Towards Robust Federated Learning using Knowledge Distillation Techniques

Arindam Jain
Arizona State University
ajain243@asu.edu

Kiran Sthanusubramonian
Arizona State University
ksthanus@asu.edu

***Abstract*—Federated Learning paradigms enable many edge-client devices to jointly share a single globalized server model while maintaining users' data privacy. Instead of divulging sensitive data, distributed learning frameworks primarily rely on model parameters sent between learners. However, due to the heterogeneity of client-level data, the requirement for personalization for each participating device becomes apparent. In this paper, we explore personalized Deep Neural Network (DNN) models specific to each client device, to solve image classification tasks on the local clients. We further develop the globalized server objective of the federated architecture and proposed novel approach using Knowledge Distillation (KD) to train the global model more efficiently. As a novelty, we also attempt to establish the robustness of our algorithm to common training-time adversarial attacks.**

***Index Terms*—Federated Learning, Knowledge Distillation, Privacy, Robustness, Adversarial Attacks, Fairness**

I. Introduction and Problem Statement

With improved privacy standards, edge computing capabilities, and large-scale machine learning requirements, Federated Learning has emerged as a privacy-preserving training paradigm for localized devices without data-sharing and aggregation requirements. With privacy-preserving benefits, users of these localized devices (e.g., mobile phones) also benefit from lower latencies in terms of required responses. Potential Federated Learning applications include improved mobile computing [1], healthcare [2], and autonomous vehicles.

Large deep learning models are commonly trained on a single compute cluster by combining data from several scattered sources. However, the slow increase of data resources severely limits the size and quality of data sets in many applications and the flow of data and processing capability. In order to protect user privacy and personal information, typical federated learning systems involve the central server defining a globalized model before sending it to all participating edge-client devices. Users with distributed terminals train the initial model using their private data once the central server distributes it. The local weight updates are transmitted back to the central server via the distributed terminals for reaggregation.

A common drawback of having a single standardized global model is the heterogeneity of training data among the client devices. Data heterogeneity can include the presence of non-Independent and Identically Distributed (i.i.d) datasets and clients with varying dataset sizes. With more participating client devices, the heterogeneity of training data increases, and as a result, the overall performance of the global model dips for local client-specific tasks in exchange for increased generality. Another drawback is the lack of robustness of federated systems, as participating clients can be affected by training-level adversarial attacks or lose connection from the global network at any moment.

In this project, we create an image classification experiment that utilizes personalized Deep Neural Network (DNN) models for each participating client and a separate DNN for representing the server (global) model. For efficiently training the global model, we utilize a Multi-teacher-based ensemble Knowledge Distillation paradigm to achieve faster convergence rates. We employ personalized models to counter the data heterogeneity issue and improve fairness among the clients. We further investigate the robustness of our approach to dealing with training-level adversarial attacks. To the best of our knowledge, this is the first report which conducts such an investigation on Knowledge Distillation-based Federated Learning techniques.

The rest of this report is structured as follows: Section II presents a literature survey that discusses pertinent related works, shortcomings, and the background material used for project execution. Section III outlines the main contributions of our work and the challenges faced during the execution of this project. Section IV discusses the core methodology and the stages of execution followed during the project. Section V comprehensively defines the experimental setups and results obtained, followed by conclusions and future work in Section VI.

II. Related Work and Shortcomings

A. Knowledge Distillation Techniques for Federated Learning

FedMD (Li and Wang) [4] defines one of the popular KD Algorithms for Heterogenous Federated Learning. Here, the primary assumption is that the client devices have the computation capability of defining and training personalized NN models specific to their local datasets - this helps solve the statistical heterogeneity of data across the devices. FedMD focuses on utilizing a core concept of KD, i.e., using the Prediction Logits output from local Deep Neural Network models averaged out as Knowledge to reach a better global consensus. FedMD also employs Knowledge Transfer (using a single large global dataset) to overcome the issue of small local datasets.

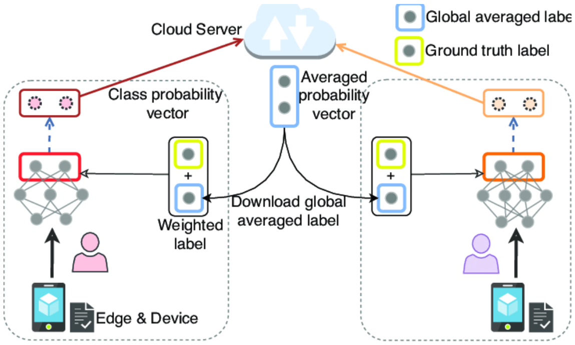

Fig. 1. Working of the FedMD Algorithm [4], [8]

However, this paradigm is prone to potential corruption of the global consensus due to adversarial attacks on one or more of the clients, which can become more pronounced in KD scenarios. This paradigm largely depends on using a single large global dataset, with no particular server model employed. We overcome this limitation by using only each participating client's private data and computation capabilities to train a server model using KD accurately.

More related work to Federated Learning using Knowledge Distillation is discussed in [5], where several Client (Teacher) models are fused into a single Server (Student) model, which is further trained using unlabeled datasets to improve robustness. However, this still does not defend against potential adversarial agents (a similar drawback to the algorithm presented in FedMD).

B. *Improving Fairness and Robustness of Federated Learning Architectures*

To improve the robustness of Federated Learning architectures against potential adversarial attacks, we take inspiration from the experiments conducted in Ditto (Tian Li et al.) [6]. This paper introduces the notion that the constraints of client-level fairness can directly compete with the global server model's robustness and the Federated setting's overall objective. To solve these competing constraints, Ditto proposes a version of model personalization to improve fairness among clients and the robustness of the overall architecture to training-time attacks. Fairness and robustness are balanced by regularizing the local model updates when the global consensus is calculated. The regularized parameter can be tuned per several factors, such as local dataset sample size and the number of devices affected by adversarial attacks. To show the effectiveness of the Ditto solver, the authors also introduce common training-time attacks, such as label poisoning and irregular model updates, to their experimentation.

We found the experimentation methodology of this work to be very interesting. However, we believe that directly creating customized models per each client device based on its available computational capability will inherently yield better performance. We attempt to use the Ditto solver to model, evaluate and improve the robustness of our solution to training-time adversarial attacks.

III. Main Contributions and Challenges

A. *Main Contributions*

1) The primary contribution of this project is to establish the robustness and fairness of Knowledge Distillation-based Federated Learning paradigms. In our project, we attempt to combine the algorithm and experimentation ideas covered in our two primary literature sources; i.e., FedMD (Li and Wang) [4] and Ditto (Tian Li et al.) [6].
2) We worked on a KD-based Federated Learning paradigm by introducing an explicit server model (behaves as the student) trained by a selected number of clients (behaves as the teachers) present at each communication round. Each client has a personalized DNN model to classify privately labeled datasets. The FedMD algorithm requires a large public dataset to implement its knowledge-transfer-based Model Distillation. With our proposed paradigm, we overcome the limitation of using a public dataset.
3) We built the training-level adversarial attack experiments conducted in Ditto to evaluate the effective fairness and robustness of our proposed paradigm. We find that our paradigm can be adapted to balance fairness and robustness constraints calculated while evaluating the global consensus using the ideas presented in the Ditto solver.

B. *General Challenges in Federated Learning*

There are several obstacles faced in the implementation of Federated Learning:

1) Data and device heterogeneity are one of the issues of federated learning; this leads to problems when a user has rich data but cannot modify the overall server model to benefit from potential personalization.
2) When creating techniques for federated learning networks, Communication Efficiency is a crucial metric to be considered. This is because federated networks may comprise a sizable number of devices, and communication throughout the network may be much slower than local computing.
3) Robustness to client-level corruption is another crucial metric not usually considered in the design of Federated systems. Clients can face a host of issues, such as mislabelled datasets, label-poisoning attacks, or dropping out of the system at any given time.
4) In federated learning, privacy concerns drive the requirement to keep raw data local to each device. We observed that recent techniques use differential privacy or secure multiparty computing to increase the privacy of federated learning. However, the federated system's effectiveness is typically sacrificed to provide privacy. Therefore, realizing private federated learning systems presents a significant difficulty reconciling these trade-offs.

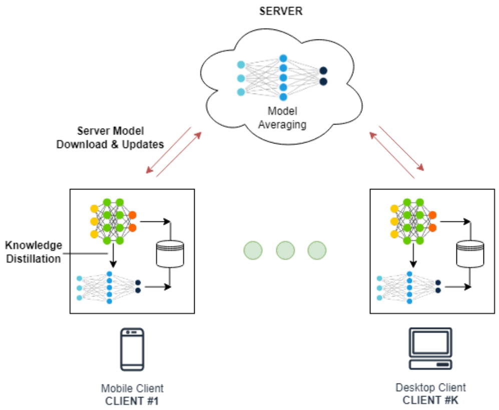

Fig. 2. Federated Distillation Algorithm - Overall Workflow

C. *Challenges faced during Project Execution*

1) We faced challenges in implementing the FedMD algorithm for the first time. This came from a combination of the poor documentation for the algorithm's working and relative inexperience in implementing Federated Learning experiments. In contrast, we would like to remark on the thorough documentation of the Ditto codebase, which was very helpful. However, due to its complex code layering, it took time to understand and use the Ditto codebase thoroughly.
2) The biggest challenge we faced was adapting the FedMD algorithm to implement the Ditto-level experimentations. We initially believed the FedMD algorithm could be used directly; however, we had to change this up and implement a Multi-teacher (Clients), single-student (Server) Knowledge Distillation architecture from scratch. This phase highlights a challenging but very productive learning experience for us during the execution of this project.
3) Due to the lack of computing (GPU) resources, it was difficult for us to extend our experiments to more complex datasets such as CIFAR100. We primarily deployed our models as Python notebooks in Google CoLab and Kaggle sites, but these sites had usage time limitations.

IV. Methodology and Proposed Solution

As per our project proposal, our primary idea was to combine the FedMD algorithm with the experimentation ideas seen in Ditto to evaluate the robustness and fairness of KD-based Federated Learning paradigms. However, we had to redesign our core Federated Learning paradigm to a more classical Multi-teacher (Client models), single-student (Server model) KD setting to incorporate the Ditto solver effectively. We provide more details as follows:

A. *Core Solution*

CNN Model at the Server is downloaded and trained using Knowledge Distillation by each personalized CNN Model on the Clients. The role of the personalized models is to improve fairness for each of their clients. In contrast, the server model effectively summarizes the classification tasks conducted by each delegated client. We choose Knowledge Distillation as a means to speed up the convergence of the training process for the server model. The overall solution is presented as Algorithm 1.

Algorithm 1 Federated Distillation Algorithm

```
Input: Server Model $S$, Personalized Client Models $C_k$,
  Private Datasets $D_k$, ($k$ = 1...$m$),
  $T$ = Distillation Temperature, $\lambda$ = Regularizing Parameter
  for $n = 1...N$ (# of communication rounds) do
      for $k = 1...m$ do
          Train for a few epochs:
          $\omega_k^{t+1} = \omega_k^t + \eta\nabla L(\omega_k^t, b, D_k)$        ▷ Train $C_k$
          $\omega_{s,k}^{t+1}$ = Knowledge-Distillation($\omega_s^t, \omega_k^{t+1}, T, \lambda, D_k$)
      end for
Aggregate: $\omega_s^{t+1} = \frac{1}{m}\sum_{k=1}^{m} \omega_{s,k}^{t+1}$        ▷ Server-side Model
  Aggregation
  end for
  return $S$, $C_k$ ($k = 1...m$)
```

The personalized client models do not train on a large public dataset; instead, they train only on their private data while simultaneously teaching the downloaded server model at each communication round. As part of our experiments, we ensure that each private dataset does not overlap. For example, if we test our solution on a pre-determined dataset (say, MNIST) with 10-digit classes, we evenly spread a few 100 samples of a maximum of three classes to each client as their private data. The overall testing accuracy of the server model is finally tested on the complete testing sampling of the pre-determined dataset.

B. *Training-time Adversarial Attacks*

To evaluate the robustness of our proposed solution, we simulate two types of training attacks, which we have taken from Ditto [6]. Each of these should drastically affect the server model's convergence toward the global objective. Overall, we experiment with two percentages of corrupted/malicious devices: 20% and 30%.

1) Label Poisoning: We select a pre-defined percentage of corrupted client devices with poisoned training sample labels using label-flipping or random noisy labels.

2) Random Model Updates: Malicious devices send back random server model updates as zero-mean Gaussian parameters.

V. Experimental Setup and Results

In this study, federated learning employing knowledge distillation and customization is assessed on various tasks, models, and data sets to confirm the efficacy of the method. A merged data set is analyzed to more properly handle statistical heterogeneity in order to better represent data heterogeneity and its influence on convergence. The non-independent-identical-distribution data properties of federated learning are

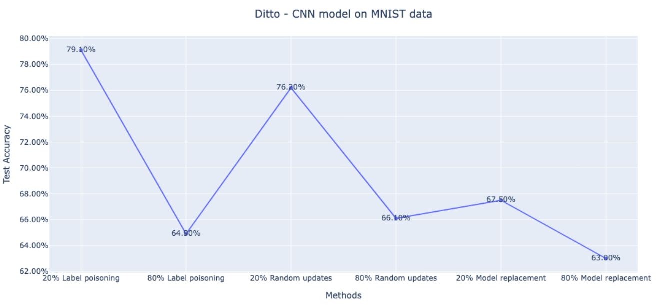

Fig. 3. Ditto: Fair and Robust Federated Learning Through Personalization [6]

completely taken into account by distributing data to many devices. Five edge device devices are chosen, with ten classification jobs used as an example, and only two types of data samples are included in each node. The true public data set and the artificial data set are described below.

A. Datasets

We strictly define our problem space to Image Classification tasks. Implementing the FedMD algorithm requires a general public dataset and local private datasets. In the first instance, a subset of the FEMNIST serves as the private data while the MNIST serves as the public data. We consider the non-.i.i.d. case and the i.i.d. scenario, where each private dataset is chosen randomly from FEMNIST. In the second instance, the private dataset is a subset of the CIFAR100, which comprises 100 subclasses and 20 superclasses, and the public dataset is the CIFAR10. For example, the big carnivores category includes the bear, leopard, lion, tiger, and wolf. Each node's task in the i.i.d. scenario is to classify the test pictures correctly. It is more difficult to categorize generic test data in the non-i.i.d. scenario since each participant only has data from one subclass per superclass during training. For instance, it is predicted that a person who has only seen wolves throughout training will properly identify lions as huge carnivores. As a result, it must rely on the information shared by other devices

B. Results and Performance Evaluation

Three presumptions are made by us in accordance with the presumptions used by the comparative methodologies. To hasten the model convergence, we first make the assumption that all users were active during the whole training process. Second, between global aggregations, the data for each user remains constant. Last but not least, across the users that participated, the hyper-parameters batch-size (B) and local epochs (E) remain constant. On a 16GB memory NVIDIA Tesla P100, we run all experiments with a 70%-30% train-test split.

Methods	**Accuracy**
FedMD	
Ditto	
FD with robustness and personlization (Novel approach)	

Model	1st conv layer filter (n1)	2nd conv layer filter (n2)	3rd conv layer filter (n3)	dropout rate	pre-trained tes accuracies on
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					

VI. Conclusion and Future Work

The primary objective of this project is to acquire foundational knowledge on Federated Learning, which is a massive breakthrough in deploying large-scale machine learning models in the real world while still preserving user data privacy. In this paper, we implemented to establish a baseline for robust and fair distillation techniques with improved personalization for heterogeneous devices and resilience to common adversarial attacks that can affect the whole system.

The study of distributed machine learning is extremely important. Federated learning is a significant advancement in distributed machine learning in recent years. When compared to the conventional distributed machine learning technique, it gives data privacy protection greater consideration. The Federated learning employing knowledge distillation and customization algorithm is suggested in the study, and it is tested for accuracy and communication costs against the FedMD and Ditto algorithms. According to the experimental findings, the suggested approach may be utilized to help the central node teach additional client devices and can enhance the overall model's capacity for learning. Future work will focus on developing a distributed machine learning technique that is more practical and effective.

In order to enhance the performance of Federated Learning, it handles the problem of statistical heterogeneity between data from many clients. During the FL training phase, it identifies the ideal teacher model for each client, and following the training period, it extracts the pertinent information from ideal instructors into each user's local data. Using data-splitting techniques, we assess the performance of the innovative strategy on the FEMINIST and MNIST datasets. Through experiments, we demonstrate that it beats FedAvg and is able to achieve nearby results as cutting-edge FL approaches, FEDMD and Ditto, with little communication overhead.

References

[1] W. Y. B. Lim et al., ”Federated Learning in Mobile Edge Networks: A Comprehensive Survey,” in IEEE Communications Surveys and Tutorials, vol. 22, no. 3, pp. 2031-2063, third quarter 2020.

[2] Xu, J., Glicksberg, B.S., Su, C. et al. Federated Learning for Healthcare Informatics. J Healthc Inform Res 5, 1–19 (2021).

[3] Gou, J., Yu, B., Maybank, S.J. et al. Knowledge Distillation: A Survey. Int J Comput Vis 129, 1789–1819 (2021).

[4] Li, Daliang and Junpu Wang. “FedMD: Heterogenous Federated Learning via Model Distillation.” ArXiv abs/1910.03581 (2019)

[5] Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. 2020. Ensemble distillation for robust model fusion in federated learning. In Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20). Curran Associates Inc., Red Hook, NY, USA, Article 198, 2351–2363.

[6] Tian Li, Shengyuan Hu, Ahmad Beirami and Virginia Smith. (2021). Ditto: Fair and Robust Federated Learning Through Personalization.

[7] Jiang, Donglin; Shan, Chen; Zhang and Zhihui. (2021). Federated Learning Algorithm Based on Knowledge Distillation.

[8] Wu, Qiong and He, Kaiwen and Chen, Xu. (2020). Personalized Federated Learning for Intelligent IoT Applications: A Cloud-Edge Based Framework. IEEE Computer Graphics and Applications. PP. 1-1. 10.1109/OJCS.2020.2993259.

VII. Appendix

https://github.com/kiran-asu5115/CSE598-ML-Privacy-Course-Project/tree/main/Codebase