-Dist exceeds the FD-Dist when $K > 6$ and starts to descend and converge to the green dotted line after $K > 8$. We try to explain this phenomenon with a thought experiment. Imagine an extreme scenario where four devices, namely A to D, compose a network. A and B only have negative samples, while C and D only have positive samples. The models on all devices can reach 100% accuracy by labeling all samples as one class. It is conceivable that introducing fully inter-device communication will worsen the overall accuracy (which used to be 100%). However, if every device in the network only communicates with the most similar one (i.e., A with B, C with D), the overall accuracy may remain unchanged. This thought experiment reveals how different communication strategies influence the performance of decentralized learning algorithms under extreme data distribution. Now let’s look back to the experimental results on the PAD dataset. The green dotted line is inferior to the red one, which means some of the models perform worse after coupling global knowledge. The orange and blue lines are inclined to converge to the green dotted line. This is because when $K$ increases to $N - 1$, the random connection method and the similarly-based connection method are equivalent to the full-connection one. The blue line exceeds the green dotted line suggests that the knowledge from a small group of analogous neighbors can be more helpful than the global knowledge, which is the core value of adopting SD-Dist.

Next, we show the test accuracy of SD-Dist on all datasets in Fig. 4(d) when $K = 8$ and $\rho$ varies in {0.01, 0.02, 0.04, 0.08}. The results suggest that the best performance on MNIST is obtained when $\rho$ is between 0.02 and 0.08. Meanwhile, higher $\rho$ leads to slight performance decay on SC and PAD. This suggests that a network with homogeneous data distribution benefits more directly and swiftly from inter-device information sharing.
The Impact of NeighborNum on MNIST

I-SGD
FD-Dist
RD-Dist
SD-Dist

The Impact of NeighborNum on SC

I-SGD
FD-Dist
RD-Dist
SD-Dist

The Impact of NeighborNum on PAD

I-SGD
FD-Dist
RD-Dist
SD-Dist

SD-dist with different
MNIST
SC
PAD

Figure 4: The performance of SD-Dist and baselines on MNIST (a), SC (b) and PAD (c) with different numbers of neighbors \( K \), and the performance of SD-Dist on three datasets with different value of \( \rho \).

6. Discussion

In this paper, we design the SD-Dist, a novel heterogeneous semi-decentralized collaborative learning framework, to support on-device personalized DNNs training. Boosted by the similarity-based co-distillation communication protocol, our SD-Dist achieves satisfactory performance compared with conventional randomly-connected distributed distillation. Furthermore, our SD-Dist shows unique superiority when the data is highly sparse and even exceeds the theoretical skyline when the data distribution becomes skewed across devices.

6.1. Theoretical Implications

In distributed learning paradigms, a device in the network only communicates with a limited number of peers (i.e., neighbors). Conventional methods randomly assign neighbors for all devices via a preset communication graph (Bistritz et al., 2020), or construct static connections guided by external information (e.g., demographics (Ye et al., 2022)). The latter usually achieves better performance but will reveal sensitive personal information during knowledge sharing. Meanwhile, high similarity on static external information does not necessarily lead to a high performance gain in the collaboration. To this end, we design a novel communication protocol where the connections are constructed based on dynamic model-wise similarity (i.e., inter-device similarity). To further protect the privacy of model param-
eters, the inter-device similarity is calculated based on the secure soft decisions on a public reference dataset rather than the local sensitive data. Experiments on real-life datasets have verified the feasibility of the proposed approach.

### 6.2. Practical implications

Our SD-Dist allows each on-device model to reach its full potential rather than being restricted by the device with the least computing resources in traditional distributed learning paradigms. The core technique to support such heterogeneous communication is the knowledge co-distillation proposed in our paper. Strengthened by the dynamic linkages between devices, the co-distillation in SD-Dist empowers the whole network to resist data sparsity and distribution imbalance, which are common problems in practical IoT scenarios. Experiments on real-life datasets exhibit the immense potential of SD-Dist in on-device predictive applications.

**CRediT authorship contribution statement**

**Guanhua Ye**: Conceptualization, Methodology, Software, Investigation. **Hongzhi Yin**: Supervision, Resources, Validation, Writing-Reviewing and Editing. **Tong Chen**: Visualization, Data curation, Writing-Original draft preparation.

### Acknowledgements

This work is supported by Australian Research Council Future Fellowship (Grant No. FT210100624), Discovery Project (Grant No. DP190101985).

### References

Allet, L., Knols, R. H., Shirato, K., and Bruin, E. D. d. (2010). Wearable systems for monitoring mobility-related activities in chronic disease: a systematic review. *Sensors*, 10(10):9026–9052.

Baker, S. B., Xiang, W., and Atkinson, I. (2017). Internet of things for smart healthcare: Technologies, challenges, and opportunities. *IEEE Access*, 5:26521–26544.

Bistritz, I., Mann, A., and Bampos, N. (2020). Distributed distillation for on-device learning. *Advances in Neural Information Processing Systems*, 33.

Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006). Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530.

Cai, W. and Hu, D. (2020). Qrs complex detection using novel deep learning neural networks. *IEEE Access*, 8:97082–97089.

Chen, G., Choi, W., Yu, X., Han, T., and Chandraker, M. (2017). Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30.

Chen, M., Zhang, Y., Li, Y., Hassan, M. M., and Alamri, A. (2015). Aiwac: affective interaction through wearable computing and cloud technology. *IEEE Wireless Communications*, 22(1):20–27.

Cheng, X., Rao, Z., Chen, Y., and Zhang, Q. (2020). Explaining knowledge distillation by quantifying the knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12925–12935.

Cho, J. H. and Hariharan, B. (2019). On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4794–4802.

Geiping, J., Bauermeister, H., Dröge, H., and Moeller, M. (2020). Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947.

Gowda, M., Dheke, A., Shen, S., Choudhury, R. R., Yang, L., Golwalkar, S., and Essanian, A. (2017). Bringing {IoT} to sports analytics. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, pages 499–513.

Guo, Q., Wang, X., Wu, Y., Yu, Z., Liang, D., Hu, X., and Luo, P. (2020). Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11020–11029.

Hammou, B. A., Lahcen, A. A., and Mouline, S. (2020). Towards a real-time processing framework based on improved distributed recurrent neural network variants with fasttext for social big data analytics. *Information Processing & Management*, 57(1):102122.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, L., Bian, A., and Jaggi, M. (2018). Cola: Decentralized linear learning. *Advances in Neural Information Processing Systems*, 31.

Ichimaru, Y. and Moody, G. (1999). Development of the polysomnographic database on cd-rom. *Psychiatry and clinical neurosciences*, 53(2):175–177.

Imteaj, A., Thakker, U., Wang, S., Li, J., and Amini, M. H. (2021). A survey on federated learning for resource-constrained iot devices. *IEEE Internet of Things Journal*, 9(1):1–24.

Jakovetić, D., Xavier, J., and Moura, J. M. (2014). Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146.

Jordan, M. I. and Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260.

Kempe, D., Dobra, A., and Gehrke, J. (2003). Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 482–491. IEEE.
Heterogeneous Distributed Collaborative Learning

Kleinberg, J. and Sandler, M. (2003). Convergent algorithms for collaborative filtering. In *Proceedings of the 4th ACM conference on Electronic commerce*, pages 1–10.

Koh, P. W. and Liang, P. (2017). Understanding black-box predictions via influence functions. In *ICML*, pages 1885–1894.

Koloskova, A., Stich, S., and Jaggi, M. (2019). Decentralized stochastic optimization and gossip algorithms with compressed communication. In *International Conference on Machine Learning*, pages 3478–3487. PMLR.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR.

Meng, F., Cheng, H., Li, K., Luo, H., Guo, X., Lu, G., and Sun, X. (2020). Pruning filter in filter. *Advances in Neural Information Processing Systems*, 33:17629–17640.

Mosallanezhad, A., Beigi, G., and Liu, H. (2019). Deep reinforcement learning-based text anonymization against private-attribute inference. In *EMNLP*, pages 2360–2369.

Mourtazaev, M., Kemp, B., Zwidermann, A., and Kamphuisen, H. (1995). Age and gender affect different characteristics of slow waves in the sleep eeg. *Sleep*, 18(7):557–564.

Phuong, M. and Lampert, C. (2019). Towards understanding knowledge distillation. In *International Conference on Machine Learning*, pages 5142–5151. PMLR.

Rechtschaffen, A. (1968). A manual for standardized terminology, techniques and scoring system for sleep stages in human subjects. *Brain information service*.

Scaman, K., Bach, F., Bubeck, S., Massoulié, L., and Lee, Y. T. (2018). Optimal algorithms for non-smooth distributed optimization in networks. *Advances in Neural Information Processing Systems*, 31.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer.

Song, S., Fang, Z., and Jiang, J. (2022). Fast-drd: Fast decentralized reinforcement distillation for deadline-aware edge computing. *Information Processing & Management*, 59(2):102850.

Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009.

Urabe, Y., Rzepka, R., and Araki, K. (2021). Find right countenance for your input—improving automatic emoticon recommendation system with distributed representations. *Information Processing & Management*, 58(1):102414.

Vanhaseebrouck, P., Bellet, A., and Tommasi, M. (2017). Decentralized collaborative learning of personalized models over networks. In *Artificial Intelligence and Statistics*, pages 509–517. PMLR.

Wang, Q., Yin, H., Chen, T., Yu, J., Zhou, A., and Zhang, X. (2021). Fast-adapting and privacy-preserving federated recommender system. In *The VLDB Journal*, pages 1–20.

Xiao, L. and Boyd, S. (2004). Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78.

Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., and Yu, H. (2019). Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(3):1–207.

Ye, G., Yin, H., Chen, T., Cui, L., and Zhang, X. (2021). Fenet: A frequency extraction network for obstructive sleep apnea detection. *JBHI*, 25(8):2848–2856.

Ye, G., Yin, H., Chen, T., Xu, M., Nguyen, Q. V. H., and Song, J. (2022). Personalized on-device e-health analytics with decentralized block coordinate descent. *IEEE Journal of Biomedical and Health Informatics*.

Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., and Gao, Y. (2021a). A survey on federated learning. *Knowledge-Based Systems*, 216:106775.

Zhang, H., Hu, Z., Qin, W., Xu, M., and Wang, M. (2021b). Adversarial co-distillation learning for image recognition. *Pattern Recognition*, 111:107659.

Zhang, R., Song, M., Li, T., Yu, Z., Dai, Y., Liu, X., and Wang, G. (2021c). Democratic learning: hardware/software co-design for lightweight blockchain-secured on-device machine learning. *Journal of Systems Architecture*, 118:102205.

Zhang, S., Yin, H., Chen, T., Huang, Z., Nguyen, Q. V. H., and Cui, L. (2022). Pipattack: Poisoning federated recommender systems for manipulating item promotion. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1415–1423.

Zhu, L., Liu, Z., and Han, S. (2019). Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32.

Zhu, X. J. (2005). *Semi-supervised learning literature survey*. Guanhua Ye et al.: *Preprint submitted to Elsevier*