Federated Knowledge Purification for Responsive Internet of Things

Irina V. Pustokhina [1,*], Denis A. Pustokhin [2]

[1] Department of Entrepreneurship and Logistics, Plekhanov Russian University of Economics, 117997, Moscow, Russia

[2]Department of Logistics, State University of Management, 109542, Moscow, Russia

Emails: pustohina.IV@rea.ru; da_pustohin@guu.ru

Abstract

The Internet of Things (IoT) has become a ubiquitous technology that enables the collection and analysis of large amounts of data. However, the limited resources of IoT devices pose challenges to enabling responsive decision-making. Many communications are required for network training, yet network updates can be very big if they include many parameters. Participants and the IoT ecosystem both bear the brunt of federated learning's high Latency due to the magnitude of its communications infrastructure requirements. In this paper, we propose a Federated Knowledge Purification (FKP) approach based on dynamic reciprocal knowledge purification and adaptive gradient compression, two strategies that allow for low-latency communication without sacrificing effectiveness, which enables responsive IoT devices with limited resources. The FKP approach leverages a collaborative learning approach to enable IoT devices to learn from each other's experiences while preserving the privacy of their data. A smaller model is trained on the aggregated knowledge of a larger model trained on a centralized server, and this smaller model can be deployed on IoT devices to enable responsive decision-making with limited computational resources. Experimental results demonstrate the effectiveness of the proposed approach in improving the performance of IoT devices while maintaining the privacy of their data. The proposed approach also outperforms existing federated learning methods in terms of communication efficiency and convergence speed.

Keywords: Internet of Things (IoT); Federated Learning; Knowledge Purification; Latency; Communication Overhead

1. Introduction

The Internet of Things (IoT) is a rapidly growing network of interconnected devices, sensors, and appliances that can collect and sharing data. This vast network has the potential to revolutionize industries ranging from healthcare to manufacturing. However, the massive amounts of data generated by these devices present a significant challenge for data processing and analysis. Federated learning (FL) is an emerging approach to machine learning that addresses this challenge by enabling collaborative model training across multiple devices without requiring centralized data storage. In FL, a central server coordinates the training of a machine-learning model by aggregating updates from multiple devices. Each device trains the model locally using its own data, and then sends the updated model parameters to the

Doi: https://doi.org/10.54216/JISIoT.070207
Received: June 19, 2022 Accepted: December 16, 2022

server for aggregation. The server then sends the updated model back to the devices, which repeats the process with new data. This iterative process continues until the model converges [1-3].

FL has several advantages over traditional machine learning (ML) approaches, especially for IoT applications. It enables real-time model training and updates, reduces the need for data transmission, and protects user privacy by keeping data on the device. Additionally, FL can leverage the diversity of data collected from different devices to improve the accuracy and robustness of the trained models. FL has already been successfully applied in several IoT use cases, including predictive maintenance, energy optimization, and smart homes. For example, in predictive maintenance, FL can be used to train machine learning models on data collected from multiple sensors on a device, enabling early detection of potential failures and reducing downtime. In energy optimization, FL can be used to train models on data collected from multiple smart meters, enabling more accurate prediction of energy demand and improving energy efficiency. In smart homes, FL can be used to train models on data collected from multiple sensors and devices, enabling personalized and context-aware automation. As the number of IoT devices continues to grow, FL is likely to become an increasingly important approach to machine learning. Its ability to enable collaborative and distributed model training without compromising privacy or requiring centralized data storage makes it an attractive solution for many IoT use cases.

While training ML algorithms on decentralized data, users' privacy is somewhat protected because the conveyed network updates encompass much fewer confidential details than the original information. As the server and participants work together in a FL setup, they must constantly update each other on network changes as they are made during the training process. Therefore, if the network is large, the communication cost will be prohibitive. Regrettably, recently, deep learning networks have grown increasingly colossal, containing billions of trainable parameters. This is especially impractical for distributed participants with low communication bandwidth and throughput because of the high overhead that can result from transmitting such large networks. Recent times have seen intensive studies into how to maximize the effectiveness of FL communication [3]. As an example, gradient compression can be used to directly shrink network updates, making them more manageable. When the compression ratio needs to be exceptionally high, nevertheless, they typically lose a lot of performance. Additionally, because of the small network capacity, compacting the global network's updated information may also diminish the network’s ability to handle the diversity of decentralized data. For FL to function efficiently in terms of communication, another popular methodology is distillation. If the local network is bigger than the public open dataset, this strategy may lower the communication overhead by only transmitting the local network's forecasts on the sharable set of data. Information is extremely privacy-sensitive and might not be capable of being shared or transferred in many practical systems, including individualized recommendations and the comprehension of electronic health records. Thus, FL which is efficient in terms of connectivity, latency, and data accessibility while maintaining confidentiality presents a significant but as-yet unanswered dilemma [4].

In this work, we present a cross-silo FL system that relies on the distillation of knowledge (KD) to reduce the burden on network communications without sacrificing learning effectiveness, wherein participants have access to more powerful computations as well as big volumes of privately kept training samples than individual appliances. Specifically, a small network (student) and a large network (teacher) learn and share knowledge together, whereas the student network is sharable among multiple participants and learned cooperatively, which could also dramatically lower the communication overhead between the participants and server. We offer an intelligent reciprocal purification (RP) technique to empower the local teacher and student networks to get finetuned by distilling knowledge from their forecast delicate labels and intermediary outcomes, in which, the purification severity is dynamically managed by the rightness of their forecasts. Finally, we implement a singular value decomposition (SVD)-based vibrant gradient interpolating technique to compact the transmitted updates with interactive exactness, which could also fulfill a viable tradeoff between performance and network latency by reducing the communication overhead associated with communicating the student network's updated information.

Doi: https://doi.org/10.54216/JISIoT.070207
Received: June 19, 2022 Accepted: December 16, 2022

2. Methodological Solution

Herein, we lay out the specifics of our knowledge-purification -based FL methodology, which allows for more effective communication between participants. Starting with a definition of the problem at hand, we proceed to introduce our method in detail before wrapping up with some debate on the computing and network complexity of our method [6-8].

A. Problem definition

Our method is based on the supposition that N participants each have their own personal information stored locally, with the source data never leaving the participant in which it is kept. The $i-th$ participant's dataset is referred to as D_i. In our method, each participant stores a replicate of a relatively small, shared student network S with parameters Θ_i^s and a huge, local Teacher network T_i with parameters Θ_i^t. Additionally, these participants are coordinated by a central server to facilitate collaborative network learning. The target is to learn a robust network while maintaining individual privacy and reducing overall communication overhead, thereby the latency of the IoT solution.

B. Federated knowledge purification

This subsection proceeds to describe our federated framework for distilling knowledge. Participants use a dynamic RP mechanism to learn from each other in a two-way process by continuously computing the update to their local teacher and student under the guidance of their locally labeled data. To be more specific, the teacher is upgraded on a per-participant basis while the student network is communicated and learned by all participants. Local teacher networks are more complex than student networks, so the student network can benefit from the instructive information encoded in the teacher network. Since the student network has access to data for all participants and the teacher network only has access to local data, the Teacher network could indeed receive support from the student's accumulated understanding.

Both the local teacher network and the global student network are trained using data from the local area. Both networks are taught using each other's predictions and hidden data from the surrounding area. Before being sent to the server, local gradients are decayed; once there, they are recreated and aggregated. After compiling global gradients, we divide them up and send them to our participants so they can use them to make local adjustments [9-10].

The proposed FL framework introduces three objective functions to train the student and Teacher networks on the local IoT devices. This includes a dynamic reciprocal (DC) cost function to transmit experience from yield smooth labels, a dynamic hidden (DH) cost function to purify the latent representation. These functions are expressed as follows:

$$\mathcal{L}_{t,i}^{t} = \mathrm{CE}(\mathrm{y}_i, \mathrm{y}_i^t) = -\sum_i \mathrm{y}_i\, \mathrm{a}_i \log\,(\mathrm{y}_i^t), \tag{1}$$

$$\mathcal{L}_{s,i}^{t} = \mathrm{CE}(\mathrm{y}_i, \mathrm{y}_i^s) = -\sum_i \mathrm{y}_i \log\,(\mathrm{y}_i^s), \tag{2}$$

The DC losses for both Teacher and student networks (denoted as $\mathcal{L}_{t,i}^{d}$and $\mathcal{L}_{s,i}^{d}$) are formulated as follows:

$$\mathcal{L}_{t,i}^{d} = \frac{\mathrm{JL}(\mathrm{y}_i^s, \mathrm{y}_i^t)}{\mathcal{L}_{t,i}^{t} + \mathcal{L}_{s,i}^{t}}, \tag{3}$$

$$\mathcal{L}_{s,i}^{d} = \frac{\mathrm{JL}(\mathrm{y}_i^t, \mathrm{y}_i^s)}{\mathcal{L}_{t,i}^{t} + \mathcal{L}_{s,i}^{t}}, \tag{4}$$

Whereas JL denotes the Jensen-Shannon loss, which is computed as follow calculated as follows

Doi: https://doi.org/10.54216/JISIoT.070207
Received: June 19, 2022 Accepted: December 16, 2022

$$\mathrm{JL}\,[P,Q] = \frac{1}{2}(\mathrm{KL}\,[P \left\| \frac{P+Q}{2} \right] + \mathrm{KL}\left[Q \left\| \frac{P+Q}{2} \right]\right) \tag{5}$$

where KL signifies the Kullback–Leibler divergence, such that., $\mathrm{KL}(\mathbf{P}, Q) = -\sum_i \mathbf{P}_i \log\left(\frac{Q_i}{P_i}\right)$. In this manner, the purification concentration is low if the projections of the Teacher and the student are not dependable, which means that the subjective cost function of them are significant. The DH objective for both networks (denoted as $\mathcal{L}_{t,i}^{h}$ and $\mathcal{L}_{s,i}^{h}$) are expressed as:

$$\mathcal{L}_{t,i}^{h} = \mathcal{L}_{s,i}^{h} = \frac{\mathrm{HL}(\mathrm{H}_i^t, \mathrm{W}_i^h \mathrm{H}^s) + \mathrm{HL}(\mathrm{A}_i^t, \mathrm{A}^s)}{\mathcal{L}_{t,i}^{t} + \mathcal{L}_{s,i}^{t}}, \tag{6}$$

whereas HL denotes Huber Loss, which is defined as follows:

$$L_\delta = \begin{cases} \frac{1}{2}(P_i - Q_i)^2, if\ \left|\tilde{Y}_i - Y_i\right| \le \delta \\ \delta|P_i - Q_i| - \frac{1}{2}\delta^2, otherwise \end{cases} \tag{7}$$

Which, the H_i^t, A_i^t, H^s, and A^s correspondingly signify the latent representations and attentive maps in the $i-th$ local Teacher and the student, and $\mathbf{W}_i^h$ denote trainable linear revolution matrix. In this paper, we offer a method for controlling the magnitude of the DH loss depending on the extent to which both the student and the Teacher are accurate in their predictions. In addition to this, we additionally train the student formula based on the task-specific labels that are associated with each participant. This is prompted by the task-specific earlier purification architecture. As a result, the final objective necessary to calculate the parameters of client's Teacher and student networks are implemented on each participant. Thus, the final losses, $\mathcal{L}_{t,i}$and $\mathcal{L}_{t,i}$ are devised as follows:

$$\mathcal{L}_{t,i} = \mathcal{L}_{t,i}^{d} + \mathcal{L}_{t,i}^{h} + \mathcal{L}_{t,i}^{t}, \tag{8}$$

$$\mathcal{L}_{s,i} = \mathcal{L}_{s,i}^{d} + \mathcal{L}_{s,i}^{h} + \mathcal{L}_{s,i}^{t}, \tag{9}$$

The gradients $\boldsymbol{g}_i$ **of** student network on the $i-th$ participant could be calculated according to $\mathcal{L}_{s,i}$, thru $\mathbf{g}_i = \frac{\partial \mathcal{L}_{s,i}}{\partial \Theta^s}$, wherever Θ^s denote the parameter set of the student network. Each participant's local gradients, which are derived from the loss function, immediately cause an update to be made to the participant's local Teacher network). In addition to this, the server sends the averaged global gradients to each participant so that they can be updated locally. In order to bring its replica of the student network up to date, the participant must first decrypt the global gradients. This procedure would be carried out again and again until both the student network and the Teacher network converge. Take note that the Teacher network is employed for label prediction while the test phase is being performed. For the benefit of the reader, we have provided a comprehensive overview of the suggested network's operation (Algorithm 1).

Algorithm 1: Pseudocode of the proposed network for federated purification in IoT.

1:	define the learning rate of teacher as $\boldsymbol{\eta_t}$, and for students as η_s,
2:	Deciding the participant number *N*, *and* hyperparameters $\boldsymbol{T_{start}}$ and $\boldsymbol{T_{end}}$
3:	Loop on all participants concurrently do
4:	set initial weights $\boldsymbol{\Theta_i^t}$, $\boldsymbol{\Theta^s}$
5:	reiterate
6:	$\boldsymbol{g_i^t}$, $\boldsymbol{g_i}$= ClientUpdate (*i*)
7:	$\boldsymbol{\Theta_i^t \leftarrow \Theta_i^t - \eta_t g_i^t}$
8:	$\boldsymbol{g_i \leftarrow U_i \sum_i V_i}$
9:	Participants encrypt $\boldsymbol{U_i \sum_i V_i}$
10:	Participants upload $\boldsymbol{U_i \sum_i V_i}$to the aggregator
11:	Aggregator decrypts $\boldsymbol{U_i \sum_i V_i}$

Doi: https://doi.org/10.54216/JISIoT.070207
Received: June 19, 2022 Accepted: December 16, 2022

12:	Aggregator rebuilds g_i
13:	Global gradients g ← 0
14:	Loop on all participants concurrently do
15:	g = g + $\boldsymbol{g_i}$
16:	Terminate Loop
17:	
18:	Aggregator encodes $\boldsymbol{U}$, ∑, $\boldsymbol{V}$
19:	Aggregator broadcast $\boldsymbol{U}$, ∑, $\boldsymbol{V}$ to user participants
20:	Participants decode U, ∑, V
21:	Participants reconstruct g
22:	$\boldsymbol{\theta^s} \leftarrow \boldsymbol{\theta^s} - \frac{\eta_s g}{N}$
23:	till Local networks converge
24:	Terminate Loop
25:	ClientUpdate(*i*):
26:	Calculate $\mathcal{L}_{t,i}^{t}$ and $\mathcal{L}_{s,i}^{t}$
27:	Calculate $\mathcal{L}_{t,i}^{d}$, $\mathcal{L}_{s,i}^{d}$, $\mathcal{L}_{t,i}^{h}$ and $\mathcal{L}_{s,i}^{h}$
28:	$\mathcal{L}_{i}^{t} \leftarrow \mathcal{L}_{t,i}^{t} + \mathcal{L}_{t,i}^{d} + \mathcal{L}_{t,i}^{h}$
29:	$\mathcal{L}_{i}^{s} \leftarrow \mathcal{L}_{s,i}^{t} + \mathcal{L}_{s,i}^{d} + \mathcal{L}_{s,i}^{h}$
30:	Calculate local Teacher gradients $\mathbf{g}_i^t$ from $\mathcal{L}_{i}^{t}$
31:	Calculate local student gradients $\boldsymbol{g_i}$ from $\mathcal{L}_{i}^{s}$
32:	return $\mathbf{g}_i^t, \mathbf{g_i}$

3. Experimentation and Analysis

A. Simulation Setup

We test two different scenarios where user input is required. Individual news suggestion is the first assignment, where the MIND dataset is used in this exercise [14]. The second is ADR text presence recognition, as a binary classification problem. We randomly split the training data for these two datasets into four folds, with the assumption that each fold is privately kept by a separate participant, to network a situation in which private data is decentralised across several users. To assess how well various FL techniques deal with non-IID data, we assume that each of these datasets is maintained by a participant. The detailed statistics of the abovementioned datasets is given in Table 1.

Table 1: The statistics of the datasets adopted in our simulations.

Datasets	Attribute	Value
MIND	Users	1000000
	News	161013
	Impressions	15777377
	Clicks	24155470
	Training set	2186683
	Validation	365200
	Test set	2341619
ADR	# Positive samples	1355
	# Negative samples	15336
	Average text length	16.48

Doi: https://doi.org/10.54216/JISIoT.070201
Received: June 19, 2022 Accepted: December 16, 2022

Multiple common metrics are adopted to evaluate the proposed framework, including area under the curve (AUC), Mean reciprocal rank (MRR), Discounted cumulative gain (i.e., nDCG@K) in case of MIND, and precision, recall, and F1-score in case of ADR. In mathematical terms, the above metrics can be defined as follows:

$$Precision\ (P) = \frac{TP}{TP + FP} \times 100, \tag{10}$$

$$Recall\ (R) = \frac{TP}{TP + FN} \times 100, \tag{11}$$

$$F1 - score\ (F1) = 2 * \frac{P * R}{P + R}, \tag{12}$$

$$\mathrm{AUC} = \frac{\sum_{p \in \mathcal{P}} \sum_{n \in \mathcal{N}} I[P(p) > P(n)]}{|\mathcal{P} \parallel \mathcal{N}|} \tag{13}$$

$$\mathrm{MRR} = \frac{1}{N_p} \sum_{i=1}^{N_p} \frac{1}{\mathrm{Rank}\ (p_i)} \tag{14}$$

$$\mathrm{nDCG@\ K} = \frac{\sum_{i=1}^{K} (2^{2_1} - 1) \frac{1}{\log_2(1 + i)}}{\sum_{i=1}^{N_p} \frac{1}{\log_2(1 + i)}} \tag{15}$$

FN and FP refer to False Negative and False Positive, while TP and TN signify the True Positive, and True Negative samples. $\mathcal{P}$ and $\mathcal{N}$ individually represent the groups of positive and negative examples. $I[\ \cdot\ ]$ denote the marker function. r_i represent the significance score of news having $i - th$ order, and t has a value of one in the case of snapped news and zero for non-snapped news.

The Shufflenet v2 [17] network is used as the local Teacher in our studies on every participant. In our investigations, we use a pre-trained language network's purification as a case study. Sub-networks of its first 4 or 2 Shufflenet layers are used as student networks. We use a language network for text categorization on the ADR dataset, and then we apply a dense layer. The starting and ending energy ratios are 0.90 and 0.97, correspondingly. The whole range of hyper-parameters is detailed in Table 2. Each experiment is performed 5 times to reduce the possibility of random error.

Table 2: The best hyperparameters for the proposed framework

Hyperparameters	**Value**
Attention query dimension	100
Batch size	32
Dropout rate	0.2
Epochs	15
Student network learning rate	0.0005
Teacher network learning rate	0.00001
Network hidden dimension	768
Negative sampling ratio	4
Optimization algorithm	AdamW
Oversampling ratio	2

Doi: https://doi.org/10.54216/JISIoT.070207
Received: June 19, 2022 Accepted: December 16, 2022

B. Simulation Experiments

The Ground edition of Shufflenet v2 is installed as the local Teacher method on each participant in our experimental tests. The sub-networks of the early four shufflenet layers are used as student networks. We evaluate the proposed network against the state-of-the-art methods of federated purification in massive IoT applications. These methods include FD [18], FEDKD [19], DFKD [20], SKD [16], and Deepobfuscation [15]. On the MIND and ADR benchmark data, we detail the outcomes of our networks and the corresponding costs of communication in Table 3 and Table 4, respectively. It turns out that the locally hosted samples on a single participant might not be enough to learn a robust network, which is why Shufflenet v2 (Local) performs poorly contrasted with the baselines that learn in distributed fashion. While the results of Shufflenet v2 (Fed) are comparable to those of centralized learning, the high communication cost for network learning may limit its use in practical settings. Compression techniques that use either networks or gradients to shrink messages have been shown to be effective at lowering the price of communication, but at the expense of either substantial performance loss or only a marginal reduction in the amount of data transmitted. Unlike these other methods, the proposed network can even compete with the results of learning a large network using centralized data.

Table 3: Comparison between the results of the proposed network against competing methods on the MIND dataset.

Methods	AUC	MRR	nDCG@5	nDCG@10	Communication cost (GB)
FD [18]	70.44±0.12	34.32±0.2	37.92±0.3	47.9±0.1	1.28±0.23
FEDKD [19]	71.53±0.24	36.12±0.03	38.96±1.23	44.73±0.05	2.77±0.22
DFKD [20]	70.59±0.05	35.67±0.27	39.12±1.28	44.58±0.16	0.99±0.2
SKD [16]	70.08±0.25	34.32±0.07	38.01±2.36	46.51±0.29	1.47±0.16
Deepobfuscation [15]	71.01±0.2	35.3±0.2	38.93±1.13	45.71±0.21	0.27±0.17
Proposed	75.13±0.16	39.46±0.11	73.87±0.19	49.53±0.03	0.18±0.1

Remarkably, the achieved improvements are demonstrated as not statistically significant ($p > 0.05$) according to a two-tailed t-test. This is because in the proposed network, multiple Teacher networks exist on separate decentralized participants, allowing for individualized instruction and the evaporation of irrelevant material. Further, in comparison to other FL -based methods, the proposed network is more communication-efficient, saving up to 96.7% and 97.9% of communication costs for MIND and ADR, respectively. This is because the proposed network is able to gain insight from more complex local teacher networks, thereby enhancing the network's performance, while also cutting down on communication costs through the sharing of updates from a more modest student network. The findings demonstrate that the proposed network can significantly cut down on the time spent communicating between nodes in a FL setup while maintaining promising network performance.

Table 4. Comparison between the results of the proposed network against competing methods on ADR dataset.

Methods	Precision	Recall	F1-score	Communication cost (GB)
FD [18]	60.41±0.05	60.56±0.05	54.73±0.18	0.79±57.41
FEDKD [19]	58.88±0.11	62.78±0.19	60.77±0.14	2.66±58.88
DFKD [20]	58.22±0.3	61.6±0.12	59.86±0.17	0.78±58.22
SKD [16]	56.8±0.12	59.83±0.14	58.28±0.13	0.42±56.8
Deepobfuscation [15]	59.53±0.23	62.27±0.07	60.87±0.11	0.09±59.53
Proposed	64.08±0.13	67.31±0.29	65.66±0.18	0.08±58

Doi: https://doi.org/10.54216/JISIoT.070207
Received: June 19, 2022 Accepted: December 16, 2022

Following this, we use the proposed network to test our adaptive RP method. We begin by contrasting the efficacy of Teacher and student strategies in This suggested network can be trained with RP or with no RP (See Figure. 1). We find that the performance of both the Teacher and student networks, regardless of their size, may be significantly improved through RP. The Teacher networks have more complex structures, and the student can learn from the information embedded in them. Since Teacher networks are learned using only locally available labelled data, a student's beneficial knowledge encoded by the student might provide valuable context for the Teacher to overcome this data scarcity. Based on our findings that local Teachers perform marginally better than their students, we decide to infer using Teacher networks during testing.

Moreover, we contrast Variants of the suggested network are created by omitting either the RP objective, the DH objective, or the DW technique (See Figure 2). Please take into account that we document the results of teacher networks. When applied to the network, both adaptive RP and adaptive hidden losses show promise for enhancing its overall performance. When the DW technique is not used, performance also suffers. This is due to the fact that taking into account the accuracy of network predictions during purification might lead to better knowledge extraction and reduce the likelihood of overfitting.

Doi: https://doi.org/10.54216/JISIoT.070207
Received: June 19, 2022 Accepted: December 16, 2022

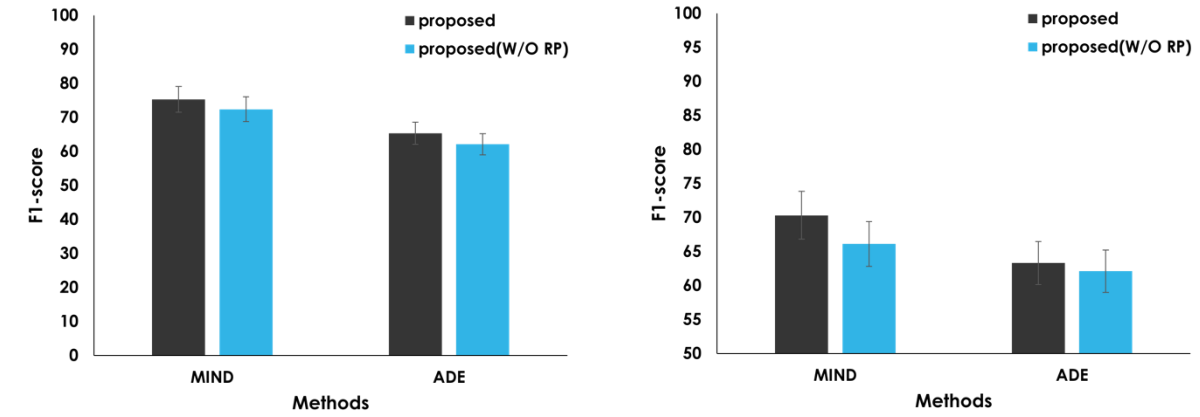

Figure 1: Illustration of Impact of mutual distillation on model performance on the student (left) and Teacher models(right).

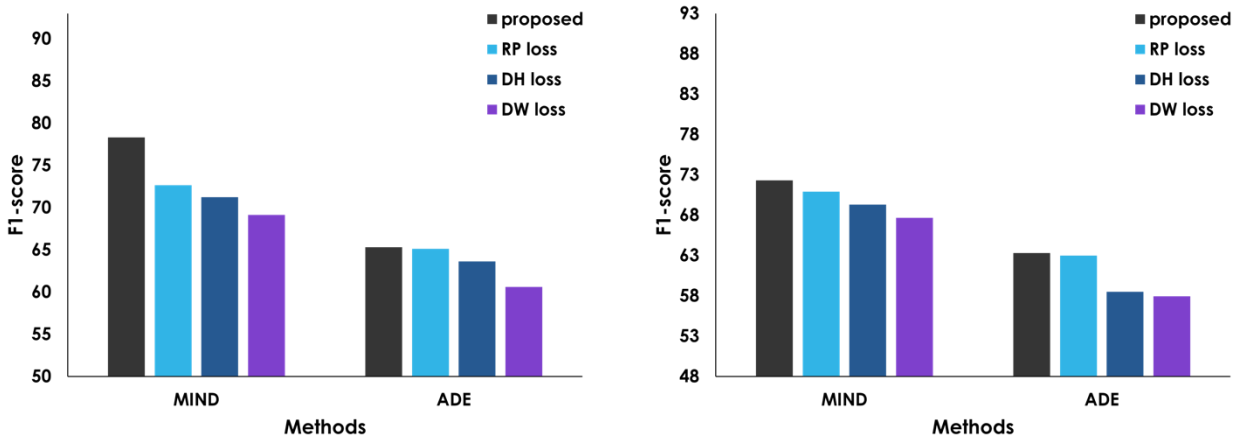

Figure 2: Efficacy of the adaptive MD methods in the proposed model with two layers (left) and four-layer settings(right).

4. Conclusion

In conclusion, the Federated Knowledge purification (FKP) approach proposed in this paper is a promising solution to enable responsive Internet of Things (IoT) devices with limited resources. By leveraging a collaborative learning approach, the FKP framework enables IoT devices to learn from each other's experiences and improve their performance, while preserving the privacy of their data. The proposed approach utilizes a smaller model that is trained on the aggregated knowledge of the larger model, which is trained on a centralized server. The smaller model can then be deployed on IoT devices to enable responsive decision-making with limited computational resources. Experimental results demonstrate the effectiveness of the proposed FKP approach in improving the performance of IoT devices while maintaining the privacy of their data. The proposed approach also outperforms existing FL methods in terms of communication efficiency and convergence speed.

References

[1]. Yang, C., Xie, L., Qiao, S., & Yuille, A. (2018). Knowledge distillation in generations: More tolerant teachers educate better students. *arXiv preprint arXiv:1805.05551*.

Doi: https://doi.org/10.54216/JISIoT.070207
Received: June 19, 2022 Accepted: December 16, 2022

[2]. Zhu, X., & Gong, S. (2018). Knowledge distillation by on-the-fly native ensemble. *Advances in neural information processing systems*, *31*.
[3]. Sau, B. B., & Balasubramanian, V. N. (2016). Deep model compression: Distilling knowledge from noisy teachers. arXiv preprint arXiv:1610.09650.
[4]. Song, X., Feng, F., Han, X., Yang, X., Liu, W., & Nie, L. (2018, June). Neural compatibility modeling with attentive knowledge distillation. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (pp. 5-14).
[5]. Yu, R., Li, A., Morariu, V. I., & Davis, L. S. (2017). Visual relationship detection with internal and external linguistic knowledge distillation. In Proceedings of the IEEE international conference on computer vision (pp. 1974-1982).
[6]. Seo, H., Park, J., Oh, S., Bennis, M., & Kim, S. L. (2020). Federated knowledge distillation. *arXiv preprint arXiv:2011.02367*.
[7]. Lee, S. H., Kim, D. H., & Song, B. C. (2018). Self-supervised knowledge distillation using singular value decomposition. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 335-350).
[8]. Wu, C., Wu, F., Lyu, L., Huang, Y., & Xie, X. (2022). Communication-efficient federated learning via knowledge distillation. *Nature communications*, *13*(1), 1-8.
[9]. Liu, X., Wang, X., & Matwin, S. (2018, November). Improving the interpretability of deep neural networks with knowledge distillation. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)* (pp. 905-912). IEEE.
[10]. Lu, L., Guo, M., & Renals, S. (2017, March). Knowledge distillation for small-footprint highway networks. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4820-4824). IEEE.
[11]. Asami, T., Masumura, R., Yamaguchi, Y., Masataki, H., & Aono, Y. (2017, March). Domain adaptation of dnn acoustic models using knowledge distillation. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5185-5189). IEEE.
[12]. Xu, Z., Hsu, Y. C., & Huang, J. (2017). Training shallow and thin networks for acceleration via knowledge distillation with conditional adversarial networks. arXiv preprint arXiv:1709.00513.
[13]. Wang, T., Zhu, J. Y., Torralba, A., & Efros, A. A. (2018). Dataset distillation. *arXiv preprint arXiv:1811.10959*.
[14]. Hou, S., Pan, X., Loy, C. C., Wang, Z., & Lin, D. (2018). Lifelong learning via progressive distillation and retrospection. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 437-452).
[15]. Xu, H., Su, Y., Zhao, Z., Zhou, Y., Lyu, M. R., & King, I. (2018). Deepobfuscation: Securing the structure of convolutional neural networks via knowledge distillation. arXiv preprint arXiv:1806.10313.
[16]. Ge, S., Zhao, S., Li, C., & Li, J. (2018). Low-resolution face recognition in the wild via selective knowledge distillation. IEEE Transactions on Image Processing, 28(4), 2051-2062.
[17]. Ma, N., Zhang, X., Zheng, H. T., & Sun, J. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European conference on computer vision (ECCV) (pp. 116-131).
[18]. Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., & Kim, S. L. (2018). Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. arXiv preprint arXiv:1811.11479.
[19]. Yurochkin, Mikhail, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. "Probabilistic federated neural matching." (2018).
[20]. Lopes, R. G., Fenu, S., & Starner, T. (2017). Data-free knowledge distillation for deep neural networks. arXiv preprint arXiv:1710.07535.

Doi: https://doi.org/10.54216/JISIoT.070207
Received: June 19, 2022 Accepted: December 16, 2022