Federated Learning With Selective Knowledge Distillation Over Bandwidth-constrained Wireless Networks

Gad Gad*[1], Zubair Md Fadlullah*[2], Mostafa M. Fouda†‡[3], Mohamed I. Ibrahem§[4], and Nei Kato¶[5]
*Department of Computer Science, Western University, London, ON, Canada.
†Department of Electrical and Computer Engineering, Idaho State University, Pocatello, ID, USA.
‡Center for Advanced Energy Studies (CAES), Idaho Falls, ID, USA.
§School of Computer and Cyber Sciences, Augusta University, Augusta, GA 30912, USA.
¶Graduate School of Information Sciences, Tohoku University, Sendai, Japan.
Emails: [1]ggad@uwo.ca, [2]zfadlullah@ieee.org, [3]mfouda@ieee.org, [4]mibrahem@augusta.edu, [5]kato@it.is.tohoku.ac.jp

ICC 2024 - IEEE International Conference on Communications | 978-1-7281-9054-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICC51166.2024.10622906

***Abstract*—Artificial Intelligence (AI) applications on Internet of Things (IoT) networks often involve relaying generated data to a server for deep learning training, which poses security risks to users' data. Federated Learning (FL) offers a distributed model training paradigm in which local data are kept at the edge and locally trained models are exchanged and aggregated by a server over several rounds to produce a global model. While successful, standard FL algorithms do not support heterogeneous local model design, an essential requirement, especially for resource-limited edge devices. Recently, Knowledge Distillation-based FL algorithms have provided model-agnostic FL to enable clients to independently design their local model and share soft labels instead of model parameters. KD-based FL algorithms are computationally expensive due to additional distillation training. We propose Federated Learning with Selective Knowledge Distillation (FedSKD) to address the limitations of system heterogeneity; and computation and communication demands. We evaluate different aspects of the proposed algorithm relative to baseline FL algorithms. Results show that FedSKD incurs significantly less per-round computation time and communication overhead relative to the considered model-based and KD-based FL algorithms.**

***Index Terms*—Machine Learning, Federated Learning, Knowledge Distillation, Edge devices, IoT.**

I. Introduction

Federated Learning (FL) offers a distributed alternative solution to central deep learning by training local models on users' devices without data sharing [1]–[3]. This is especially beneficial for IoT applications [4]–[6] in which sending data centrally overlooks edge computing resources and increases communication overhead.

Standard FL (FedAvg [7]) imposes a global model architecture on participating FL clients. However, participating clients' devices have heterogeneous resources and independent model architecture design is preferable for optimal resource utility across the network. Tackling the communication overhead and system heterogeneity challenges gave rise to a new class of FL algorithms called Knowledge Distillation-based (KD) FL. KD was first introduced as a compression technique where a small model (called the student) distills the representational knowledge of a larger trained model (called the teacher model) [8].

Despite the numerous advantages of KD-based FL algorithms, the computation cost of this class of FL is substantially higher than standard model-based FL algorithms. This is attributed to the longer training period that clients' of KD-based algorithms undergo to perform distillation training in addition to local training.

In this paper, we address this challenge and propose a KD-based FL algorithm with a low computation and communication cost. Specifically, our contribution is the Federated Learning with Selective Knowledge Distillation (FedSKD) algorithm which has many advantages over model-based and KD-based FL algorithms. First, unlike model-based FL algorithms that enforce a global model architecture on all clients, undermining the system heterogeneity IoT applications are characterized with, FedSKD leverages a shared public dataset to distill and transfer ensemble clients' knowledge across the network. Since soft labels are model-agnostic, no global model architecture is imposed on clients, hence each client designs its local model according to its resources and needs. Secondly, an integral step of KD-based FL algorithms is distillation training in which the aggregated soft labels are used to train the client's local models for several epochs each round in addition to local training. This step introduces significant computation overhead. FedSKD returns a particular client only these global soft labels that belong to classes which the client's local soft labels produced by its local model are far from.

The paper's sections are arranged as follows. Section II lays a brief overview of the literature on Federated Learning and communication-efficient FL techniques. Section III introduces relevant preliminaries and presents the proposed FedSKD framework. The experiments and performance evaluation are given in section IV. Section V presents and discusses the obtained results on communication efficiency, per-round computation time, and test accuracy. Finally, section VI concludes the methodology and findings in this paper.

II. Related Work

Federated Learning (FL) is a decentralized approach that preserves data privacy by enabling devices to train local models and share updates [7], [9]. The key challenges addressed in this paper that face FL deployment on edge devices include system heterogeneity, and communication and computation overheads. System heterogeneity refers to the diverse and resource-constrained nature of edge devices. To address the system heterogeneity, recent work incorporated KD with FL leveraging soft label aggregation as an alternative strategy to model aggregation [10]–[13]. As a model-agnostic algorithm, clients under KD-based FL algorithms design their local models independently. Many recent works have also addressed communication efficiency in FL. For example, Deep Gradient Compression [14] introduces a gradient-based FL algorithm where locally-clipped sparse gradients are shared by clients and aggregated by the server. Clients then receive the aggregated gradients to update their models. However, a downside of such an approach is that it may result in performance degradation with high compression ratios.

Knowledge Distillation (KD) [8], [15] was introduced to transfer knowledge from a larger model (teacher) to a smaller one (student) in centralized settings [16], [17]. This method has been adapted for FL [11], [12], [18], [19] eliminating the need for model parameter sharing, which is the crux of model-based FL algorithms [7], [9]. KD-based FL algorithms leverage a shared unlabeled public dataset. For instance, [11] utilizes a proxy dataset D_p accessible to all clients for soft labels, which are aggregated at the server. The global soft labels are then broadcast to clients to perform distillation training. Gad *et al.* [12] introduced, FedAKD, which incorporates Mixup augmentation [20] to synthesize an augmented public dataset from the public dataset each round. Server-controlled parameters are used to sync the synthesized public dataset across clients. One of the disadvantages of KD-based FL algorithms is that they require high computational demand due to the additional distillation training step that clients have to undergo to distill knowledge after receiving the global soft labels [21], [22].

In general, previous FL algorithms can be categorized into model-based FL algorithms that impose a global model architecture on clients, exhausting weaker edge devices that can not run large models, and KD-based Fl algorithms which support independent local model design but introduce a large computation overhead due to the time of distillation training. This work proposes a KD-based FL algorithm that offers a balanced solution that ensures both supporting the independent design of local models and affordable and adjustable local computation budget.

III. Method

A. Preliminaries

Federated Learning. In Federated Learning (FL), we aim to create a global model using data from N_k clients. Each client has a local dataset D_i, where $i \in N_k$. Each data sample is represented as x_i^n with label $y_i \in \{1, 2, \ldots, C\}$. The global dataset, D, comprises all local datasets:

$$D = \{D_1, D_2, \ldots, D_{N_k}\}, \tag{1}$$

$$D = \sum_{i=1}^{N_k} D_i \tag{2}$$

The goal in FL is to train a global model g_i collaboratively across clients to minimize the empirical loss over D. This can be represented by the following optimization problem:

$$\begin{aligned} &\min_{w_1,\ldots,w_U} \quad \sum_{i=1}^{U} L_i(g_i) \\ &\text{s.t.} \quad w_1 = w_2 = \ldots = w_U, \quad \forall i \in \{1, \ldots, U\}, \end{aligned} \tag{3}$$

where g is the aggregated model weights given by:

$$g = \frac{\sum_{i=1}^{U} |D_i| w_i}{\sum_{i=1}^{U} |D_i|} \tag{4}$$

and $L_i(g_i)$ is the local loss for the i^{th} client which is given by:

$$L_i(g_i) = \frac{1}{\|D_i\|} \sum_{n=1}^{D_i} L_{CEE}(g_i; x_i, y_i), \tag{5}$$

where L_{CE} denotes the Cateogrical Cross-Entropy loss function.

The optimization problem 3 states that the goal of FL is to train a local model $w_1 = w_2 = \ldots = w_U$ on client devices such that the aggregate model g achieves a minimum score on all local losses.

Knowledge Distillation. Beginning with Knowledge Distillation (KD) in the context described in [24], for a C-way classification, the class probability for a given output $z \in \mathbb{R}^C$ is:

$$p_j = \frac{e^{z_j/T}}{\sum_{c=1}^{C} e^{z_c/T}}, \tag{6}$$

where T is the temperature scaling parameter. For $T = 1.0$, it defaults to the usual Softmax.

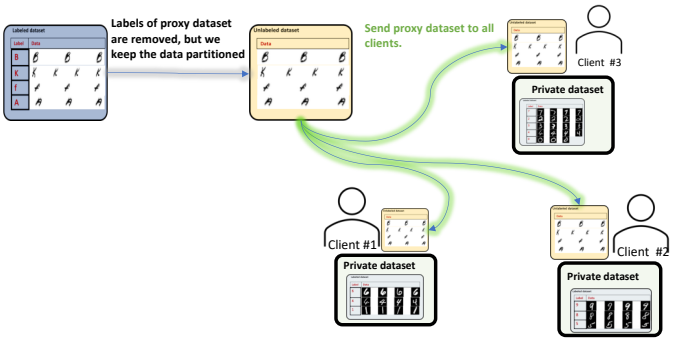

Fig. 1: To use a labeled dataset as a proxy dataset for Knowledge Distillation, the labels are first removed, and the proxy dataset is distributed to all the FL clients.

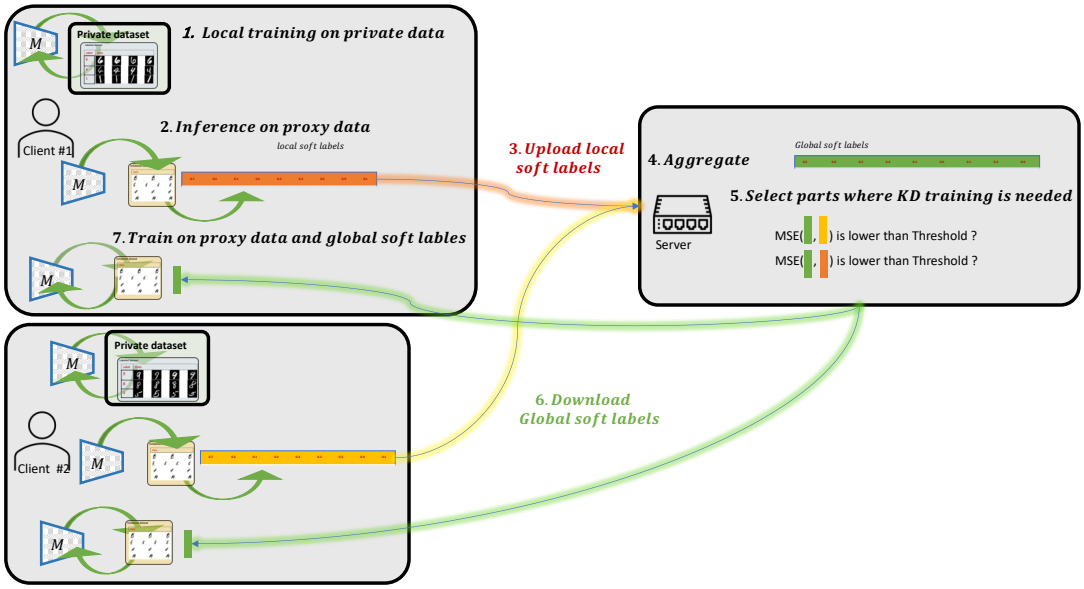

Fig. 2: An overview of the main steps in FedSKD.

TABLE I: A comparison of the characteristics of considered FL algorithms.

Algorithm	Type	Heterogeneous Local Models	Low Communication Overhead	Low Per-Round Computation Overhead	High Test Accuracy
FedAvg [7]	Model-based			✓	✓
FedMD [11]	KD-based	✓	✓		
FedAKD [12]	KD-based	✓	✓		
CFedAKD [23]	KD-based	✓	✓		
FedSKD (ours)	KD-based	✓	✓	✓	✓

In KD-based FL, KD is repurposed from the central setting to the Federated Learning. While in the central training setting, a student model distills the knowledge from a teacher model, in the FL context, a proxy/public dataset is shared among all clients on which they calculate their local soft labels. If an originally labeled dataset is chosen as a proxy dataset, its labels are removed before distributing the data to each client, as shown in Fig. 1. The proxy dataset functions as a knowledge transfer medium: When a client wants to transfer its learned knowledge to other clients, it generates local soft labels from the proxy dataset (bypassing the proxy dataset as inputs to its model) then it uploads these local soft labels to the server, which then aggregates them into global soft labels and downloads the global soft labels to all clients, and when a client wants to acquire knowledge from other clients, it downloads the global soft labels and trains its model to minimize the distance between its local soft labels and the downloaded global soft labels. Therefore, the purpose of KD is to transfer knowledge across clients by minimizing the distance between clients' local soft labels and the aggregated global soft labels. Notice that in 1 we keep the data partitioned class-wise even after discarding the labels, keeping the data partitioned according to their labels will be useful in the 5^{th} step in FedSKD shown in Fig. 2 where instead of downloading the whole aggregated global soft labels from the server to all clients. We can reduce the size of the global soft labels downloaded by first examining the distance between the global soft labels and the local soft labels of each client. This examination is performed on the label level. So, the distance between the global soft labels that belong to a certain class and the local soft labels of a client ki that belong to the same class is calculated and only the global soft labels whose distances exceed a predefined threshold are downloaded. This simple approach results in saving significant bandwidth due to not downloading all global soft labels. Additionally, KD training time is reduced greatly due to the proxy data that belong to classes whose global soft labels were not downloaded are excluded from KD training.

B. Problem Description

The FL setup considered in this paper is described as follows. Consider N_k clients/devices. Every device possesses a minor labeled dataset $D_i, \forall i \in N_k$, which could originate from identical or varying distributions. In the KD-based FL context, an additional public dataset D_p is assumed to be accessible to all entities. Each device independently constructs its distinct model architecture g_i. The primary objective is to devise a collaborative framework that boosts the accuracy of g_i beyond

local training performance using the shared public dataset D_p to distill knowledge across clients.

C. Federated Learning with Selective Knowledge Distillation

In this section, the proposed algorithm called Federated Learning with Selective Knowledge Distillation (FedSKD) is introduced to solve the problem given in III-B.

In FedSKD, all clients share a public dataset used for knowledge distillation as an alternative to model aggregation [7]. Therefore, each client designs his own local model architecture. The local soft labels for the i^{th} client in round r are denoted as p_i^r. The server aggregates local soft labels from all clients $p_i^r, \forall i \in \{1, 2, \ldots, N_c\}$ into p^r. Other KD-based algorithms listed in table I work by broadcasting the global soft labels p^r to all clients. However, the server under FedSKD examines the local soft labels received from clients and compares each with the aggregated global soft labels for each class k. The server then returns to client i only those global soft labels whose Mean Squared Error (MSE) with p_i^r is at least $\mathcal{T}_{select}$. It is worth mentioning that although the public dataset is unlabeled, we consider the classes of the private dataset to be also the classes of the public dataset since the models being trained on the public dataset are the client's local model that is designed to work on the private dataset. Let $p_{i,k}^r$ represent the local soft labels of class k of the i^{th} client and p_k^r represent the global soft labels of class k, at the r^{th} FL round.

Specifically, in each FL round, the server calculates a distance matrix $\mathbf{D} \in \mathbb{R}^{N_c \times K}$ which captures the distance between the local soft labels produced by each client $p_{i,k}^r$ and the global soft labels p_k^r, for every class k. After the distance matrix $\mathbf{D}$ is calculated, the server considers class k, and normalizes the distance between the local soft labels of the i^{th} client on class k and the global soft labels on class k. Let $\mathcal{T}_{select} \in [0, 1]$ be a variable that represents a distance threshold. The server downlink packet to client i consists of global soft labels that belong to q classes where $q < K$, this is because if the normalized distance value corresponding to the soft labels belonging to class k denoted as $\mathbf{D}[i][k]$, where i is the index of the client and k is the class, is less than the threshold $\mathcal{T}_{select}$, then the global soft labels of that class, denoted p_k^r, are not downloaded to that particular client. In this case, we assume that this client does not need to perform distillation training on global soft labels of class k since its local soft labels are already close to the global soft labels for class k.

By picking which global soft labels to download to each client based on the distance between his local soft labels and the global soft labels, FedSKD not only significantly reduces the communication overhead, as shown in Figs. 4 and 5, it also introduces a significant reduction in computational demand, as reflected by the reduced per-round time in Fig. 3, as clients are trained on less classes each round under the distillation training step. However, the local training remains untouched in both model-based and KD-based FL algorithms.

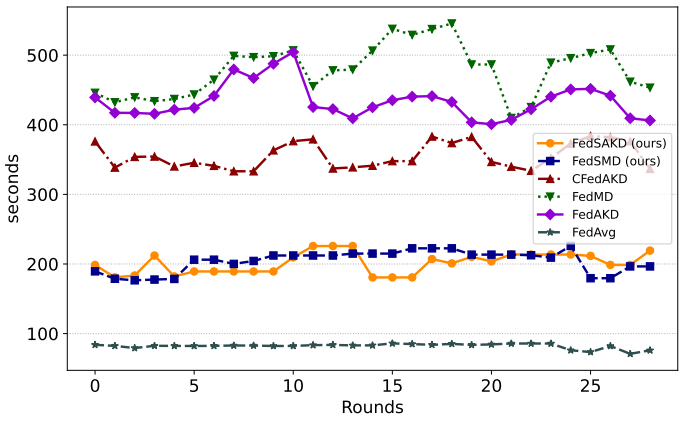

Fig. 3: The average round time for federated learning algorithms on MNIST dataset. Our proposed selective knowledge distillation mechanism reduces the average time to 200 seconds/round while clients under other KD-FL algorithms take 400 seconds/round.

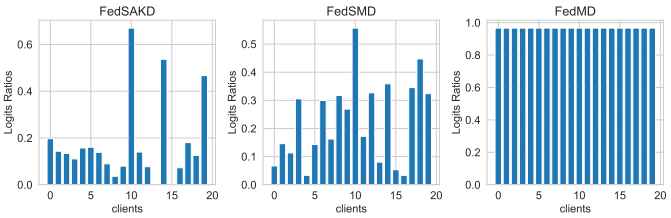

Fig. 4: Ratio of public logits used by each client (averaged over all rounds).

IV. Experiments and Performance Evaluation

To evaluate the efficiency of the proposed algorithm, FedSKD, two versions of the proposed algorithm were implemented with slight differences. The first algorithm is called Federated Learning with Selective Model Distillation (FedSMD), it is based on the vanilla KD-based FL algorithm presented in [11]. The second implementation of FedSKD is called Federated Learning with Selective Augmented Knowledge Distillation (FedSAKD) which is based on [23] which uses Mixup augmentation [20] to generate a synthetic public dataset synced across clients each round. Both algorithms were run on two datasets: CIFAR100 and MNIST, and three key performance aspects were evaluated with respect to baseline FL algorithms.

The size of the public dataset was set to 5,000 for the CIFAR100 dataset and to 10,000 for the MNIST dataset. Both implemented versions of the proposed algorithms use the class selection mechanism on top of the recent algorithms they are built upon achieving lower communication overhead and less computation time per round. The baseline FL algorithms employed in this comparative analysis are FedAvg [7], FedMD [11], FedAKD [12], CFedAKD [23]. The previous FL algorithms were evaluated with the proposed FedSMD and FedSAKD on both datasets in terms of communication overhead, per-round computation time, and test accuracy.

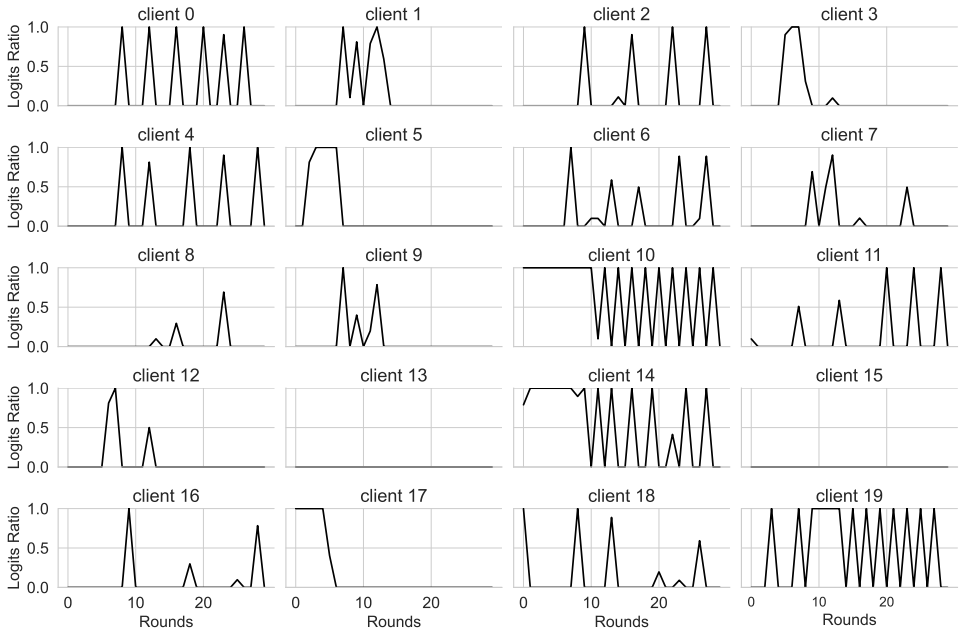

Fig. 5: Ratio of public logits used by each client per round. Using selective knowledge distillation, each client downloads only the logits that he needs to perform based on the discrepancy between his locally calculated logits and the aggregated logits.

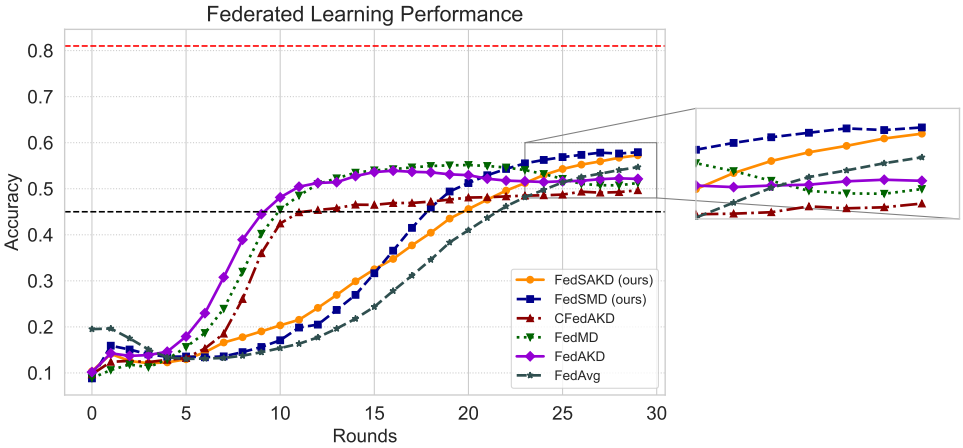

Fig. 6: Comparing the test accuracy of different federated learning algorithms. Our proposed selective knowledge distillation mechanism boosts the performance of KD-based FL algorithms while significantly reducing the communication overhead.

V. Results

The obtained results from the experiments on both datasets considering model-based (FedAvg) and KD-based (FedMD, FedAKD, CFedAKD) FL algorithms against our proposed KD-based FL algorithms (FedSMD, FedSAKD) show that KD-based FL algorithms are superior in terms of communication overhead. On the other hand, FedAvg is considerably less computationally demanding than KD-based FL algorithms. However, both FL paradigms (model-based and KD-based) achieve comparable results in terms of the test accuracy.

Fig. 4 shows the ratio of global soft labels used by each client (averaged over all rounds). Using selective knowledge distillation, each client downloads only the logits that he needs to perform based on the discrepancy between his locally calculated logits and the aggregated logits. As can be observed, clients under both variants of the proposed algorithm: FedSAKD and FedSMD incur significantly less per-round communication overhead compared to the vanilla KD-based Fl algorithm FedMD [11]. Fig. 5 provides per-client plots showing the ratio of global soft labels sent by the server relative to the total global soft labels that the server calculated by aggregating local soft labels.

The test accuracy achieved by the considered model-based and KD-based Fl algorithms over 30 FL rounds is shown in Fig.

6. The proposed FedSAKD and FedSMD outperform all considered FL algorithms, highlighting the importance of eliminating extra distillation training not only for communication and computation efficiency but also for improving the overall test accuracy of the algorithm. Finally, the per-round computation time from the MNIST simulation shown in Fig. 3 shows the per-round time for each considered FL algorithm averaged across clients. As expected, FedAvg has the least computation overhead of less than 100 seconds per round because it does not perform distillation training. KD-based algorithms perform distillation training using the global soft labels as labels in addition to local training where clients train on the private dataset. This additional training step significantly increases the computation cost. FedMD, FedAKD, and CFedAKD achieved an average of 500, 400, and 350 seconds per round, respectively. As for FedSAKD and FedSMD, the use of the selective feature to select the specific global soft labels returned to each client cut the computation time to an average of 200 seconds per round.

VI. Conclusion

In this paper, we consider the challenges of system heterogeneity and communication and computation efficiency in model-based and KD-based FL algorithms. We propose Federated Learning with Selective Knowledge Distillation (FedSKD) which leverages a shared public dataset as a medium for knowledge distillation across the FL environment. FedSKD adopts a selective approach where the server determines specific global soft labels to return to each client, based on the proximity between its local and the global soft labels.

To validate the efficacy of FedSKD, it was tested against standard baselines using two datasets. Results highlight FedSKD's ability to reduce the size of soft labels broadcasted by the server which results in a reduction in per-round computation time and communication overhead. The crux of FedSKD is the selective knowledge distillation feature which increases the efficiency of KD-based Fl algorithms, especially for IoT applications characterized by limited resources.

Code Availability

The implementation of algorithms and experiments in this paper can be accessed at https://github.com/gadm21/FedSKD (last accessed on 10 October 2023).

References

[1] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, article no. 106775, 2021.

[2] M. M. Badr *et al.*, "Privacy-preserving federated-learning-based net-energy forecasting," in *SoutheastCon 2022*, 2022, pp. 133–139.

[3] M. I. Ibrahem, M. Mahmoud, M. M. Fouda, B. M. ElHalawany, and W. Alasmary, "Privacy-preserving and efficient decentralized federated learning-based energy theft detector," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 287–292.

[4] M. M. Badr, M. M. E. A. Mahmoud, Y. Fang, M. Abdulaal, A. J. Aljohani, W. Alasmary, and M. I. Ibrahem, "Privacy-preserving and communication-efficient energy prediction scheme based on federated learning for smart grids," *IEEE Internet of Things Journal*, vol. 10, no. 9, pp. 7719–7736, 2023.

[5] Y. Gupta, Z. M. Fadlullah, and M. M. Fouda, "Toward asynchronously weight updating federated learning for AI-on-edge IoT systems," in *2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*, 2022.

[6] N. Nasser, Z. M. Fadlullah, M. M. Fouda, A. Ali, and M. Imran, "A lightweight federated learning based privacy preserving B5G pandemic response network using unmanned aerial vehicles: A proof-of-concept," *Computer Networks*, vol. 205, article no. 108672, 2022.

[7] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.

[8] J. Gou, L. Sun, B. Yu, S. Wan, and D. Tao, "Hierarchical multi-attention transfer for knowledge distillation," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 20, no. 2, pp. 1–20, 2023.

[9] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[10] D. Yao, W. Pan, Y. Dai, Y. Wan, X. Ding, C. Yu, H. Jin, Z. Xu, and L. Sun, "FedGKD: Toward heterogeneous federated learning via global knowledge distillation," *IEEE Transactions on Computers*, vol. 73, no. 1, pp. 3–17, 2024.

[11] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[12] G. Gad and Z. Fadlullah, "Federated learning via augmented knowledge distillation for heterogenous deep human activity recognition systems," *Sensors*, vol. 23, no. 1, article no. 6, 2022.

[13] Z. Chen, P. Tian, W. Liao, X. Chen, G. Xu, and W. Yu, "Resource-aware knowledge distillation for federated learning," *IEEE Transactions on Emerging Topics in Computing*, vol. 11, no. 3, pp. 706–719, 2023.

[14] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," *arXiv preprint arXiv:1712.01887*, 2017.

[15] Q. Xu, Y. Li, J. Shen, J. K. Liu, H. Tang, and G. Pan, "Constructing deep spiking neural networks from artificial neural networks with knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7886–7895.

[16] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[17] T. Kim, J. Oh, N. Kim, S. Cho, and S.-Y. Yun, "Comparing Kullback-Leibler divergence and mean squared error loss in knowledge distillation," *arXiv preprint arXiv:2105.08919*, 2021.

[18] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-efficient federated learning via knowledge distillation," *Nature communications*, vol. 13, no. 1, article no. 2032, 2022.

[19] X. Gong, A. Sharma, S. Karanam, Z. Wu, T. Chen, D. Doermann, and A. Innanje, "Preserving privacy in federated learning with ensemble cross-domain knowledge distillation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 11, 2022, pp. 11 891–11 899.

[20] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *International Conference on Learning Representations*, 2018.

[21] G. Gad, Z. M. Fadlullah, K. Rabie, and M. M. Fouda, "Communication-efficient privacy-preserving federated learning via knowledge distillation for human activity recognition systems," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 1572–1578.

[22] G. Gad, A. Farrag, Z. M. Fadlullah, and M. M. Fouda, "Communication-efficient federated learning in drone-assisted IoT networks: Path planning and enhanced knowledge distillation techniques," in *2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2023.

[23] G. Gad, "Light-weight federated learning with augmented knowledge distillation for human activity recognition," Master's thesis, Lakehead University, 2023.

[24] T. Kim, J. Oh, N. Y. Kim, S. Cho, and S.-Y. Yun, "Comparing Kullback-Leibler divergence and mean squared error loss in knowledge distillation," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021, pp. 2628–2635.