Model Optimization Techniques in Personalized Federated Learning: A Survey

Fahad Sabah[a,b], Yuwen Chen[a], Zhen Yang[a,*], Abdul Raheem[a], Muhammad Azam[b], Raheem Sarwar[c]

[a]*Faculty of Information Technology, Beijing University of Technology, Beijing, China*
[b]*Faculty of CS&IT, Superior University, Lahore, Pakistan*
[c]*OTEHM, Faculty of Business and Law, Manchester Metropolitan University, Manchester, United Kingdom*

Abstract

Personalized federated learning (PFL) is an exciting approach that allows machine learning (ML) models to be trained on diverse and decentralized sources of data, while maintaining client privacy and autonomy. However, PFL faces several challenges that can deteriorate the performance and effectiveness of the learning process. These challenges include data heterogeneity, communication overhead, model privacy, model drift, client heterogeneity, label noise and imbalance, federated optimization challenges, and client participation and engagement. To address these challenges, researchers are exploring innovative techniques and algorithms that can enable efficient and effective PFL. These techniques include several optimization algorithms. This research survey provides an overview of the challenges and motivations related to the model optimization strategies for PFL, as well as the state-of-the-art (SOTA) methods and algorithms which seek to provide solutions of these challenges. Overall, this survey can be a valuable resource for researchers who are interested in the emerging field of PFL as well as its potential for personalized machine learning in a federated environment.

Keywords: Personalized Federated Learning, Model Optimization, Distributed Machine Learning, Collaborative Learning, Privacy-Preserving

1. Introduction

1.1. Overview

Personalized Federated Learning(PFL) is an emerging research area that combines two popular machine learning techniques, namely Federated Learning(FL) and Personalization. The field of PFL has garnered significant attention, as it offers a promising solution for handling the challenges of personalized machine learning while upholding client data privacy and security. However, despite its potential, PFL is still a nascent area of research, necessitating a comprehensive survey to assess the current state-of-the-art (SOTA), highlight challenges, and outline future directions. This survey will contribute to a deeper understanding of PFL and its potential research domains regarding model optimization. In this research work, we provide a comprehensive overview of PFL model optimization strategies, including their definitions, algorithms, architectures, challenges, and future directions. We start by introducing the basic concepts of Personalization, then we discuss the various model optimization techniques with respect to their architectures. We focus on the future directions and open research questions in PFL strategies. Finally, we conclude the survey paper by summarizing the key contributions and limitations of the existing research on PFL and providing directions for future research.

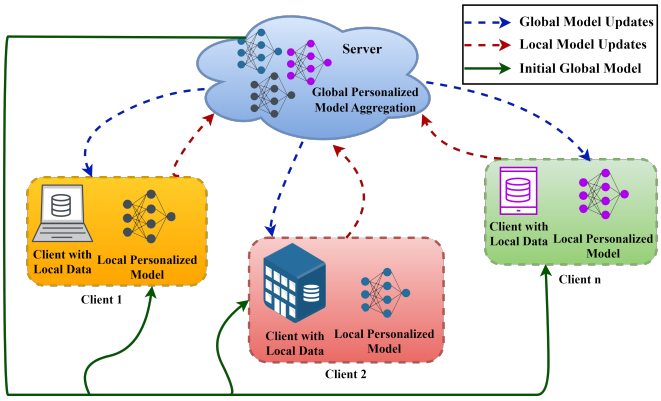

Figure 1: Personalized Federated Learning Architecture

1.2. Definition of PFL

FL is a method which allows number of clients to participate in training a shared model collaboratively, in which raw data is not exchanged. On the other hand, Personalization is an approach that tailors a model to a specific client or group of clients, by leveraging their individual data and preferences. Whereas the PFL is a specific type of FL where the objective is to train machine learning (ML) models that are tailored to the individual characteristics of each client in a federated network. In general, as shown in figure. 1, in PFL a global model is trained first over a central server. Then, every client further fine-tunes global model by using the local data, while preserving the privacy of its data. Then global model is personalized for every client, based on the local data and preferences, while

*Corresponding author
Email addresses: fahad.sabah@emails.bjut.edu.cn (Fahad Sabah), yuwen.chen@bjut.edu.cn (Yuwen Chen), yangzhen@bjut.edu.cn (Zhen Yang), raheemabdul@emails.bjut.edu.cn (Abdul Raheem), pmit@superior.edu.pk (Muhammad Azam), r.sarwar@mmu.ac.uk (Raheem Sarwar)

benefiting from the collective knowledge of all clients.

Algorithm 1 Algorithm for PFL

Input(s):
Initial global model with shared parameters and personalized layers for each client.
The dataset, split into multiple non-overlapping subsets assigned to each client.
Number of rounds and clients to be selected for each round.
The learning rate for updating global model's parameters.
Output(s):
The trained global model with updated shared parameters and personalized layers for each client.
The training loss and evaluation metrics of global model on a held-out validation set.
Algorithm:

1: Initialize global model and personalized layers.
2: Split the dataset into multiple non-overlapping subsets.
3: Assign each subset to a different client.
4: **for** each round of PFL **do**
5: Select a subset of clients to participate.
6: **for** each selected client **do**
7: Compute local gradient of the personalized layers.
8: Send the local gradient to the server.
9: **end for**
10: Aggregate local gradients to update global model's personalized layers.
11: Send updated personalized layers to each client.
12: **for** each selected client **do**
13: Freeze the shared parameters.
14: update personalized layers.
15: Compute the local gradient of entire model.
16: Send the local gradient to the server.
17: **end for**
18: Aggregate local gradients to update global model's shared parameters.
19: **end for**
20: Repeat steps 4 to 19 for number of rounds or until convergence.

Algorithm 1 illustrates the implementation of PFL, which is an extension of the basic FL algorithm that includes personalized layers for each client, allowing for client-specific adaptation of the model. There are many variations and optimizations depending on the specific use case and problem being solved.

In every round of PFL, the selected subset of clients or clients participates by providing their local gradients and updating the personalized layers and shared global model parameters. These local gradients are aggregated to update the global model's personalized layers and shared parameters. The updated personalized layers are then transmitted back to each client, where they are combined with the shared parameters to calculate the local gradient of the entire model. The local gradients from each client are then aggregated for updating the global model's shared parameters . This iterative process continues for the specified number of rounds, ultimately resulting in a trained global model as the final output.

Lets consider a PFL environment where $C = \{1, 2, \ldots, n\}$ denotes the set of n clients. Where every client $i \in C$ owns local data D_i. Let θ represent the global model parameters, which are to be learned collaboratively across all clients. At each iteration, the global model is updated using a subset of clients' local updates. The objective is to minimize the PFL loss function $\mathcal{L}_i(\theta)$ for each client $i \in C$. The loss function calculates the inconsistency between the global model and the local data at client i. The PFL problem can be expressed in mathematical form as follows:

$$\min_{\theta} \sum_{i \in C} w_i \mathcal{L}_i(\theta) \tag{1}$$

Where, w_i represents the weight assigned to client i to reflect its importance or contribution to the FL process. These weights can be determined based on various factors, for example; size of the local dataset, and computational resources, or data distribution properties. The global model parameters θ are updated iteratively using a federated optimization algorithm, which typically involves a communication and aggregation steps. The updated global model is obtained as follows:

$$\theta \leftarrow \text{Aggregate}(\{\theta_i\}_{i \in C}) \tag{2}$$

Where, θ_i is the updated local model parameters by client i using its local data. The aggregation function can be performed using various methods, such as averaging, weighted averaging, or other consensus algorithms. The FL process lasts till maximum number of iterations defined or until a convergence requirement is met. The final learned global model θ^* represents the collective intelligence obtained from the collaboration among all clients in the PFL setting.

1.3. Motivations for this paper

Despite of significant advances achieved in multiple applications, still there are several challenges that are required to be addressed in PFL. Some of these issues on which we are focussed, include; The increasing number of edge servers participating in FL has led to challenges related to models of existing PFL approaches [1] [2]. These challenges arise due to several factors e.g.; In deep learning tasks commonly used in FL, the size of model parameters can be substantial, ranging from tens to hundreds of megabytes. The FL training convergence typically requires hundreds or thousands of communication rounds. This leads to frequent long-distance transmission, global model aggregation, and backhaul of model parameters, which contribute to significant communication overhead. These factors collectively impose limitations on the scalability of PFL systems. Efforts should be focused on developing techniques that minimize communication overhead while ensuring effective collaboration and convergence in FL training processes. Following are the motivations behind conducting a research survey on PFL:

- **Emerging research area:** PFL is an emerging area of research which targets to leverage the benefits of FL while considering the personalized nature of client data. As this

field continues to evolve, there is a need to review and analyze the existing optimization strategies employed in PFL.

- **Diverse optimization approaches:** There is a wide range of optimization approaches and algorithms used in PFL. These include meta-learning techniques, adaptive optimization methods, gradient aggregation strategies, and more. A research survey can provide a comprehensive overview of these approaches, their strengths, limitations, and applicability in different scenarios.
- **Insights for future developments:** A survey can highlight the existing research gaps and challenges in optimization strategies for PFL. By understanding the limitations of current approaches, researchers can propose new techniques and directions for future developments.
- **Practical implications:** PFL has various applications in healthcare, finance, and other domains where privacy and personalization are crucial. A research survey on optimization strategies can provide insights and guidelines for practitioners and policymakers to effectively implement PFL systems and address potential optimization challenges.

1.4. Contributions

Number of surveys are available, which provide general overview of concepts, techniques, and applications of FL [3, 4]. Some of them specifically delve into FL from the robustness and privacy perspectives [5, 6]. The obective of this survey paper is to bridge the gap in the current literature on PFL. This paper provides characteristics, graphical overview, algorithms used, advantages, disadvantages and challenges in PFL model architectures for the researchers. The contributions of this survey are summarized as follows:

- **Comprehensive overview:** This survey provides a comprehensive overview of the existing strategies used in PFL. It catalogs different approaches, algorithms, and techniques employed in the field, giving researchers and practitioners a holistic understanding of the optimization landscape.
- **Categorization of Model Architectures:** The survey paper presents PFL model architectures, their benefits and limitations and their applicability in different PFL scenarios. This includes approaches; parameter de-coupling, neural architecture search, hyperparameter optimization, data augmentation, regularization, adversarial training, meta-learning and clustering.
- **Identifying research gaps:** Reviewing the literature, this survey identifies research gaps and open challenges in the field of optimization in PFL. These gaps include unexplored optimization techniques, limited evaluation on certain types of data or applications, or specific issues related to scalability, fairness, or privacy. These gaps can guide future research efforts.
- **Inspiring future research:** This research survey inspires and stimulates further research in the field by highlighting promising directions and emerging trends. It proposes novel research avenues, such as hybrid optimization approaches, adaptive learning rate scheduling, compression schemes, or personalized aggregation methods. By identifying these research opportunities, the survey can foster innovation and advancements in the field.

In summary, this research survey contributes by providing a comprehensive overview, classification, and performance analysis of existing approaches in PFL for model architectures. It also identifies research gaps, offers guidelines, and inspires future research directions. These contributions collectively enhance the understanding, development, and application of these strategies in PFL.

1.5. Structure of this paper

This research survey provides a comprehensive overview of model optimization in PFL, offering valuable insights and guidance for researchers and practitioners working in this field. Following is the brief overview of this research survey:
Introduction: In this section we present the overview of PFL, its basic architecture and algorithm. Then we discuss our motivation behind conducting this research survey. After that we discuss our contributions. **Research Work Classification:** The main contribution of this survey includes; classification and analysis of existing research on PFL model optimization techniques, detailed description of each technique, including advantages. **Limitations:** This section discusses the limitations of existing research works, **Challenges & Open Problems:** Identifies the key challenges and includes discussion of potential open problems and future directions of research in PFL. **Conclusion:** Summary of the key findings and contributions of the paper Discussion of the implications of the study for future research and practical applications of PFL.

2. Research Work Classification

In this section we classify the research works, including the main ideas and contributions, proposed architectures, advantages, algorithms, related to the model optimization in Personalized Federated Learning (PFL), as shown in figure. 2.

2.1. Model Optimization Techniques

In order to attain accurate and effective learning in PFL models, it is essential to optimize the architecture of the model. This section explores different model architectures that are often used in the context of PFL. These optimization methods can be tailored and integrated to fit specific PFL scenarios, taking into account factors such as data properties, client clients, and learning goals. The choice of techniques could be based on the inherent trade-offs between communication efficiency, model accuracy, privacy preservation, and computational constraints. Following are some widely employed model optimization techniques frequently utilized in PFL:

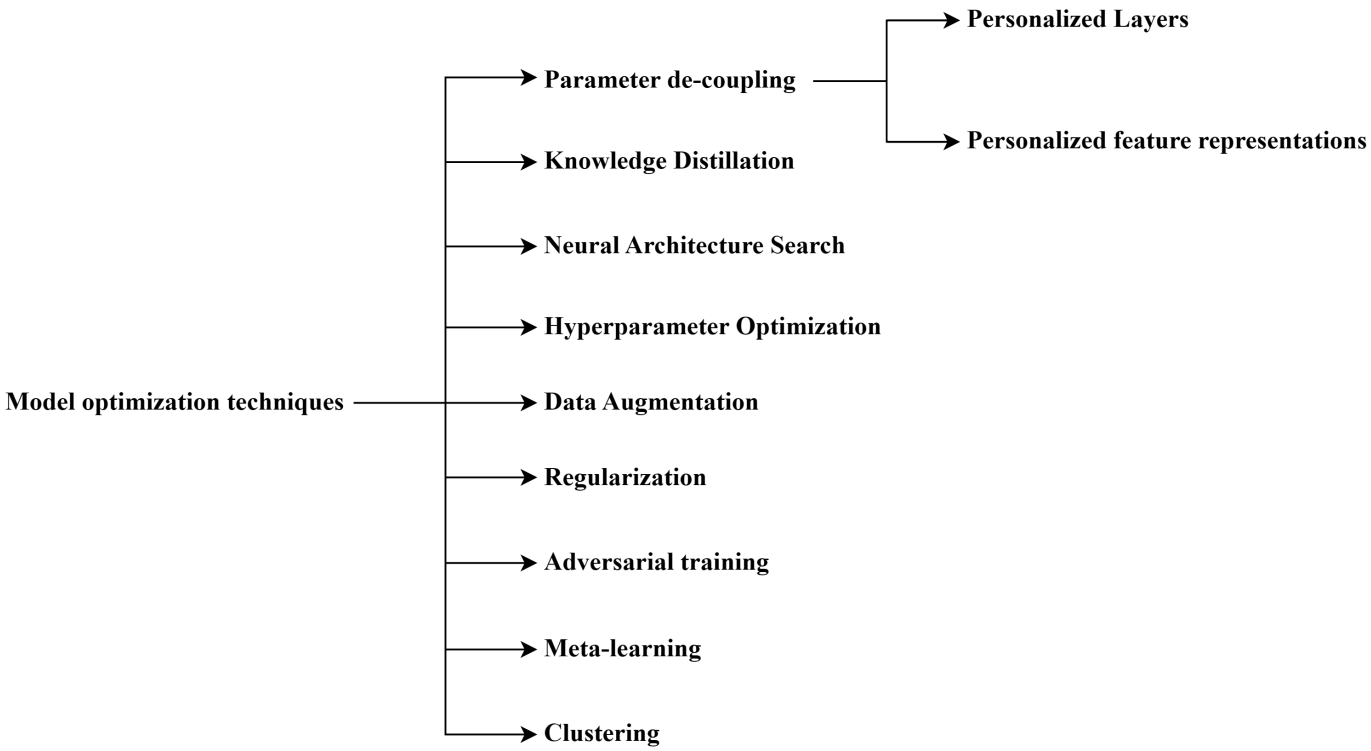

Figure 2: Classification of Model Optimization Strategies in PFL

2.1.1. Parameter de-coupling

A technique utilized in PFL to separate the local and global model parameters. The purpose of this methodology is to exclusively train private parameters on individual client clients without sharing them with the server. This approach allows the acquisition of task-specific representations, thereby enhancing personalization in the PFL framework. Parameter de-coupling is a widely employed technique in machine learning (ML) for the improvement in training efficiency of neural networks. Its fundamental concept involves isolating different sets of parameters within a neural network, allowing for independent updates to be performed on each set. By decoupling the parameters, the training process becomes more flexible and can adapt to specific requirements or constraints, leading to enhanced learning outcomes.

Parameter de-coupling proved to be a valuable technique for enhancing the efficiency and efficacy of training neural networks, particularly in scenarios involving large and intricate networks, where different sets of parameters may exhibit distinct convergence rates or update requirements. Algorithm 2 illustrates the implementation of parameter de-coupling:

It is important to highlight that the results obtained from Algorithm 2 can be utilized for fine-tuning the global model or training a new model specifically tailored to each individual client. The personalized parameters obtained through this process can also be used to evaluate the model's performance on individual clients or compare the performance across different clients. By decoupling the global and local parameters, Algorithm 2 offers a means to enhance the performance of models trained on distributed data by effectively capturing both global and local information. Furthermore, it helps mitigate the risk of overfitting and improves the model's generalizability.

In a research study [7], the authors proposed a PFL algorithm that allows the generation of personalized local models for each client while maintaining a up-to-date global model on a central server. The algorithm achieves this by dividing the deep neural network model into global and personalized layers, where the process of training consists on two stages: an earlier stage and a later stage. To enable effective personalization, the authors employed a layer-wise parameter division approach. This approach facilitates a cumulative learning strategy where parameters on different layers are updated at varying frequencies. The intention is to leverage the global features learned in the earlier stage to enhance personalization of the local models in the later stage. In [8], authors introduced pFedLA, a Layer-wise PFL framework that optimizes personalized model aggregation by considering the importance of each layer from different clients, resulting in superior performance compared to existing PFL methods. It incorporates a dedicated hyper-network per client on the server side and a parameterized mechanism to update layer-wise aggregation weights.

Parameter decoupling and split learning (SL) are two distinct private and distributed ML paradigms [9, 10]. Parameter decoupling, as applied in PFL, focuses on separating local and global model parameters. It allows clients to retain and update their personalized model parameters privately, while sharing and aggregating global model parameters with the central

Algorithm 2 Algorithm for PFL with Parameter De-Coupling

Input(s):
Global model parameters initialized on the server.
Partitioned groups of participating clients.
Training data on each client.
Personalization factor or threshold for the parameter de-coupling technique.

Output(s):
Global model parameters updated with personalized information from participating clients.
Personalized parameters for each client's local model.

Algorithm:

1: Global model parameters Initialization on server.
2: Partition the participating clients into groups.
3: **for** each group **do**
4: Send global model parameters to all clients in group.
5: **for** each client in the group **do**
6: Train local model of each client on its data.
7: Compute local gradients of the client's model with respect to the global parameters.
8: Compute the personalized parameters of client's model by applying parameter de-coupling technique.
9: Send the personalized parameters to the server.
10: **end for**
11: **end for**
12: Aggregate personalized parameters received from all clients.
13: Update global model parameters using aggregated personalized parameters.
14: Repeat steps 3-13 until convergence or maximum number of iterations is reached.

server. This approach enables clients to learn personalized representations while leveraging shared knowledge for improved performance. On the other hand, split learning (SL) partitions the deep neural network between the server and clients on a layer-by-layer basis. In SL instead of transmitting the server model to the client for training, just the split layer weights are transmitted, located at the boundary between the clients and server, are shared at time of forward propagation, whereas gradients are exchanged during backpropagation from the split layer. This privacy characteristic of SL offers an advantage as compared to FL as neither the clients nor server possess access to the global and local models completely [11]. SL also has certain drawbacks, one limitation is reduced training efficiency due to sequential client training, where clients need to be trained one after another, leading to increased latency. Additionally, SL may exhibit inferior performance compared to FL when dealing with non-IID (non-independent and identically distributed) data, as the split layers may not capture all relevant information from the clients' local data. Furthermore, SL involves higher communication overheads, as gradients need to be exchanged between the clients and server during the training process, which can impact scalability and efficiency in distributed settings. It is important to consider these trade-offs when selecting between parameter decoupling in PFL and split learning, depending on the specific constraints and requirements of the application [12].

There are two common configurations for parameter decoupling used in FL. First configuration, known as "base layers + personalized layers" was introduced by Arivazhagan et al. in their work [13]. In their work, clients keep private their personalized layers for locally training, which allows the clients to learn personalized task-based representations. whereas, the base layers are shared with server, enabling the learning of low-level generic features that are shared across all clients. The second design focuses on considering personalized feature representations for each client. Bui et al. explored this approach in a classification model based on a Bidirectional Long Short-Term Memory (LSTM) architecture trained with FL [14]. In this setup, client embeddings are treated as private model parameters specific to each client. The character embeddings, LSTM layers, and Multilayer Perceptron (MLP) layers, on the other hand, are considered as FL model parameters shared among all clients. In addition, Liang et al. proposed a method called Local Global Federated Averaging (LG-FedAvg) [15]. LG-FedAvg integrates local representation learning with global federated training to improve communication and computational efficiency. The different configurations and approaches for parameter decoupling in FL provide flexibility in learning personalized representations while leveraging shared knowledge and improving efficiency in the training process.

Personalized Layers.
Parameter de-coupling can be extended to incorporate personalized layers, which can further enhance the performance of the model. Personalized layers are layers that are specific to each client and are learned independently of the global parameters. Recognizing the significance of personalization in FL, in [16], authors provide an overview of recent research conducted in this area, highlighting the advancements and strategies developed to solve the issue of statistical heterogeneity and enhance performance of model on a client-specific basis.

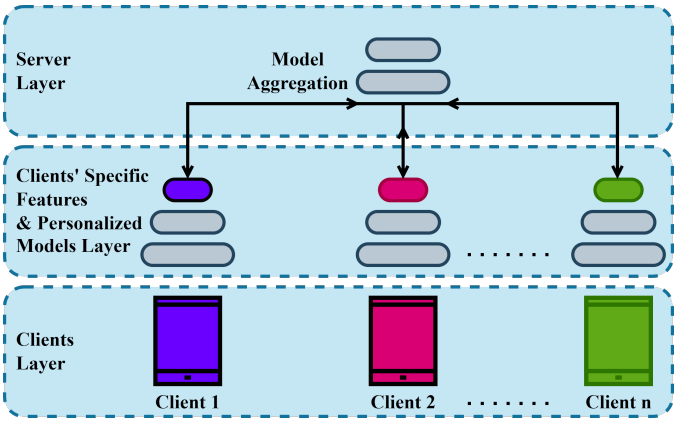

Figure 3: Privacy in Federated Learning Using Private Personalized Layers [17]

The figure. 3 illustrates that all client clients possess a common set of base layers with identical weights (highlighted in

blue), alongside individual personalization layers that have the potential to adapt to each client's specific data. The Algorithm 3 represents parameter de-coupling in FL with personalized layers:

Algorithm 3 Parameter De-coupling with Personalized Layers

Input(s): Personalized layers L_i for each participant i, global model parameters θ

Output(s): Updated global model parameters θ'

Algorithm:

Initialize $\theta' \leftarrow \theta$
for $i = 1$ to N **do** ▷ N is the number of participants
    Set model parameters to θ' ▷ Initialize with global parameters
    Set personalized layer to L_i ▷ Replace the shared layer with personalized layer
    Train the model on participant i's data using personalized layer L_i
    Update the global model parameters θ' with participant i's trained parameters
end for
Return θ'

Algorithm 3 uses personalized layers L_i for each participant i. The main steps of the algorithm involve initializing the global model parameters, setting the personalized layer for each participant, training the model using the personalized layer, and updating the global model parameters based on each participant's trained parameters. The incorporation of private personalized layers in the proposed scheme by [17], aims to improve model accuracy through local adaptation while minimizing the transmission of model information to the server.

Personalized Feature Representations.
Parameter de-coupling in FL with personalized feature representations can be done by splitting the model into a shared feature extractor and multiple personalized classifiers, where each classifier is responsible for predicting labels for a specific subset of the data. During training, each client updates personalized classifier using only local data and then sends the updated classifier to the server. Then the server updates shared feature extractor based on the decoupled gradients obtained from the personalized classifiers.

Algorithm 4 represents the parameter de-coupling in FL with personalized feature representations. The algorithm 4 takes personalized feature representations X_i and global model parameters theta as input and returns updated global model parameters θ' as output. The main steps of the algorithm involve computing the local gradients and updating the global model parameters on the basis of those gradients. In [18], the authors proposed a novel framework and algorithm for FL that aims to learn a shared data representation across clients while incorporating unique local heads for each client. Their algorithm takes advantage of the distributed computational power available across clients to perform multiple local updates with respect to low-dimensional local parameters during each update of the shared representation. FL and multi-task learning have shown great success by leveraging knowledge from different tasks to overcome limitations imposed by limited training samples. To deepen the understanding of mechanisms underlying knowledge utilization, the authors conducted a comprehensive study in [19] on the sample complexity of FL algorithms, focusing on their expected population risks. They examined several representative algorithms: (i) training personalized models using local data exclusively, (ii) training a single model jointly using data from all clients, and (iii) jointly training a shared feature representation while learning individual classifiers for different clients, which has gained popularity recently. The study revealed that the effectiveness of the shared representation algorithm is influenced by various factors, including task similarities, model dimensionality, and sample size.

Algorithm 4 Parameter De-coupling with personalized feature representations

Input(s): Personalized feature representations X_i for each participant i, global model parameters θ

Output(s): Updated global model parameters θ'

Algorithm:

Initialize $\theta' \leftarrow \theta$
for $i = 1$ to N **do** ▷ N is the number of participants
    Compute local gradients $g_i \leftarrow \nabla_\theta \mathcal{L}(X_i, \theta')$ ▷ $\mathcal{L}$ is the loss function
    Update $\theta' \leftarrow \theta' - \eta g_i$ ▷ η is the learning rate
end for
Return θ'

In [26], authors proposd a novel FL method called AlignFed. This method addresses the feature shift issue in cross-domain FL scenarios by dividing the model into personalized feature extractors and a shared classifier. The personalized feature extractors align the features of different clients to specific points in the feature space, mitigating the feature shift problem. The shared classifier then aggregates knowledge across clients in the aligned feature space, improving model performance. In [27], authors proposed PFL method for ECG classification, in which initially a global model is trained by using a FL framework on number of clients. This global model is then used as a starting point for training local models with the clients' private data. To address feature inconsistency and improve fitting of the local data, a "feature alignment" module is introduced, comprising global alignment and local alignment components. This approach aims to overcome challenges related to insufficient data, privacy preservation, and local deployment in ECG classification. In [28], the authors present a novel framework called Partial Model Aggregation Federated Learning (PMA-FL) to addressed challenges in communication resources and data heterogeneity in FL. PMA-FL focuses on aggregating lower layers for feature extraction while keeping upper layers of neural network at clients for personalized pattern recognition. In [20], the authors introduce a novel FL architecture called multibranch multilevel federated learning (MBMLFL) to address limitations in feature extraction and hierarchical structure. MBMLFL improves feature extraction

Table 1: Summary of contributions in PFL Model Optimization Techniques using Parameter de-coupling

Model(s) / Technique(s)	Main Idea/Contribution	Advantage(s)/Real-World Application(s)
FedVF, FedVFDP [7]	Personalized local models for individual clients by maintaining the most recent global model on a central server through a unified federated training process.	Health monitoring, language modeling, road traffic prediction
pFedLA [8]	A parameterized mechanism to update the aggregation weights at each layer, enabling a progressive exploration of inter-client similarity and achieving precise model personalization.	Can be used in large-scale scenarios, where communication capacity is limited.
FedPer [13]	A novel approach for federated training of deep feed-forward neural networks. FedPer utilizes a combination of base layers and personalization layers to address the challenges posed by statistical heterogeneity, thereby mitigating their adverse effects.	Personalization, recommendation, fraud detection, etc.
LG-FEDAVG [15]	A novel FL algorithm that concurrently learns compact local representations on each individual client and a global model that encompasses information from all clients.	Personalized mood prediction from real-world data where privacy is key.
MBMLFL, MBMLFL-SGD, M2DCFedAvg [20]	Especially designed FL architecture(s) having an excellent feature extraction	Friendly, useful for large scale cooperation, enhances privacy [21, 22, 23, 24].
TailorFL [25]	A dual-PFL framework that customizes a submodel for each client. This framework incorporates personalized structure during the training phase and utilizes personalized parameters for local inference.	Image Classification, Human Activity Recognition, health monitoring, language modeling, road traffic prediction
FedRep [18]	A novel FL framework and algorithm to learn a shared data representation across multiple clients while maintaining unique local heads for each client.	Generalization to new clients, health monitoring, language modeling, road traffic prediction

by leveraging multiple branches and levels, each with its specific effect. The framework, designed to be privacy-friendly, extends FedAvg and proposes the M2DCFedAvg algorithm for distributed optimization.

2.1.2. Knowledge Distillation

Knowledge distillation is a technique commonly used in machine learning to transfer knowledge from one model, known as the Teacher model, to another model, called the student model. This process involves training the local model to copy the behavior and predictions of the more complex teacher model.

The objective function for knowledge distillation is given by:

$$L_{\text{total}} = L_{\text{ce}} + \lambda \cdot L_{\text{dist}} \tag{3}$$

where, L_{total} is the total loss combining standard and distillation losses, L_{ce} is cross-entropy loss between Local model predictions and ground truth labels, L_{dist} is the distillation loss measuring the difference between Local and global predictions, λ is a weighting factor used to control the effect of the distillation loss.

In [30] authors proposed a novel model; Federated Codistillation (FedCodl), in which a distillation term is added to the

Algorithm 5 Knowledge Distillation in PFL

Input(s): Centralized Global model T, Client clients $C_1, C_2, \ldots, C_n$

Output(s): Updated global model

Algorithm:

for $i = 1$ to n **do**
  Initialize Local model S_i on client client C_i
  Obtain personalized data D_i on client client C_i
  Train S_i on D_i with distillation
  Distillation Loss:
  Generate soft targets Q_i using T on D_i
  Calculate cross-entropy loss L_{ce} between S_i predictions and ground truth labels
  Calculate distillation loss L_{dist} between S_i predictions and Q_i
  Calculate total loss $L_{\text{total}} = L_{\text{ce}} + \lambda \cdot L_{\text{dist}}$ ▷ λ is the weight for distillation loss
  Update S_i parameters by optimizing L_{total}
end for

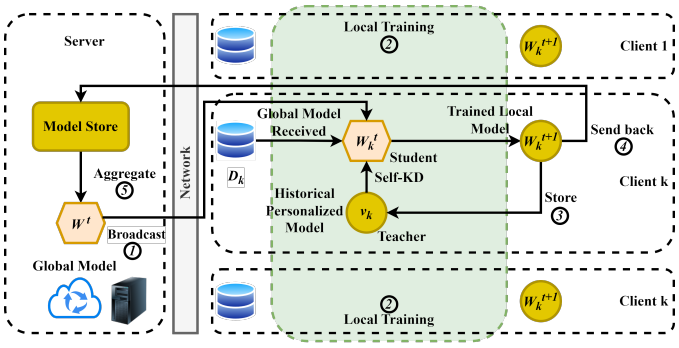

Figure 4: Illustration of pFedSD workflow [29].

local objective function, so that local models can be trained on outputs of the global model. They also extend FedCodl to Federated Two-way codistillation (Fed2Codl) to personalize local models for each device, meanwhile the global model is retained and iteratively updated in parallel. In their work [31], the authors proposed MetaFed, a novel framework aimed at facilitating reliable FL in heterogeneous federations. MetaFed introduces a unique technique called Cyclic Knowledge Distillation, which enables the generation of personalized models for every federation instead of relying on server. The MetaFed framework adopts a perspective where each federation is considered as a meta distribution. This perspective allows for the cyclic aggregation of knowledge from the different federations in a collaborative manner. The training process within MetaFed is based on two primary stages: common knowledge accumulation and personalization. During the common knowledge accumulation, knowledge is aggregated from the participating federations. This aggregation process aims to capture the shared information and patterns across federations, enabling the development of a common model that benefits from the collective knowledge of the entire system. This shared knowledge accumulation helps to improve overall performance and generalization capabilities of model. Following the common knowledge accumulation, the personalization stage takes place. In this stage, the aggregated common model is adapted or personalized to the specific characteristics and requirements of each individual federation. This personalization allows the model to be customized and tailored to the unique data distributions and task specifications within each federation, enhancing its performance and relevance to the local context. By cyclically repeating the two stages, MetaFed ensures an iterative and collaborative process of knowledge aggregation and personalization among federations.

In [35], authors proposed an innovative solution to address the challenges posed by heterogeneous FL. They introduce a data-free knowledge distillation method as part of their approach. The key idea involves training a lightweight generator at the server level, enabling the aggregation of client information without requiring access to the client's raw data. Afterwards, resulting knowledge is distributed to individual clients, who utilize this acquired knowledge as an inductive bias during their local training process. In their research, Jeong et al. [32] introduced a novel PFL algorithm. The proposed algorithm leverages knowledge distillation (KD) techniques to empower individual client clients by enabling them to estimate statistical distances between their local models. This approach facilitates performance enhancement for each client without the need to share their local data. By assessing the similarity between intermediate outputs derived from local samples, akin to knowledge distillation, the clients can autonomously and effectively improve their models within the decentralized Federated Learning (FL) framework.

In the research work [36], the authors propose a novel FL algorithm called FedHKD (Federated Hyper-Knowledge Distillation). This algorithm leverages KD techniques to train local models on client clients. In FedHKD, each client extracts the means of local data representations and the corresponding soft predictions, known as "hyper-knowledge," which are then transmitted to the server. The server aggregates this hyper-knowledge and broadcasts it back to the clients to aid in their local training. In the research paper [29], the authors conducted an investigation into Personalized Federated Learning (PFL), with a specific focus on training models that exhibit strong performance for individual clients. They observed that the initialization process during each communication round results in the loss of historical personalized knowledge. Building upon this observation, they proposed a novel PFL framework called pFedSD, which incorporates self-knowledge distillation. The pFedSD framework, depicted in Figure 4, enables clients to distill knowledge from their previous personalized models into their current local models. This approach facilitates the rapid retrieval of personalized knowledge for the most recent initialized clients.

2.1.3. Neural Architecture Search (NAS)

A technique used to automatically discover effective neural network architectures for a given task. It involves exploring a search space of possible network architectures and selecting the best architecture based on a predefined objective, such as maximizing accuracy or minimizing computational resources. But it is important to note that combining NAS with PFL introduces additional challenges. Researchers are actively exploring the integration of NAS with PFL to address these challenges and unlock the potential benefits of automated architecture search in PFL scenarios.

The algorithm 6 takes participant attributes A_i, federated data D, and the search space $\mathcal{S}$ as input, and it outputs the optimal architectures A^* for each participant. The main steps of the algorithm involve encoding participant attributes into an architecture representation, initializing and updating architectures based on candidate architectures, training and evaluating models with the candidate architectures, and selecting the optimal architecture for each participant based on their performance.

In [45], the authors introduced Resource-aware Federated Learning (RaFL), a framework designed to address the challenges of data heterogeneity and system/resource heterogeneity in FL systems. RaFL leverages Neural Architecture Search (NAS) to allocate resource-aware models to edge clients, allowing customized model deployment based on diverse com-

Table 2: Summary of contributions in PFL Model Optimization Techniques using Knowledge Distillation

Model(s) / Technique(s)	Main Idea/Contribution	Advantage(s)/Real-World Application(s)
FedCodl, Fed2Codl[30]	Automatically tune the parameter specific to heterogeneous datasets at different phases of training.	personalized learning and performance improvements.
KDGaN[32]	A personalized and fully decentralized FL algorithm that leverages KD techniques. This approach empowers each client to discern statistical distances between local models, allowing for enhanced personalization and improved performance in the FL setting.	advantageous for agents with small datasets, as it eliminates the necessity for a central server while still providing benefits in terms of personalized learning and performance improvements.
pFedSD [29]	Expedites the retrieval of personalized knowledge for newly initialized clients by enabling them to distill the knowledge obtained from previous personalized models to their current local models.	beneficial to applications with small datasets
MetaFed [31]	Utilizes Cyclic KD to obtain a personalized model for each federation without relying on a central server.	healthcare related applications
FedKD [33]	Utilizes adaptive mutual KD and dynamic gradient compression techniques to optimize the learning process.	Strong model with privacy-preserving and less communication cost, personalized news recommendation, adverse drug reaction (ADR) mentioning text detection, and medical named entity recognition (NER).
Def-KT [34]	Incorporates mutual knowledge transfer among local clients. Clients fuse their learned knowledge by transferring it to each other, facilitating a collaborative learning process.	IoT applications
FeDGen [35]	A lightweight generator is trained by the server to combine client information in a data-free manner. This generated knowledge is then shared with the clients, serving as a guiding factor during their local training processes.	Healthcare, Smart Cities, Edge Computing, Mitigates discrepancy of latent distributions across clients, and directly regulates local model updating.

putational resources. It incorporates a multi-model architecture fusion scheme for aggregating distributed learning results. By combining NAS and FL, RaFL offers a promising solution for handling data and system heterogeneity in FL systems, enabling efficient model deployment and learning while preserving privacy in distributed settings. RaFL is a resource-aware federated neural architecture search to search for resource-tailored models for edge clients as shown in figure. 5. The OFA supernetwork has been trained by minimizing the objective function shown in Equation:

$$\min_{\Theta} \sum_{arch_i} L_{\text{val}}(C(\Theta, arch_i)) \tag{4}$$

where Θ represents the search space of architecture parameters, $arch_i$ denotes a particular architecture within the search space, $C(\Theta, arch_i)$ represents the neural network model with architecture $arch_i$ and parameters Θ, and $L_{\text{val}}(C(\Theta, arch_i))$ is the validation loss or error of the model on the given dataset.

In [41], the authors introduce a novel approach called Federated Modular Network (FedMN) for PFL. While PFL has gained popularity for collaboratively training federated models with privacy constraints, to overcome the limitations of existing

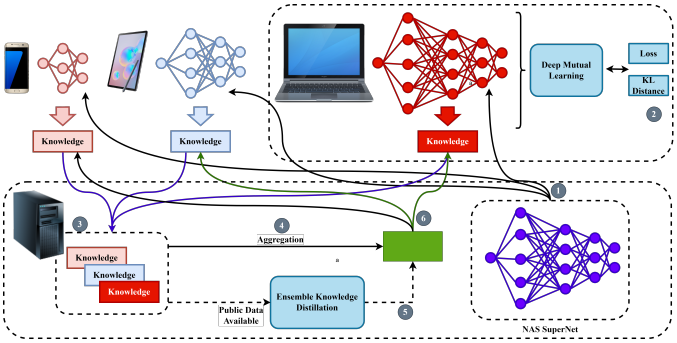

Figure 5: Resource-Aware Heterogeneous FL using NAS [45].

Personalized Federated Learning (PFL) approaches, which may result in sub-optimal solutions due to divergence in the joint distribution among local clients, the FedMN method introduces an adaptive selection mechanism. It involves assembling heterogeneous neural architectures by choosing sub-modules from a module pool, specifically tailored to the unique characteristics and requirements of individual clients.

In [39], the authors proposed a novel method called fed-

Table 3: Summary of contributions in PFL Model Optimization Techniques using Neural Architecture Search(NAS)

Model(s) / Technique(s)	Main Idea/Contribution	Advantage(s)/Real-World Application(s)
FedSup, E-FedSup [37]	Combines the training of supernet architectures and FL. FedSup addresses scenarios where clients both send and receive a supernet containing all possible architectures sampled from their own local supernet. E-FedSup, a sub-model is sent to clients during broadcast stage, resulting in improved efficiency also incorporates different approaches to enhance supernet training.	Enhanced representation capabilities compared to its static counterpart. Additionally, the supernet mitigates the problem of excessive energy consumption and CO_2 emissions associated with designing dedicated Deep Neural Networks (DNNs)
FedorAS [38]	Designed to address the challenges of discovering and training promising architectures in a resource-aware manner. Specifically focuses on scenarios where clients have varying capabilities.	non-IIDness handling
FedClassAvg [39]	Unlike existing methods that necessitate the collection of auxiliary data or model weights to generate a counterpart It only requires clients to communicate with few of fully connected layers.	Ensures high communication efficiency, Stabilize decision boundaries Improved local feature extraction abilities for clients.
GsONG [40]	Integrates the searchability of evolutionary computation with the learning ability of a hybrid artificial neural networks method.	Computational efficiency
FedMN [41]	Adaptive selection of submodules from pool of modules and their integration to construct heterogeneous neural architectures that are tailored to the specific characteristics and needs of individual clients.	Reduces the communication burden between the server and the clients
SPIDER [42]	Focuses on exploration and identification of personalized neural network architectures tailored to individual clients.	Can be beneficial for business models.
FEDPNAS [43]	Learns a base architecture that can be structurally personalized, allowing for quick adaptation to each local task. This approach enables efficient and effective customization of architectures, tailored to requirements of individual clients.	superior performance compared to other benchmarks on heterogeneous multitask scenarios.
RT-FedEvoNAS [44]	Approach that optimizes model performance while reducing local payload. It incorporates a double-sampling technique to reduce computational and communication costs.	reduced computational and communication costs, suitable for real-time federated NAS applications

erated classifier averaging (FedClassAvg) for PFL. The goal of FedClassAvg is to enable clients with heterogeneous neural network architectures to participate in collaborative training without exchanging private data, while ensuring communication efficiency. FedClassAvg focuses on supervised learning tasks using deep neural networks, which consist of feature extractor and classifier layers. The method aggregates the weights of the classifier layers across different clients, creating an agreement on decision boundaries in the feature space. This allows clients with non-iid data to effectively learn scarce labels, improving the generalization performance of the federated model. The contribution of FedClassAvg lies in its ability to improve the collaborative training process, achieve better generalization performance, and reduce communication and computation overhead in PFL scenarios. In [46], the authors tackled the challenge of achieving personalized sleep state tracking using deep learning techniques.

In [38], the authors explored the combination of NAS and FL in the context of cross-client federated settings. While NAS has shown promise in centralized settings, it relies on access to centralized datasets, which may not always be available. Furthermore, existing work combining NAS and FL has primarily focused on cross-silo federated settings, assuming homogeneous compute environments with datacenter-grade hardware. In this study, the authors investigate the feasibility of designing architectures with different footprints in a cross-client federated setting, where clients vary significantly in terms of capabilities, availability, and scale. They propose a system called FedorAS shown in figure. 5, which aims to discover and train architectures in a resource-aware manner while dealing with clients having different capabilities and non-IID distributed data.

In [37], authors presented a novel framework called Fed-

Algorithm 6 Neural Architecture Search in PFL

Input(s): Participant attributes A_i for each participant i, federated data D, search space $\mathcal{S}$
Output(s): Optimal architectures A^* for each participant
Algorithm:

Initialize optimal architectures A^* for each participant
for $i = 1$ to N **do** ▷ N is the number of participants
  Encode participant attributes A_i into architecture representation R_i
  Initialize architecture A_i from search space $\mathcal{S}$
  while not converged **do**
    Sample a set of candidate architectures C_i from A_i
    for $A_c \in C_i$ **do**
      Train model with architecture A_c on participant i's data D_i
      Evaluate model performance on validation data
      Update A_i if A_c outperforms the current best architecture
    end for
  end while
  Set the optimal architecture A_i^* as the final architecture for participant i
end for
Return A^* for each participant

eration of Supernet Training (FedSup) that addresses the challenges posed by data and system heterogeneity in the context of efficient deployment of deep neural networks. While previous approaches have focused on either FL or NAS separately, FedSup combines both approaches to tackle data and system heterogeneity concurrently. The framework leverages the observation that the parameter averaging in FL can be viewed as weight-sharing in supernet training. By incorporating weight-sharing techniques into the FL averaging process (FedAvg), FedSup enables clients to exchange a supernet that encompasses various sampled architectures from their local datasets. Existing FL approaches typically use predefined architectures that are shared among all clients, neglecting the private nature of client data and the variations in data distributions across clients. To overcome this limitation, the authors of [42] proposed SPIDER, a framework for searching personalized neural architectures in FL. SPIDER incorporates two key features to enable personalized adaptations: **i. Alternating Optimization:** SPIDER optimizes both a homogeneous global model (Supernet) following conventional FL principles and heterogeneous local models that are connected to the global model through weight sharing-based regularization. This allows for personalized adaptations based on each client's specific data distribution. **ii. Neural Architecture Search (NAS):** SPIDER utilizes a novel NAS method to obtain architecture-heterogeneous local models. It progressively selects optimal subnets through operation-level perturbations, using accuracy as the criterion.

The authors of [43] proposed FED PNAS, a personalized Neural Architecture Search algorithm designed for FL. Their approach enables the learning of a base architecture that can be customized to adapt to the unique characteristics of each local task. By personalizing the model structure, FED PNAS aims to improve FL performance in scenarios with heterogeneous multitask settings. To address the customization of model architecture for individual tasks in the FL workflow, FEDPNAS adopts a sub-network representation approach. For each task, the model architecture is represented as a sub-network sampled from a large, over-parameterized network. The sampling distribution and the parameters of the sampled network are jointly learned in a collaborative manner. This process builds upon the Discrete Stochastic NAS (DSNAS) method [47], which lacked the capability to customize architecture for individual tasks.

2.1.4. Hyperparameter Optimization

This strategy can be used for personalization by searching for the optimal hyperparameters for each client based on their data and preferences. Hyperparameter optimization (HPO) in PFL refers to the selection process of the optimal hyperparameters for training models in an FL setting, where each participant trains a local model using their own data while collaborating with a central server. Algorithm 7 illustrates the HPO in PFL.

Algorithm 7 Hyperparameter Optimization in PFL

Input(s):
Participants' datasets $\{D_1, D_2, ..., D_N\}$
Set of hyperparameters $\mathcal{H}$
Number of iterations T
Output(s):
Optimal hyperparameters $\hat{h}$ for all clients
Algorithm:

Initialize best hyperparameters $\hat{h}$
Initialize best performance $\hat{p} = 0$
for $t = 1$ to T **do**
  Sample hyperparameters h from $\mathcal{H}$
  Broadcast h to all participants
  for each participant i **do**
    Train local model M_i using D_i and h
    Evaluate M_i on local validation set to obtain performance p_i
    Transmit p_i to the server
  end for
  Aggregate participants' performances: $P = \{p_1, p_2, ..., p_N\}$
  Calculate average performance $\bar{p} = \frac{1}{N} \sum_{i=1}^{N} p_i$
  if $\bar{p} > \hat{p}$ **then**
    Update best hyperparameters: $\hat{h} = h$
    Update best performance: $\hat{p} = \bar{p}$
  end if
end for
Return $\hat{h}$

In [52], the authors focused on addressing client heterogeneity and non-IID data in a FL setting. They investigated the effectiveness of two FL algorithms, namely FedAvg and FedProx, using a heterogeneous data split consisting of three different forms of cancer: cervical, lung, and colon. The authors also

Table 4: Summary of contributions in PFL Model Optimization Techniques using Hyperparameter Optimization

Model(s) / Technique(s)	Main Idea/Contribution	Advantage(s)/Real-World Application(s)
HPN [48]	This approach ensures privacy protection for individual clients while effectively optimizing hyperparameters for improved performance.	address the heterogeneity among clients.
HyperFed [49]	Global-sharing imaging network and institution-specific hypernetwork are used. The global-sharing imaging network learns common features from different institutions, improving performance and personalization in CT imaging.	Competitive performance in CT reconstruction compared with several other state-of-the-art methods.
HNTROJ [50]	A locally infected model containing a backdoor is transferred to legitimate and personalized local models generated by the HyperNetFL model. Furthermore, the method incorporates backdoor-resistant training algorithms for FL into the HyperNetFL framework.	Outperforms data poisoning and model replacement attacks and bypasses robust training algorithms.
pFedHN [51]	Addresses the problem of parameter sharing and personalization in FL. In pFedHN, a central hypernetwork is trained for the generation of set of models, with every client having their own unique model.	The use of hypernetworks enables effective customization and adaptation of models in the FL setting.

examined the impact of hyperparameters in FL, which presents unique challenges because of distributed nature of the learning process. To optimize the hyperparameters, the authors employed Bayesian optimization, a technique used to fine-tune the hyperparameters and find optimal values that enhance performance. They performed HPO for both local and global models within the FL environment. This highlights the effectiveness of FedProx in addressing challenges related to client heterogeneity and non-IID data in FL. To overcome the challenges of PFL, in [48], authors propose a solution that involves training a Hyper Parameter Network (HPN) capable of determining personalized hyperparameters using client encodings. Another study focuses on PFL, which involves training ML models for various clients with unique data distributions. This scenario presents challenges in managing data disparities and minimizing communication costs during collaborative model training. To overcome these challenges, in [51], authors propose a novel approach called pFedHN (personalized Federated HyperNetworks) that leverages hypernetworks. In pFedHN, a central hypernetwork model is trained that generates client-specific models, tailoring each model to the specific requirements of individual clients.

In [50], authors aimed to uncover and analyze previously backdoor risks which were undisclosed in HyperNet-based PFL (HyperNetFL) through an investigation of poisoning attacks. Building upon these findings, they introduced a first of its kind, model transferring attack called HNTROJ. This model targets the transfer of a locally infected backdoor model to all legitimate personalized local models that are generated by the HyperNetFL model. This is achieved by effectively leveraging consistent malicious local gradients computed across compromised clients throughout the entire training process. The basic objective of HNTROJ is to minimize number of compromised clients required for a successful attack, while maintaining stealthiness to avoid any noticeable degradation in the model's utility on legitimate data samples.

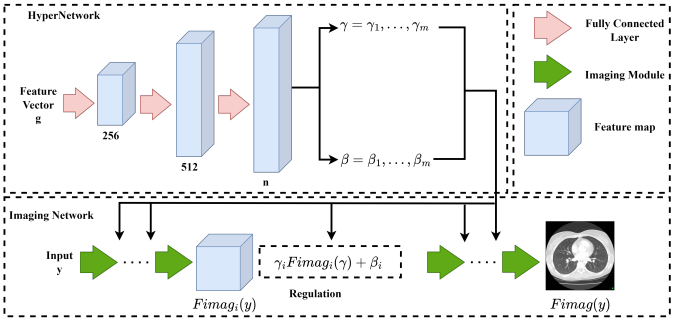

Figure 6: The architecture of the HyperFed [49].

In their study [49], Yang et al. proposed HyperFed, a novel approach for personalized CT imaging in the field of Federated Learning (FL). HyperFed exploits the concept of decomposing the optimization problem within medical institutions into two components: the local data adaptation problem and the global CT imaging problem. Figure. 6 illustrates how HyperFed tackles these components by integrating an institution-specific hypernetwork and a global-sharing imaging network. The global-sharing imaging network captures stable and effective common features from diverse institutions, while the institution-specific hypernetwork obtains hyperparameters to customize the global-sharing imaging network for personalized local CT reconstruction.

2.1.5. Data Augmentation

This technique involves generating new training data by applying transformations to existing data. This can be used for personalization by generating new training data that is more similar to a specific client's data. It is used in ML to increase the diversity and size of a dataset through generating new objects or data points on the basis of existing ones. In PFL, data augmentation can be applied to improve the performance and accuracy of the model by creating more training data for each client. One approach to data augmentation in PFL is to apply augmentation techniques locally on each client's dataset before sending the updated dataset to the central server for model training. This can be done using standard augmentation techniques such as random cropping, flipping, rotation, and color jittering. Another approach is to apply data augmentation techniques at the server level, before sending the updated model to each client. This can be done by generating synthetic data points based on the existing data, using techniques such as Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), or Deep Belief Networks (DBNs).

Algorithm 8 Data Augmentation in PFL

Input(s): Client datasets $D_1, D_2, ..., D_N$
Output(s): Augmented client datasets $D'_1, D'_2, ..., D'_N$
Algorithm:

1: **for** $i \leftarrow 1$ to N **do** ▷ Apply data augmentation locally on each client
2: $D'_i \leftarrow$ Empty dataset
3: **for** each sample x in D_i **do**
4: Apply random cropping, flipping, rotation, color jittering, etc. to x
5: Add augmented sample x' to D'_i
6: **end for**
7: **end for**
8: Send $D'_1, D'_2, ..., D'_N$ to the central server

The Algorithm 8 assumes that client datasets $D_1, D_2, \ldots, D_N$ are already available. It then iterates over each client's dataset and applies data augmentation techniques locally to create augmented datasets $D'_1, D'_2, \ldots, D'_N$. Finally, the augmented datasets are sent to the central server for model training.

The objective of a typical FL model can be written as the following problem:

$$\min_{w} F(w) = \sum_{k=1}^{K} \frac{n_k}{N} F_k(w_k) \tag{5}$$

Where, w is the global model, F is the objective function, K is the number of clients, n_k is the number of samples at client k, N is the total number of samples, and w_k is the local model at client k. The objective function $F_k(w_k)$ represents the local loss function at each client k that measures the model's performance on its local data.

Existing PFL methods often assume a uniform distribution of global data across clients, overlooking the challenges posed by data heterogeneity and long-tail distributions. These factors can significantly impact the performance of personalized models. To address this joint problem, authors in [53] devised a new method called Federated Learning with Adversarial Feature Augmentation (FedAFA) shown in figure. 7. FedAFA focuses on optimizing personalized models for every client by generating a balanced feature set that enhances the representation of local minority classes. This is achieved through an adversarial feature augmentation process. FedAFA transfers knowledge from the local majority class features to generate local minority class features. By learning from adversarial examples, FedAFA effectively augments the feature space and improves the representation of underrepresented classes. A fusion strategy that combines personalized and generic approaches is introduced by; [54], taking into account the network layer function. For each network layer number, a fusion threshold is designed. The fusion weights for the feature extraction layer parameters of each client are calculated using the L2-Norm negative exponential similarity metric. This approach improves the efficiency of personalized collaboration with heterogeneous data. Additionally, a federated global model approximation strategy is applied to the fully-connected layer of the network. This generic fusion strategy helps mitigate the overfitting that can occur when focusing solely on personalized models.

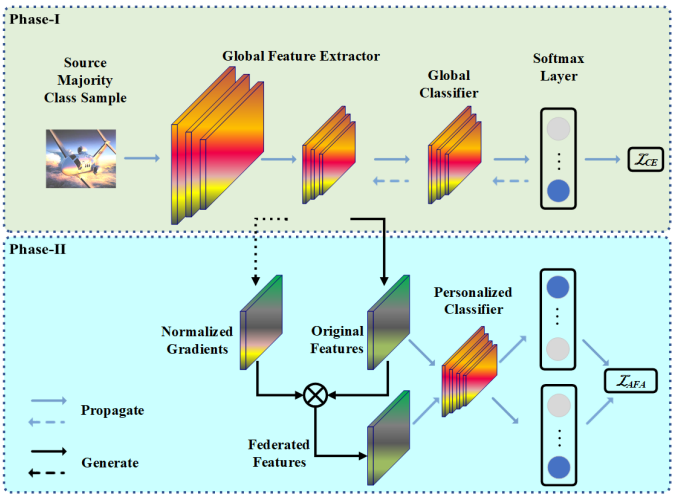

Figure 7: An overview of FedAFA. [53]

In [56], authors introduced PerFL, a PFL framework that aims to enhance the performance of clients' models by leveraging prior information that can be shared between clients. The traditional FL approach may not be effective for a small number of clients, as it results in a significant loss of localization information. To address this, PerFL utilizes client features that can be shared and calculates the incidence matrix of all clients based on the available shareable side information. Instead of updating the local models using all clients, PerFL selects similar clients and updates the models accordingly. The framework employs neural networks as the classification model and iteratively learns the parameter matrices at each client.

The authors in [57], proposed a self-balancing FL framework called Astraea. Astraea aims to alleviate imbalances in

Table 5: Summary of contributions in PFL Model Optimization Techniques using Data Augmentation

Model(s) / Technique(s)	Main Idea/Contribution	Advantage(s)/Real-World Application(s)
FedAFA [53]	Generates balanced features for each client by transferring knowledge from the global model's majority class features in an adversarial example learning manner. This approach improves the performance of personalized models and addresses class imbalance in PFL.	FedAFA can enhance the generalization ability of the local minority classes while preserving the robust performance of the local majority classes.
pFedCFR [54]	Proposed a multi-layer multi-fusion technique where the server utilizes network layer parameters from each client's uploaded model as the fundamental unit for information-sharing calculations.	Improving the efficiency of heterogeneous data personalized collaboration.
Astraea [55]	Addresses imbalances through Z-score-based data augmentation and Mediator-based multi-client rescheduling. while addressing local imbalances by rescheduling client training based on Kullback-Leibler divergence (KLD). Astraea aims to improve fairness and performance in FL.	It reduces global imbalances through adaptive data augmentation and downsampling.

the training data by employing adaptive data augmentation and downsampling techniques based on the Z-score, focusing on addressing global imbalance. Additionally, it tackles local imbalance by introducing a mediator that reschedules client training using the Kullback-Leibler divergence (KLD) of their data distribution.

In [58], authors introduced ChannelFed as a PFL method that focuses on personalizing the channel attention module, emphasizing how channel attention can exploit knowledge from diverse data sources. The experimental results on CIFAR-10, Fashion-MNIST, and CIFAR-100 datasets, demonstrating the superior performance of ChannelFed compared to other PFL methods in scenarios involving statistical heterogeneity. Another study reveals a significant decrease in the accuracy of FL models when trained on highly skewed non-IID data, where each client client is limited to training on a single class of data. The decrease in accuracy is attributed to weight divergence, which is quantified using the earth mover's distance (EMD) between the class distribution on each client and the population distribution. To address this challenge, the authors propose a strategy to improve training on non-IID data by introducing a small subset of data that is shared globally among all edge clients. By incorporating this globally shared data, the accuracy of FL models can be increased [59].

In [60], authors propose a generative convolutional autoencoder (GCAE). It aims to refine model by generating a class-balanced dataset from each client's personal data, thereby facilitating accurate and personalized health monitoring. Moreover, GCAE is lightweight, allowing efficient transfer between the cloud and edge clients and reducing communication cost associated with FL in FedHome framework.

2.1.6. Regularization

In PFL, regularization techniques are used to mitigate overfitting and improve the generalization performance of the learned models. Regularization helps prevent the models from becoming too specific to the training data and encourages them to capture more general patterns that can be applied to unseen data. The regularization techniques can be applied within each local model during the training process in PFL.

Algorithm 9 Algorithm for Regularization in PFL

Input(s):
Federated dataset $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_n\}$
Regularization parameter λ
Number of training epochs T
Model architecture $f(\cdot, \cdot)$
Output(s):
Global model M
Algorithm:

Initialize global model M with random weights
for $t = 1$ to T **do**
  for each client $i = 1$ to n in parallel **do**
    Receive global model M from the server
    Randomly sample a batch of data $\mathcal{B}_i$ from $\mathcal{D}_i$
    Update local model M_i using gradient descent
    Send updated local model M_i to the server
  end for
  Aggregate and update global model M on the server:
    $M \leftarrow \text{Aggregate}(\{M_1, M_2, ..., M_n\})$
end for
return Global model M

Authors [70], in their research introduced ModFed, a novel model-based federated learning framework that addresses these challenges. In this approach, model-driven neural networks are utilized to reduce the dependence on large client-side datasets, resulting in more efficient learning. The framework introduces an adaptive dynamic aggregation scheme to handle data het-

Table 6: Summary of contributions in PFL Model Optimization Techniques using Regularization

Model(s) / Technique(s)	Main Idea/Contribution	Advantage(s)/Real-World Application(s)
FedSLR [61]	A two-stage proximal-based algorithm for FL that efficiently searches for mixed models. It combines sparse and low-rank representations to optimize the model.	reduces the number of parameters, and lowers the down-link communication complexity.
FedABC [62]	Adopts one-vs-all training strategy to address unfair competition between classes in each client. FedABC incorporates a personalized binary classification loss that combines under-sampling and hard sample mining strategies.	Mitigates class imbalance challenges, This framework promotes fairness and improves the accuracy of personalized models in PFL.
Elastic Transfer [63]	An approach for social robots that combines FL and Continual Learning. Captures interaction dynamics across multiple robots and encounters while prioritizing privacy and personalization.	Improves robot performance
CD2-pFed [64]	Introduced channel decoupling, a channel-wise assignment strategy, and utilizes cyclic distillation to guide the process. This approach enables fine-grained personalization and improves performance.	Accurate and generalized results for various types of heterogeneity, including feature skew, label distribution skew, and concept shift.
CPFLViT [65]	Architecture for distributed and heterogeneous COPD CT scans. Involves partially personalizing certain heads in the self-attention layers of ViT to capture personalized attention patterns.	Identifying COPD, potential advancements in personalized medical imaging analysis
CCVR [66]	Classifier calibration, utilizing virtual representations sampled from an approximated Gaussian mixture model (GMM). It aims to adjust the classifier by incorporating virtual representations, which are generated based GMM.	Improves the performance of PFL in diverse heterogeneity settings.
GRP-FED [67]	Through adaptive aggregation, the global model ensures fair treatment of multiple clients and effectively mitigates the global long-tailed issue.	GRP-FED demonstrates improvements in both global and local scenarios using real-world datasets, including ECG analysis and the CIFAR-10.
Private Mean-Regularized MTL [68]	An algorithm for mean-regularized MTL, an objective commonly used for applications in PFL, subject to JDP.	Improved privacy/utility trade-offs relative to global baselines.
Fed+ [69]	Provides convergence guarantees for both convex and non-convex loss functions. It is equipped to handle heterogeneous computing environments.	Improved Convergence and addressed Heterogeneity.

erogeneity, leading to enhanced generalization capabilities and improved robustness of the trained models. Moreover, ModFed incorporates a spatial Laplacian attention mechanism and personalized client-side loss regularization techniques, enabling the system to capture fine-grained information and achieve accurate image reconstruction.

Existing PFL algorithms primarily focus on model-centric approaches, neglecting the unique data characteristics of individual clients. Authors in [71], introduced a novel PFL framework called pFedPT, specifically tailored for image classification tasks. pFedPT incorporates personalized visual prompts to implicitly capture local data distribution information and integrates it into the aggregation process to improve classification performance. In the pFedPT framework, each client generates a personalized visual prompt that encapsulates information about the local data distribution during each training round. The local model is trained using both the raw data and the personalized visual prompt, enabling it to learn and encode the distribution information represented by the prompt. During model testing, the aggregated model benefits from the prior knowledge of data distributions obtained through the prompts.

The authors in [61], proposed a personalized model framework for PFL that is based on low-rank and sparse decomposition. The framework comprises two stages: (1) the extraction of a low-rank global knowledge representation (GKR) with appropriate regularization, and (2) the fusion of personalized patterns using a sparse component. To efficiently search for the mixed models by optimizing both the GKR and sparse component simultaneously, they introduce a two-stage proximal-based algorithm named FedSLR. The proposed framework effectively reduces the number of parameters and minimizes down-link communication complexity, making it a desirable choice for FL

algorithms. To further enhance collaboration between private and shared weights in the model, a cyclic distillation scheme CD2-pFed is proposed by [64]. Their scheme ensures consistent regularization between local and global model representations throughout the FL process. Another research study introduces a formulation of the federated multitask learning (FMTL) problem using Laplacian regularization, applicable to both conventional FL and PFL. The authors propose two algorithms for solving the FMTL problem, offering improved convergence rates and sublinear speedup [72].

In [66], the authors proposed a novel algorithm; Classifier Calibration with Virtual Representations (CCVR) to adjust the classifier using virtual representations sampled from an approximated Gaussian mixture model. Experimental results demonstrate that CCVR achieves state-of-the-art performance on popular FL benchmarks. The paper's contributions include experimental insights into layer representations and the introduction of CCVR as an effective algorithm for classifier calibration in the presence of non-IID data. In another study the authors discussed the potential of FL in predicting drug-related properties and addresses the challenges associated with small and biased data in drug discovery. By harnessing distributed data sources while upholding data privacy, FL has the potential to significantly enhance the success rate of AI-powered drug discovery pipelines [73].

In [74] the authors introduced GIFAIR-FL, a novel framework that integrates group and individual fairness principles into FL. The framework includes a regularization term that penalizes the spread in the loss of client groups, encouraging the optimizer to converge to fair solutions. GIFAIR-FL is designed to be applicable in both global and personalized FL scenarios. The paper provides theoretical analysis demonstrating the convergence properties of GIFAIR-FL in both nonconvex and strongly convex settings. The convergence guarantees hold for both independent and identically distributed (IID) and non-IID. data, enhancing the versatility of the framework. To assess the practical performance of GIFAIR-FL, the authors apply it to image classification and text prediction tasks.

2.1.7. Adversarial training

Adversarial training (AT) can be used as a technique for model optimization in PFL to enhance the privacy and robustness of the models. Adversarial training involves the use of an additional model called the adversary or the critic. The adversary's objective is to differentiate between the updates generated by the local models of individual clients and the updates generated by the global model. By doing so, the adversary aims to identify any potential information leakage from the updates. The algorith 10 shows a high-level overview of how adversarial training can be incorporated into PFL.

The PFL framework shown in figure. 7, on Long-Tailed Data via Adversarial Feature Augmentation (FedAFA) proposed by; [53] successfully addresses the challenges of data heterogeneity and long-tail distribution in PFL, resulting in improved personalized performance for each client. The authors in [75], focused on scenarios with a fixed communication budget and non-i.i.d. data distribution among the agents. The authors observe a significant decrease in both natural and adversarial accuracies when applying AT in the federated setting compared to centralized training. They attribute this drop to two factors: the drift between local models caused by the number of AT epochs performed locally, and the increased convergence time measured in communication rounds.

Algorithm 10 Adversarial Training in PFL

```
Initialize global model G
Initialize adversary model A
Initialize learning rate η
Initialize number of iterations T
for t = 1 to T do
    Distribute G to clients
    for each client c do
        Generate client update U_c based on local data
    end for
    Update adversary model A using updates U_c
    for each client c do
        Compute adversarial loss L_adv = A(U_c)
        Update G using U_c and L_adv with learning rate η
    end for
end for
```

AT has been widely adopted to enhance the robustness of deep neural networks against adversarial attacks. However, recent studies have shown that AT can inadvertently introduce vulnerabilities to privacy attacks. In [77], authors delved deeper into this unsettling property of AT and introduces a novel privacy attack specifically targeted at FL systems, which are particularly sensitive to privacy concerns. Through their proposed method, an attacker can exploit AT models within the FL system to accurately reconstruct clients' private training images, even when the training batch size is large.

In [78], authors introduced a semi-centralized adversarial training approach combined with the use of a Variational AutoEncoder (VAE) in FL to address concerns related to discrimination and enhance group fairness. By incorporating sensitive attribute alignment and preserving privacy through the VAE architecture, the proposed method demonstrates improved performance compared to SOTA FL frameworks. In [76], authors proposed a FL method that combines co-training and generative adversarial networks (GANs) shown in figure. 8. By leveraging co-training and GANs, the proposed method enables clients to participate in FL with tailored models.

2.1.8. Meta-learning

Meta-learning, also referred to as "learning to learn," is a research field dedicated to enhancing learning algorithms by exposing them to diverse tasks or datasets. The objective is to enable the model to acquire new tasks rapidly and effectively. Optimization-based meta-learning is one approach within this field, encompassing algorithms such as Model-Agnostic Meta-Learning (MAML) and Reptile. MAML and Reptile are known for their capacity to generalize well and adapt swiftly to novel

Table 7: Summary of contributions in PFL Model Optimization Techniques using Adversarial training

Model(s) / Technique(s)	Main Idea/Contribution	Advantage(s)/Real-World Application(s)
FedAFA [53]	optimizes personalized models through the transfer of knowledge from the local majority class features, which are extracted by the global model using adversarial example learning techniques. Effectively addresses class imbalances and improves the performance.	It addresses class imbalance and improves the performance of personalized models in PFL.
PrivayAttackATFL [77]	identify a privacy vulnerability in adversarial training (AT) and demonstrate a practical attack in FL systems. The attack allows for the accurate reconstruction of clients' private training images, even with a large batch size.	Feature restoration from gradients, Image reconstruction from feature, Robust privacy protection in FL systems.
Fairscat [78]	Utilizes VAE in FL scenarios to generate adversarial samples. The VAE decoder remains on the server side, while the encoder remains on the client side to encode local samples into feature dimensions for transmission, ensuring client data privacy.	This approach combines privacy preservation, adversarial training, and attribute alignment to address fairness concerns.
pFedDef [79]	A defense mechanism that enhances adversarial robustness in PFL while considering resource limitations at client clients that may hinder adversarial training. Proposed approach significantly improves relative grey-box adversarial robustness, achieving a 62% increase.	Performs effectively even when clients have limited system resources, making it a practical solution for improving the security.
PerFED-GAN [76]	A method that leverages co-training and generative adversarial networks (GANs) to enable each client to independently design its own model for participation in FL training. By combining co-training and GANs, this method facilitates collaborative learning among clients while maintaining privacy and independence.	This approach allows for personalized model development within the FL framework, enhancing the flexibility and privacy of the learning process.
pFLSynth [80]	pFLSynth utilizes an adversarial model to generate site-specific and source-target contrast-specific latents. It uses novel personalization blocks that dynamically adjust the statistics and weighting of feature maps across the generator stages.	Enhancing the reliability and performance of MRI synthesis in FL settings.

and heterogeneous tasks. In conclusion, optimization-based meta-learning algorithms like MAML and Reptile have proven their effectiveness in enhancing the learning capabilities of models by facilitating rapid adaptation to new tasks and promoting generalization across diverse datasets [81, 82, 83].

In [90], the authors establish a connection between meta-learning and FL by emphasizing the similarities in their formulations. They specifically draw a parallel between the phases of meta-learning, namely metatraining and meta-testing, and different aspects of the FL process. The metatraining phase in meta-learning involves training a model on various tasks to enhance its learning capabilities. The authors relate this phase to the FL global model training process, where a global model is trained using data from multiple clients. The objective is to improve the performance of the global model across diverse clients and tasks. Similarly, the meta-testing phase in meta-learning entails quickly adapting the model to a new task through a few steps of gradient descent. The authors connect this phase to the FL personalization process, where the global model undergoes further fine-tuning on local data during the local adaptation phase on each client. This step aims to personalize the global model according to the specific data distribution and requirements of each client. Additionally, the authors demonstrate that the FL algorithm FedAvg is analogous to the Reptile algorithm from meta-learning. The authors emphasized the potential for integrating meta-learning approaches to improve the FL framework.

The Algorithm 11 takes the set of clients, C, and the meta-training data, $\mathcal{D}_{\text{meta}}$, as input. The output of the algorithm is the meta-trained model parameters, θ. The "Output" statement at the end clarifies the algorithm's output.

In [88], the authors extend the Per-FedAvg approach and propose a federated meta-learning framework called pFedMe (Personalized Federated Meta-Learning) that utilizes Moreau envelopes. The objective of pFedMe is to find a balance between personalization and generalization performance in FL.

Table 8: Summary of contributions in PFL Model Optimization Techniques using Meta-learning

Model(s) / Technique(s)	Main Idea/Contribution	Advantage(s)/Application(s)
HAM [84]	A Hierarchical Attention-enhanced Meta-learning Network, for PFL. It utilizes a meta-learning approach to analyze similarities among clients and strike a balance between individuality and common knowledge.	reasonably achieves a trade-of between clients' personalities and commonality
MFPCDR [85]	Personalized cross-domain recommendation model that leverages a server-side meta-recommendation module to uncover personalized preferences for cold-start clients. To ensure client privacy, clients employ an attention mechanism to identify transferable features for knowledge transfer, generating client and item embeddings.	Effectively addresses both user privacy concerns and the user cold-start problem.
G-FML [86]	A Group-based Federated Meta-Learning framework that dynamically groups clients based on data distribution similarity. Within each group, personalized models are learned using meta-learning.	This approach enables effective personalization in highly heterogeneous environments.
Per-FedAvg [87]	A personalized version of FL aimed at finding an initial shared model that can be easily adapted by clients to their local datasets through a few steps of gradient descent.	This approach leverages the benefits of FL while promoting personalized models.
pFedMe [88]	It separates the optimization of personalized models from the global model learning, which improved convergence rates. The algorithm achieves quadratic speedup for strongly convex objectives and sublinear speedup of order 2/3 for smooth nonconvex objectives.	state-of-the-art convergence speedup rate.
ARUBA [89]	Integrates task-similarity formalizations with online convex optimization and sequential prediction algorithms for designing and analyzing meta-learning methods. It allows adaptive learning of task-similarity, improves transfer-risk bounds, and enables the derivation of average-case regret bounds.	Improves the meta-test-time performance on standard problems
MAML [90]	Practical applications of MAML in FL, including the interpretation of Federated Averaging as a meta learning algorithm are discussed. Models trained using standard datacenter optimization methods are more challenging to personalize compared to those trained with Federated Averaging.	Better personalized accuracy, suggesting need a novel way to predict the generalization of personalized models.

Algorithm 11 Meta-Learning for PFL

Input(s): Set of clients C, Meta-training data $\mathcal{D}_{\text{meta}}$
Output(s): Meta-trained model parameters θ
Algorithm:

Initialize meta-model parameters θ
for meta-epoch = 1 to $N_{\text{meta-epochs}}$ **do**
  for each client $c \in C$ **do**
    Sample task-specific data $\mathcal{D}_c$ from client c
    Initialize client-specific model parameters $\phi_c \leftarrow \theta$
    for local-epoch = 1 to $N_{\text{local-epochs}}$ **do**
      Compute loss $L_c(\phi_c, \mathcal{D}_c)$
      Update client-specific model parameters
    end for
    Update meta-model parameters
  end for
end for

To achieve this, pFedMe introduces an l2-norm regularization loss term into the optimization objective. This regularization term allows for controlling the trade-off between personalization and generalization. By adjusting the strength of the regularization, one can control the degree to which the model is personalized to individual clients while still maintaining generalization across clients. The pFedMe algorithm demonstrates better convergence and accuracy as compared to the FedAvg and Per-FedAvg methods. It leverages the concept of Moreau envelopes, which provides a smooth approximation of the non-smooth regularization term. This facilitates efficient optimization and enhances performance in federated meta-learning scenarios. Overall, pFedMe expands on the Per-FedAvg approach by incorporating an l2-norm regularization loss and leveraging Moreau envelopes. This formulation enables better control over the balance between personalization and generalization, resulting in improved accuracy and convergence in FL settings. In

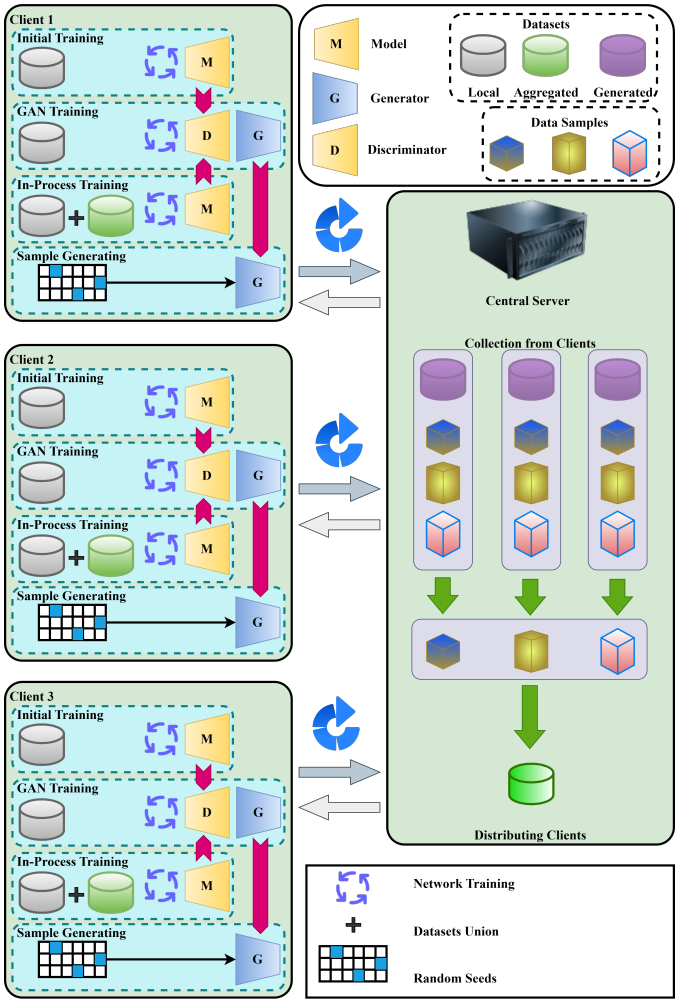

Figure 8: Framework of PerFED-GAN [76].

their work, Khodak et al. [89] introduced the ARUBA framework, which focuses on achieving adaptive meta-learning in the context of FL using online learning techniques. By incorporating ARUBA with the FedAvg algorithm, the authors observed enhancements in the generalization performance of the models and eliminated the requirement for hyperparameter optimization during the personalization process. The ARUBA framework offers a promising approach to improve the adaptability and performance of meta-learning in FL scenarios.

2.1.9. Clustering

Clustering involves grouping similar participants into clusters to improve the efficiency and effectiveness of the FL process. In FL, data is distributed across multiple clients, and each client trains a local model using its own data. In PFL, each client trains a personalized model for its client. However, training personalized models can be computationally expensive and may require a large amount of communication between clients. Clustering can help address this issue by grouping clients that have similar data or similar personalized models. By doing this, clients in the same cluster can share information and learn from each other, reducing the amount of communication required and improving the overall efficiency of the FL process. Clustering can be done using various techniques, such as k-means clustering, hierarchical clustering, or density-based clustering. The choice of clustering algorithm depends on the characteristics of the data and the specific requirements of the FL task. Algorithm 12 shows the implementation of k-means clustering in PFL.

The task of clustering clients with similar objectives and training a model for each cluster is a popular approach in PFL. However, achieving provable and optimal guarantees for this approach has been an ongoing challenge. In their work, Werner et al. [91] address this challenge by formulating PFL as a stochastic optimization problem, where the stochastic gradients from a client can belong to one of K distributions. The authors demonstrate that by employing a simple thresholding-based clustering algorithm and utilizing local client momentum, optimal convergence guarantees can be achieved in this stochastic optimization setting. This work provides insights into effectively clustering clients in PFL while ensuring optimal performance.

Algorithm 12 k-means Clustering in PFL

Input(s):
Personalized model updates from each client
Number of clusters and Maximum number of iterations
Output(s):
Cluster assignments, Final centroids for each of the k clusters, Number of iterations
Algorithm:

1: Initialize k centroids randomly
2: **repeat** step 3 and 4
3: Assign personalized model update to nearest centroid
4: Update each centroid to the mean of the personalized model updates in its cluster
5: **until** convergence or maximum number of iterations
6: Aggregate personalized model updates in each cluster
7: Communicate the cluster-level model updates to client
8: Repeat steps 1-7 until desired level of accuracy is achieved

In their paper, Yoo et al. [92], introduced Personalized Federated Cluster Models, a hierarchical clustering-based process for FL, specifically applied to predict the severity of Major Depressive Disorder using Heart Rate Variability data. The proposed approach addresses the challenges posed by non-Independent and Identically Distributed (non-IID) data commonly encountered in medical settings. By allowing clients to receive more personalized models through the clustering process, the authors observed an improvement in accuracy for severity prediction. This performance gain suggests that Personalized Federated Cluster Models hold promise for various FL scenarios, including those involving medical data where the assumption of IID data is often unrealistic. There are several algorithms that can be used for clustering in PFL.

In [93], the authors presented a novel framework called PerFed-CKT for PFL in their paper. PerFed-CKT enables clients to utilize heterogeneous model architectures and avoids direct transmission of model parameters. Instead, the framework employs clustered co-distillation, where clients transfer knowledge using logits to other clients with similar data distributions. Through empirical evaluation, the authors demonstrated that PerFed-CKT achieves high test accuracy while significantly reducing communication costs compared to state-of-the-art personalized FL schemes. In their paper, Armacki et al. [94] introduced a novel approach for PFL that enables automatic model clustering without prior knowledge of the hidden cluster structure or the number of clusters. The authors provide analytical bounds on the weight parameter, which allows for simultaneous personalization, generalization, and automatic model clustering. This approach offers the flexibility of providing different models across different clusters for personalized learning, while also ensuring generalization by providing models that differ from the per-client models computed in isolation. To address the formulated problem within a federated server-clients setting, the authors propose an efficient algorithm based on the Parallel Direction Method of Multipliers (PDMM). The paper by Jie et al. [95], proposes a federated recommendation system that uses historical parameter clustering to improve personalization. Clients combine their historical learning parameters with global parameters received from the server using a weighted average. The server aggregates and clusters the parameters to identify client groups with similar preferences. To address limited raw data availability, the authors introduce a recommendation model based on client embedding features. The server leverages these features for personalized recommendations without accessing individual data, while clients use their original data locally. In their paper Li et al. [102]address the challenge of optimizing the accuracy and cost trade-off in FL by considering model accuracy, communication resource allocation, and energy consumption. They propose an iterative solution procedure that incorporates parameter encryption techniques and employs a deep reinforcement learning approach at the cloud server for edge association. The objective is to minimize energy consumption at each base station while maximizing the averaged model accuracy of all clients. The proposed approach utilizes a reward function that combines energy consumption and model accuracy to guide the learning process. By jointly considering these factors, the authors aim to improve the efficiency and effectiveness of FL in energy-constrained environments.

In the research work, by Lyu et al. [106], an intuitive approach for PFL by incorporating parameter regularization and cluster communication is proposed. The authors suggest regularizing the model parameters in a way that encourages clients within the same cluster to share similar weights. This regularization promotes similarity and coherence among the models of clients within each cluster. Simultaneously, the distances between different clusters are also regularized to reflect the dissimilarity between clusters, capturing the diversity among different groups of clients. The work presented in [106] offers an innovative perspective on PFL by leveraging parameter regularization and cluster communication. Their algorithm demonstrates improved convergence properties, surpassing independent learning and shared weight learning.

In their paper [98], the authors propose a novel framework called BPFL (Blockchain-Enabled Personalized Federated Learning) that combines the advantages of blockchain technology and edge computing for enhanced privacy, security, and efficiency in PFL. The integration of blockchain technology in BPFL provides several benefits. Firstly, it enhances client privacy and security by leveraging the immutability and transparency of distributed ledger networks. Transactions and interactions between clients and the PFL system are recorded on the blockchain, ensuring data integrity and preventing unauthorized access. This contributes to building trust among participants and maintaining the confidentiality of sensitive information. In the research paper by Cho et al. [97], a novel and practical framework for PFL called COMET is introduced. This framework addresses the challenge of heavy communication overhead in PFL by allowing clients to use their own heterogeneous models and avoiding direct communication of model parameters. COMET employs a technique called clustered codistillation, where clients utilize knowledge distillation to transfer their learned knowledge to other clients who have similar data distributions. By leveraging knowledge transfer within clusters, COMET offers a practical solution for training edge clients in IoT networks,

Table 9: Summary of contributions in PFL Model Optimization Techniques using Clustering

Model(s) / Technique(s)	Main Idea/Contribution	Advantage(s)/Application(s)
CFedPer [96]	Consists of a pre-start phase for client clustering and an in-training phase with a base layer and a personalization layer.	Addresses the limitations of FL
COMET [97]	Allows clients to use their preferred heterogeneous models without directly sharing model parameters. It employs clustered co-distillation to transfer knowledge.	An efficient PFL solution for training edge clients via IoT networks Reduces communication overhead.
BPFL [98]	Integrates the benefits of edge computing and blockchain.	Enhanced privacy, improved real-time services and communication.
HPFL-CN [99]	Employs clustering to extract privacy-preserving feature representations and incorporates an edge-mediator-cloud architecture with Effective Hierarchical Scheduling (EHS).	Efficient model aggregation, Flexibility, Resource Efficiency
PerFedRec [100]	It learns joint representations using a federated GNN, clusters clients based on these representations, and trains personalized models for each cluster.	Reduces communication costs, suitable for applications with limited band-width and low latency.
FedCHAR-DC [101]	Improve model accuracy and fairness by leveraging the similarity between clients. It features dynamic clustering and adaptation to new clients or evolving datasets in FL-based HAR scenarios.	Enhancing system robustness by identifying malicious nodes in attack scenarios
Historical parameter clustering [95]	Addresses problem of non-IIDness, and proposed recommendation system based on clustering of historical parameters.	Provides a method to solve the non-IID problem in FL.
DRL [102]	This approach utilizes deep reinforcement learning for edge association and minimizes energy consumption while maintaining high model accuracy.	Optimization of model accuracy, energy consumption and communication resource allocation.
FedPRC [103]	Incorporates a local outlier factor-based anomaly detection to identify outliers. Formulated a nested bi-level optimization.	Detection of anomaly clients or outliers.
FedGroup [104]	Employs data-driven distance measure to cluster, allowing clients with similar optimization features to be grouped.	Improved training performance and incorporate new clients seamlessly.
PFCM [92]	Tailoring the models to each individual, the approach seeks to enhance the accuracy of predicting the severity of Major Depressive Disorder using Heart Rate Variability data.	Enhance the performance of federated training.
ClusterGrad [105]	Leverages the observation that, only small fraction of gradients have values that are significantly different from zero.	Significantly reduces the volume of computations communicated.

where communication resources are limited.

In the research conducted by Sattler et al. [107], hierarchical clustering is employed as a post-processing step in Federated Learning (FL). The authors utilize a bi-partitioning algorithm based on cosine similarity of gradient updates to divide FL clients into clusters. However, the recursive nature of this bi-partitioning clustering framework necessitates multiple communication rounds to separate dissimilar clients, resulting in high computation and communication costs. This limits the practical feasibility of the approach in large-scale settings. In a anoother study conducted by Briggs et al. [108], a distinct hierarchical clustering framework for Federated Learning (FL) is proposed. The procedure entails training a global FL model for a specific number of communication rounds, followed by fine-tuning the global model on the private datasets of all clients to calculate the parameter difference, denoted as Δw, between the global and local model parameters. These computed Δw values for all clients are utilized as inputs to the agglomerative hierarchical clustering algorithm, which generates multiple client clusters. This approach is specifically designed to handle a wider range of non-IID settings and enables training on a subset of clients during each round of FL model training.

The study [109] introduces the Iterative Federated Clustering Algorithm (IFCA) as a novel approach. In contrast to traditional methods that rely on a single global model, IFCA employs K global models, which are shared among all clients for local loss computation. Subsequently, the server performs cluster-based aggregation of the Federated Learning (FL) models within the partitioned clusters. This iterative clustering approach enhances collaboration and the learning process among clients, while simultaneously ensuring model diversity and scalability within the federated learning framework. Huang et al.

[110] propose a community-based Federated Learning (CBFL) approach for predicting patient hospitalization time and mortality. Following the clustering, separate Federated Learning (FL) models are trained for each cluster. Duan et al. [104], introduce FedGroup, an FL clustering framework that incorporates a static client clustering strategy and a newcomer client cold start mechanism. FedGroup utilizes the K-means++ algorithm [111].

In the work by Li et al. [99], a novel framework called HPFL-CN (Hierarchical Personalized Federated edge Learning via Complex Network feature clustering) is proposed. HPFL-CN is designed to cluster edge servers based on similar environmental data distributions and to train personalized models efficiently for each cluster using a hierarchical architecture. The objective of this framework is to address the communication and computation overhead in FL and enable more efficient and effective PFL in the context of urban environmental prediction tasks within Mobile Edge Computing. In their paper, Yin et al. [40] introduced a technique called growing semiorganizing neural gas (GsONG) for personalized and inclusive differentiated learning. GsONG combines the searchability of evolutionary computation with the learning capability of a hybrid artificial neural network approach. It utilizes an adaptable neural network architecture composed of competing and cooperating neurons that operate in an unstructured mode. Through a cooperation-competition process, a topological neighborhood of neurons in a grid is established, allowing for the identification of patterns whose classes are initially unknown.

3. Limitations

In our analysis and comparison of the discussed techniques, we have identified certain limitations. Parameter decoupling methods, as implemented in studies such as Arivazhagan et al. [13] and Bui et al. [14], provide a straightforward approach by introducing personalized layers for each client. However, these methods have limitations in supporting extensive personalization of model design. On the other hand, knowledge distillation (KD)-based PFL methods offer more flexibility in terms of personalized model architectures and are advantageous in communication and computation-constrained edge FL settings, as demonstrated in works like He et al. [112] and Bistritz et al. [113]. However, the KD process often requires a representative proxy dataset, which can be challenging to obtain.

Parameter decoupling requires careful classification of private and federated parameters to strike a balance between generalization and personalization performance. Arivazhagan et al. [13] note the importance of finding optimal strategies for parameter privatization, which remains an ongoing research challenge in the field of PFL. In the case of KD, the effectiveness of knowledge transfer is not solely dependent on model parameters but also on the model architecture itself. Large capacity gaps between the teacher and student models can impede the learning process, as highlighted by Liu et al. [114] and Li et al. [115]. This emphasizes the need for optimal design choices for both server and client models in order to facilitate efficient knowledge transfer in PFL. Parameter decoupling and KD methods encounter difficulties in model building in PFL. The challenges include finding optimal strategies for parameter privatization in parameter decoupling and making appropriate design choices for the server and client models in KD to overcome capacity gaps. In PFL, when the local data on each client is limited, it becomes challenging to train accurate mentor models. It is important to consider the computational cost imposed on local clients during the FedKD process [33].

PFL operates in a distributed setting with privacy constraints, where each client has its own data and task-specific requirements. The limitation lies in finding a suitable NAS approach that can effectively navigate the search space while considering the distributed nature of PFL [39]. Researchers are actively investigating the integration of NAS with PFL to overcome these limitations and leverage the benefits of automated architecture search in PFL scenarios [44, 116].

Hyperparameter optimization requires frequent communication and coordination between the clients and the central server and limited communication and computation resources become a limitation of this technique. another limitation of hyperparameter optimization is; heterogeneity which makes it difficult to perform optimization, as the optimal hyperparameters for one client may not generalize well to others. It also involves exchanging information about model architectures, training progress, and performance metrics which is another limitation of this technique [48].

Some data augmentation techniques, introduce distortions that may not accurately represent the real-world data. In certain cases, this can lead to overfitting or biased model predictions. Data augmentation techniques may not always be applicable or effective for all types of data or domains. Different data modalities, such as text, images, or time series, may require specific augmentation methods tailored to their characteristics. Finding domain-specific augmentation strategies is still a limitation of data augmentation [58]. Although this technique is often used to address data imbalance issues by generating synthetic samples for underrepresented classes. However, if the original data distribution is highly imbalanced, augmentation alone may not fully resolve the problem, and other techniques such as class weighting or data resampling may be necessary [55]. Data augmentation can significantly increase the computational requirements during training, as each augmented sample adds to the overall training workload. This can lead to longer training times and increased resource consumption, particularly in large-scale or resource-constrained settings [117].

Many regularization techniques involve hyperparameters that need to be manually tuned, such as the regularization strength or dropout rate. The performance of the model can be sensitive to the choice of these hyperparameters, and finding the optimal values is one of the limitations of works proposed so far. Improperly tuned hyperparameters can lead to under- or over-regularization, resulting in suboptimal model performance. Another limitation of existing works is to find the right level of regularization that achieves this balance. Excessive regularization can lead to high bias and an underfit model, while insufficient regularization may result in overfitting. Regularization techniques primarily focus on preventing overfitting to the train-

ing data. However, they may not effectively handle outliers or corruptions in the data during training. Outliers or corrupted data points can disproportionately influence the regularization process, leading to biased models or poor generalization performance [118]. Regularization techniques alone may not be sufficient to address inherent biases present in the training data, especially in scenarios where the data is highly imbalanced or contains systematic biases [62, 70].

In adversarial training the process of learning becomes more computationally intensive, which can increase the training time and resource requirements, especially in large-scale PFL settings. The effectiveness of adversarial training depends on the quality and strength of the adversary model. Designing and training a robust adversary model that can accurately distinguish between local and global updates is challenging. If the adversary model is not well-designed or is weak, it may not effectively detect potential information leakage, limiting the privacy-enhancing benefits of adversarial training. Adversarial training aims to improve privacy by making it difficult for the adversary to differentiate local and global updates. However, this may come at the cost of sacrificing model performance. Adversarial training may be vulnerable to novel and sophisticated attacks that exploit weaknesses in the training process. These methods need to be continuously updated and improved to address emerging privacy threats and adapt to evolving attack strategies [77]. The diversity of PFL scenarios and adversaries makes it challenging to establish a one-size-fits-all adversarial training framework. This lack of standardization may result in inconsistencies and variations in the effectiveness and practicality of different adversarial training methods [78, 80]. PFL often involves clients with non-IID (non-identically distributed) data, meaning the data distribution across clients may vary significantly. Adversarial training techniques may struggle to handle such data heterogeneity, as the adversary model may not be able to effectively distinguish between updates from different clients with distinct data distributions, potentially leading to reduced privacy guarantees [119].

Meta-learning approaches often require a large amount of data from diverse tasks or domains to effectively learn generalizable representations. In PFL, where data is distributed across multiple clients, accessing a diverse range of tasks or domains can be one of the limitations of Meta-Learning, limiting the scalability of meta-learning techniques [84]. Meta-learning techniques may struggle to effectively capture and leverage the heterogeneity present in the data across clients. In PFL, where computation and communication resources are limited, the computational complexity of meta-learning techniques may pose challenges in terms of time and resource efficiency [85]. Meta-learning often requires the aggregation of information across multiple clients or tasks, which can raise privacy concerns. Meta-learning models can be highly complex and difficult to interpret, making it challenging to understand the underlying mechanisms and decision-making processes. In PFL, interpretability is crucial for ensuring transparency and trust in the learning process, especially when dealing with sensitive data.

Clustering methods are beneficial when inherent client partitions exist. However, they often come with high computation and communication costs, making them impractical for large-scale settings, as discussed in Sattler et al. [107]. Moreover, additional architectural components are required for managing and deploying the clustering mechanism, as highlighted in Briggs et al. [108]. For instance, when dealing with a large label space like the Glink360 K dataset, the COMET approach may become less communication efficient. Additionally, incorporating a public dataset that is unrelated to the task of interest may not enhance the generalization performance of the clients [97]. Other challenges include data scarcity, communication overhead, privacy concerns, heterogeneity, and label noise. These factors can impact the effectiveness and efficiency of the federated learning process.

4. Challenges & Open Problems

The Table 10 provides an overview of different models and methods used in personalized federated learning (PFL), along with the associated challenges and open problems and future directions for each approach. Future research in PFL should address the issues disscussed in the section III and explore innovative approaches to improve model interpretability, handle data heterogeneity and distribution shifts, address privacy concerns, optimize communication efficiency, handle dynamic client participation, and validate knowledge discovery techniques in real-world scenarios.

Approaches parameter de-coupling involve introducing personalized layers for each client in order to support model design personalization. However, these techniques face challenges e.g. as non-IID data, security and privacy concerns, increased communication overhead, and difficulty in global model coordination. Future directions include improving communication efficiency, supporting heterogeneity and non-IID data, enabling adaptive decoupling, and addressing robustness and fault tolerance issues. Whereas Knowledge Distillation (KD) based methods allow flexible model architectures and are advantageous in communication and computation-constrained edge FL settings. However, they have limitations in terms of model interpretability, distribution shifts, privacy and security, and communication issues. Future directions involve exploring KD in heterogeneous and healthcare applications [29], combine MetaFed with common methods such as FedAvg, to implement a complete FL system, including intra- and inter- federations. Apply MetaFed for heterogeneity architectures and more realistic healthcare applications [31], rigorous analysis along this line [35], there is a need to explore how to train large models on low-resource clients to support the application of FedKD in cross-client settings. Deployment of FedKD in real-world personalization systems can also be explored to learn intelligent client profiling models that serve clients privacy-preservation [34]. In terms of distillation-based personalization, future work can extend it to unsupervised learning tasks where the distance measurement component does not require class labels, as long as the local loss function does not include target labels in its metric. Another interesting research problem is the exploration of topologies and connectivity graphs that consider physical distances among edge clients [120].

Table 10: PFL Model Optimization Techniques, Limitations and Open Problems

Model(s) / Method(s)	Challenges(s)	Open Problem(s)/Future Direction(s)
Parameter de-coupling	Privacy, Communication overhead, Global model coordination and Interpretability [20].	Communication efficiency, Heterogeneity and Non-IID support, Representation learning in non-linear settings [18], Optimization strategies [25].
Knowledge Distillation (KD)	Distribution shifts, Privacy, security, and Communication issues, Real-world problems Generalization, difficult to learn models, computational cost.	Fine-grained tuning strategy [29], KD in healthcare and multi-task learning [31, 32], Integration with other FL techniques [33], Communication-efficiency [34], Personalization with unsupervised learning [120], Local loss minimization [**?**]
Neural Architecture Search (NAS)	Heterogeneity, Non-IID data, Communication overhead, Resources, Adversarial NAS [116]	Combining other un/semi-supervised methods [41]. NAS for heterogeneous clients, real-time applications [44], To validate for tasks other than image classification,
Hyperparameter Optimization (HPO)	Increased computation complexity, Communication overhead, Heterogeneity, Privacy, Scalability and interpretability	To improve quality of CT imaging and achieve personalized demands [49], Privacy-preservation and Heterogeneity [51], Bayesian optimization, [52, 48].
Data Augmentation	Client-specific data distributions, Communication and Privacy concerns	Client-specific, Privacy-aware and Adaptive data augmentation evaluation, [54, 59, 60, 117, 121].
Regularization	Data scarcity, non-IIDness [62], Communication overhead, Privacy	Integration with communication strategies and experiments on more large-scale datasets [62]. Client-specific, Privacy-preserving and Adaptive regularization [63, 64].
Adversarial training	Communication efficiency, Heterogeneity, Data availability	Client-specific, Privacy-preserving and Adaptive AT, Robustness against diverse attacks [77, 78, 119].
Meta-learning	Communication overhead, Privacy concerns, Limited data	Communication issues [84], Speed, Preserving privacy [85], new methods, fairness issues [122], trust mechanisms [123].
Clustering	Generalization performance, Communication overhead, Privacy, Heterogeneity.	Methods to solve non-IID issue, representative dataset, Privacy implications [97], Clustering-based client selection convergence, fairness and balance data [109, 124]

Neural Architecture Search (NAS) techniques aim to automatically discover effective neural network architectures for PFL. Challenges include heterogeneity, non-IID data, communication overhead, and computational resource requirements. Future research directions include exploring adversarial and encrypted federated NAS, developing more efficient NAS algorithms, better integration, applying NAS to non-IID data and real-time applications, and validating for tasks beyond image classification. In the future, federated evolutionary NAS techniques can be developed which can further enhance the classification performance without significantly increasing computational costs. In addition, the proposed algorithms for real-time NAS in large-scale FL systems can be verified and extended. Furthermore new techniques remain to be investigated to deal with data that are vertically partitioned and distributed on the clients [44].

Hyperparameter Optimization (HPO) approaches focused on selecting optimal hyperparameters for training models in PFL. Challenges include increased computational complexity, communication overhead, heterogeneity across clients, privacy concerns, scalability, and interpretability. Future directions involve privacy-preserving and heterogeneity-awareness in HPO, exploring AutoML and Bayesian optimization for PFL, and addressing distributed HPO [48]. Hyperparameter transfer learning, can be extended by integrating with the existing methods such as differential privacy and multi-party computation. It can also be extended by considering more heterogeneous datasets [52].

Data augmentation techniques aim to generate additional training data for each client in PFL. Challenges include client-

specific data distributions, communication overhead, privacy concerns, and limited data availability. Future directions include client-specific and privacy-aware data augmentation, federated data augmentation, adaptive data augmentation, and privacy-preserving evaluation of data augmentation techniques. In future researchers can work on more detailed and generic fusion strategy based on pFedCFR [54]. Considering that the similarity matrix is sensitive to features and is easily affected by redundant or noise features, feature engineering [117], is also the future optimization direction [121]. Improving model training on non-IID data is key to make progress in this area [59]. FedHome can be applied to many healthcare applications without incurring data leakage and can be a powerful approach for in-home health monitoring in the future [60].

Regularization methods are used to prevent overfitting and improve generalization in PFL. Challenges include data scarcity, non-IID data challenges, communication overhead, privacy concerns, and limited data availability. Future directions involve integration with communication strategies, client-specific and privacy-preserving regularization, adaptive regularization, and regularization-aware model selection. Real-life FL datasets for specific engineering or health science applications are still scarce [74]. In [64], authors assigned a fixed personalization ratio for all layers, which yields an interesting future direction on searching a layer-specific optimal ratio. Federated Continual Learning offers a decentralized learning framework that can be extended to various domains in human-robot interaction (HRI). To further reduce communication overhead, there is potential to explore reformulations similar to those introduced by FedCurv [63]. In addition, future directions in research include extending the privacy model to cases where data subjects have multiple records across different data silos. Furthermore, it is valuable to extend theoretical characterizations to deep learning cases or conduct large-scale empirical studies to gain deeper insights into the performance and scalability of Federated Continual Learning approaches [66, 125].

Adversarial training (AT) techniques enhance privacy and robustness in PFL. Challenges include communication efficiency, privacy and heterogeneity concerns, and limited data availability. Future directions include client-specific, privacy-preserving, and adaptive AT methods, as well as addressing robustness against diverse attacks. Incorporating dynamic scheduling with more advanced model fusion techniques remains a topic of future research [75], pFLSynth proposed by [80], might also be adopted for other image translation tasks involving CT or PET, and other dense-prediction tasks such as reconstruction or super-resolution. Future studies can be done to address these issues and develop some kind of privacy-aware AT [77]. Researchers may explore the collaborative optimization scheme of fairness and accuracy, and improve the decision making performance of the model while optimizing fairness [78]

Meta-learning aims to enhance learning algorithms in PFL by enabling rapid acquisition of new tasks. Challenges include communication overhead, privacy concerns, and limited data availability. Future directions involve reducing communication issues, improved speed while preserving privacy, exploring new methods for lifelong learning, addressing privacy-preserving and fairness issues, and developing new trust mechanisms. There exist possibility of fairness issues within the the meta-learning frameworks, which can be addressed in future [122]. We see great potential for applying ARUBA to derive many other new LTL methods in a similar manner [89]. The current blockchain federated technology requires more complex calculations in future work [123]. Researchers can continue to investigate privacy-preserving solutions for cross-domain recommendations in the future in order to improve speed while safeguarding client privacy even further [85]. Researchers can also investigate model compression algorithms for the FedHAM framework to decrease problems related to communication between the server and the client [84].

Clustering methods aim to improve generalization performance in PFL by grouping similar clients. Challenges include data scarcity, communication overhead, privacy concerns, heterogeneity, and label noise. Future directions include improving clustering-based client selection convergence, proportional fairness, understanding privacy implications, developing representative datasets, solving non-IID issues, and obtaining higher quantity and balanced data. Future work directions require more elaborated experiments on real data sets, as well as further comparisons with other personalized FL approaches and practical implementations of the proposed approaches, their management and deployment mechanisms [94]. In order to advance the field of PFL, there is a need for further research in several areas. Firstly, developing a representative dataset that encompasses various types of attacks and defenses would enable a comprehensive examination of robust protocols for PFL [98]. Additionally, future directions should aim to address the limitations of the COMET framework and gain a better understanding of its privacy implications. This involves determining the optimal level of data correlation to maximize the performance of personalized models while preserving privacy [97]. Moreover, extending the analysis of federated clustering algorithms, such as the Iterative Federated Clustering Algorithm (IFCA), to weakly convex and non-convex functions, as well as considering stochastic gradients on worker machines and a small subset of participating clients, would provide valuable insights for improving the efficiency and scalability of personalized federated learning [109].

5. Conclusion

This paper provides an introduction to Personalized Federated Learning (PFL), including the fundamental definitions, related technologies, and state-of-the-art model optimization techniques. It discusses various implementation scenarios of model optimization techniques and identifies the current challenges and open problems in the field of PFL. The potential of PFL to offer secure and shared security services across different applications and contribute to the stable development of artificial intelligence is highlighted. The survey aims to serve as a valuable roadmap for researchers and practitioners interested in entering the field of PFL. It emphasizes the importance of privacy and security protection mechanisms, client cooperation training modes, fairness, and robustness as key areas of future

research in PFL. By addressing these aspects, the deployment and application of PFL technology can be further explored and advanced. Overall, the this research work provides a comprehensive overview of the different approaches used in PFL, their associated challenges, and potential future research directions. It highlights the ongoing efforts in addressing these challenges and the need for further advancements in various aspects of PFL to enable further research in this domain.

Acknowledgment

This work is sponsored by the National Key R&D Program of China under Grant number (2022YFB3103100). This work is sponsored by the R& D Program of Beijing Municipal Education Commission (KM202210005028). This work is also supported by National Natural Science Foundation of China (62302020) and the Major Research Plan of National Natural Science Foundation of China (92167102). This work is also supported by the Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions (CIT&TCD20190308) and the "Engineering Research Center of Intelligent Perception and Autonomous Control, Ministry of Education"

References

[1] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale: System design. *Proceedings of machine learning and systems*, 1:374–388, 2019.

[2] Qi Zhang, Lu Cheng, and Raouf Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1:7–18, 2010.

[3] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *arXiv: Artificial Intelligence*, 2019.

[4] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37:50–60, 2019.

[5] Viraaji Mothukuri, Reza Meimandi Parizi, Seyedamin Pouriyeh, Yan ping Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Gener. Comput. Syst.*, 115:619–640, 2021.

[6] L. Lyu, Han Yu, Xingjun Ma, Lichao Sun, Jun Zhao, Qiang Yang, and Philip S. Yu. Privacy and robustness in federated learning: Attacks and defenses. *IEEE transactions on neural networks and learning systems*, PP, 2020.

[7] Yuan Mei, Binbin Guo, Danyang Xiao, and Weigang Wu. Fedvf: Personalized federated learning based on layer-wise parameter updates with variable frequency. *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pages 1–9, 2021.

[8] Xiaosong Ma, J. Zhang, Song Guo, and Wenchao Xu. Layer-wised model aggregation for personalized federated learning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10082–10091, 2022.

[9] Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. *J. Netw. Comput. Appl.*, 116:1–8, 2018.

[10] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *ArXiv*, abs/1812.00564, 2018.

[11] Chandra Thapa, Pathum Chamikara Mahawaga Arachchige, and Seyit Ahmet Çamtepe. Splitfed: When federated learning meets split learning. *ArXiv*, abs/2004.12088, 2020.

[12] Yansong Gao, Minki Kim, Sharif Abuadbba, Yeonjae Kim, Chandra Thapa, Kyuyeon Kim, Seyit Ahmet Çamtepe, Hyoungshick Kim, and Surya Nepal. End-to-end evaluation of federated learning and split learning for internet of things. *2020 International Symposium on Reliable Distributed Systems (SRDS)*, pages 91–100, 2020.

[13] Manoj Ghuhan Arivazhagan, V. Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *ArXiv*, abs/1912.00818, 2019.

[14] Duc Viet Bui, Kshitiz Malik, Jack Goetz, Honglei Liu, Seungwhan Moon, Anuj Kumar, and Kang G. Shin. Federated user representation learning. *ArXiv*, abs/1909.12535, 2019.

[15] Paul Pu Liang, Terrance Liu, Liu Ziyin, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *ArXiv*, abs/2001.01523, 2020.

[16] Viraj Kulkarni, Milind Kulkarni, and Aniruddha Pant. Survey of personalization techniques for federated learning. *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pages 794–797, 2020.

[17] Théo Jourdan, Antoine Boutet, and Carole Frindel. Privacy assessment of federated learning using private personalized layers. *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2021.

[18] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, 2021.

[19] Xin-Yi Tong, Xiangxiang Xu, and Shao-Lun Huang. On sample complexity of learning shared representations: The asymptotic regime. *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1–8, 2022.

[20] Maoye Ren and Xinhai Yu. Multibranch multilevel federated learning for a better feature extraction and a plug-and-play dynamic-adjusting double flow personalization approach. *Applied Intelligence*, 2022.

[21] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13:1333–1345, 2018.

[22] M. Sadegh Riazi, Christian Weinert, Oleksandr Tkachenko, Ebrahim M. Songhori, T. Schneider, and Farinaz Koushanfar. Chameleon: A hybrid secure computation framework for machine learning applications. *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 2018.

[23] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. B. McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[24] Roger Iyengar, Joseph P. Near, Dawn Xiaodong Song, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Towards practical differentially private convex optimization. *2019 IEEE Symposium on Security and Privacy (SP)*, pages 299–316, 2019.

[25] Yongheng Deng, Weining Chen, Ju Ren, Feng Lyu, Yang Liu, Yunxin Liu, and Yaoxue Zhang. Tailorfl: Dual-personalized federated learning under system and data heterogeneity. *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022.

[26] Guogang Zhu, Xuefeng Liu, Shaojie Tang, and Jianwei Niu. Aligning before aggregating: Enabling cross-domain federated learning via consistent feature extraction. *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, pages 809–819, 2022.

[27] Renjie Tang, Junzhou Luo, Junbo Qian, and Jiahui Jin. Personalized federated learning for ecg classification based on feature alignment. *Security and Communication Networks*, 2021.

[28] Zhixiong Chen, Wenqiang Yi, Arumugam Nallanathan, and Geoffrey Y. Li. Federated learning for energy-limited wireless networks: A partial model aggregation approach. *ArXiv*, abs/2204.09746, 2022.

[29] Hai Jin, Dongshan Bai, Dezhong Yao, Yutong Dai, Lin Gu, Chen Yu, and Lichao Sun. Personalized edge intelligence via federated self-knowledge distillation. *IEEE Transactions on Parallel and Distributed Systems*, 34:567–580, 2023.

[30] Xuanming Ni, Xinyuan Shen, and Huimin Zhao. Federated optimization via knowledge codistillation. *Expert Systems with Applications*, 191:116310, 2022.

[31] Yiqiang Chen, Wang Lu, Xin Qin, Jindong Wang, and Xing Xie. Metafed: Federated learning among federations with cyclic knowledge distillation for personalized healthcare. *ArXiv*, abs/2206.08516, 2022.

[32] Eunjeong Jeong and Marios Kountouris. Personalized decentralized federated learning with knowledge distillation. *ArXiv*, abs/2302.12156, 2023.

[33] Chuhan Wu, Fangzhao Wu, Ruixuan Liu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. Communication-efficient federated learning via knowledge distillation. *Nature Communications*, 13, 2021.

[34] Chengxi Li, Gang Li, and Pramod K. Varshney. Decentralized federated learning via mutual knowledge transfer. *IEEE Internet of Things Journal*, 9:1136–1147, 2020.

[35] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. *Proceedings of machine learning research*, 139:12878–12889, 2021.

[36] Huancheng Chen, John Wenyu Wang, and Haris Vikalo. The best of both worlds: Accurate global and personalized models through federated learning with data-free hyper-knowledge distillation. *ArXiv*, abs/2301.08968, 2023.

[37] Taehyeon Kim and Se-Young Yun. Supernet training for federated image classification under system heterogeneity. *ArXiv*, abs/2206.01366, 2022.

[38] Lukasz Dudziak, Stefanos Laskaridis, and Javier Fernández-Marqués. Fedoras: Federated architecture search under system heterogeneity. *ArXiv*, abs/2206.11239, 2022.

[39] Jaehee Jang, Heonseok Ha, Dahuin Jung, and Sungroh Yoon. Fedclassavg: Local representation learning for personalized federated learning on heterogeneous neural networks. *Proceedings of the 51st International Conference on Parallel Processing*, 2022.

[40] Wenjing Yin. Personalized hybrid education framework based on neuroevolution methodologies. *Computational Intelligence and Neuroscience*, 2022, 2022.

[41] Tianchun Wan, Wei Cheng, Dongsheng Luo, Wenchao Yu, Jingchao Ni, Liang Tong, Haifeng Chen, and Xiang Zhang. Personalized federated learning via heterogeneous modular networks. *2022 IEEE International Conference on Data Mining (ICDM)*, pages 1197–1202, 2022.

[42] Erum Mushtaq, Chaoyang He, Jie Ding, and Amir Salman Avestimehr. Spider: Searching personalized neural architecture for federated learning. *ArXiv*, abs/2112.13939, 2021.

[43] Minh Hoang and Carl Kingsford. Personalized neural architecture search for federated learning. In *1st NeurIPS Workshop on New Frontiers in Federated Learning (NFFL 2021)*, 2021.

[44] Hangyu Zhu and Yaochu Jin. Real-time federated evolutionary neural architecture search. *IEEE Transactions on Evolutionary Computation*, 26:364–378, 2020.

[45] Sixing Yu, Phuong Nguyen, Waqwoya Abebe, Justin Stanley, Pablo Muñoz, and Ali Jannesari. Resource-aware heterogeneous federated learning using neural architecture search. *ArXiv*, abs/2211.05716, 2022.

[46] Bar Rubinstein, Yoav Filin, Nir Shlezinger, and Nariman Farsad. Personalized sleep state classification via learned factor graphs. *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 1427–1431, 2022.

[47] Rui Hu, Yuanxiong Guo, Hongning Li, Qingqi Pei, and Yanmin Gong. Personalized federated learning with differential privacy. *IEEE Internet of Things Journal*, 7:9530–9539, 2020.

[48] Anda Cheng, Zhen Wang, Yaliang Li, and Jianwei Cheng. Hpn: Personalized federated hyperparameter optimization. *ArXiv*, abs/2304.05195, 2023.

[49] Ziyuan Yang, Wenjun Xia, Zexin Lu, Yin Chen, Xiaoxia Li, and Yi Zhang. Hypernetwork-based personalized federated learning for multi-institutional ct imaging. *ArXiv*, abs/2206.03709, 2022.

[50] Phung Lai, Nhathai Phan, Abdallah Khreishah, Issa M. Khalil, and Xintao Wu. Model transferring attacks to backdoor hypernetwork in personalized federated learning. *ArXiv*, abs/2201.07063, 2022.

[51] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, 2021.

[52] Malliga Subramanian, V. R. Rajasekar, Sathishkumar V E, Kogilavani Shanmugavadivel, and P. S. Nandhini. Effectiveness of decentralized federated learning algorithms in healthcare: A case study on cancer classification. *Electronics*, 2022.

[53] Yang Lu, Pinxin Qian, Gang Huang, and Hanzi Wang. Personalized federated learning on long-tailed data via adversarial feature augmentation. *ArXiv*, abs/2303.15168, 2023.

[54] Wangzhuo Yang, Bo Chen, Yijun Shen, Jiong Liu, and Ling Yu. Crossfusion rule for personalized federated learning. *ArXiv*, abs/2302.02531, 2023.

[55] Moming Duan, Duo Liu, Xianzhang Chen, Renping Liu, Yujuan Tan, and Liang Liang. Self-balancing federated learning with global imbalanced data in mobile systems. *IEEE Transactions on Parallel and Distributed Systems*, 32:59–71, 2021.

[56] Yupei Zhang, Shuangshuang Wei, Yifei Wang, Yunan Xu, Yuxin Li, and Xuequn Shang. A personalized federated learning framework using side information for heterogeneous data classification. *2022 IEEE International Conference on Big Data (Big Data)*, pages 3455–3461, 2022.

[57] Moming Duan. Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. *2019 IEEE 37th International Conference on Computer Design (ICCD)*, pages 246–254, 2019.

[58] Kaiyu Zheng, Xuefeng Liu, Guogang Zhu, Xinghao Wu, and Jianwei Niu. Channelfed: Enabling personalized federated learning via localized channel attention. *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pages 2987–2992, 2022.

[59] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *ArXiv*, abs/1806.00582, 2018.

[60] Qiong Wu, Xu Chen, Zhi Zhou, and Junshan Zhang. Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring. *IEEE Transactions on Mobile Computing*, 21:2818–2832, 2020.

[61] Tiansheng Huang, Li Shen, Yan Sun, Weiwei Lin, and Dacheng Tao. Fusion of global and local knowledge for personalized federated learning. *ArXiv*, abs/2302.11051, 2023.

[62] Dui Wang, Li Shen, Yong Luo, Han Hu, Kehua Su, Yonggang Wen, and Dacheng Tao. Fedabc: Targeting fair competition in personalized federated learning. *ArXiv*, abs/2302.07450, 2023.

[63] Luke Guerdan and Hatice Gunes. Decentralized robot learning for personalization and privacy. *arXiv preprint arXiv:2201.05527*, 2022.

[64] Yiqing Shen, Yuyin Zhou, and Lequan Yu. Cd2-pfed: Cyclic distillation-guided channel decoupling for model personalization in federated learning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10031–10040, 2022.

[65] Yiqing Shen, Baiyun Liu, Ruize Yu, Yudong Wang, Shaokang Wang, Jiangfen Wu, and Weidao Chen. Federated learning for chronic obstructive pulmonary disease classification with partial personalized attention mechanism. *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1706–1709, 2022.

[66] Mi Luo, Fei Chen, D. Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *ArXiv*, abs/2106.05001, 2021.

[67] Yen hsiu Chou, linda Qiao, Chenxi Sun, Derun Cai, Moxian Song, and Hongyan Li. Grp-fed: Addressing client imbalance in federated learning via global-regularized personalization. In *SDM*, 2021.

[68] Shengyuan Hu, Zhiwei Steven Wu, and Virginia Smith. Private multi-task learning: Formulation and applications to federated learning. *Trans. Mach. Learn. Res.*, 2023, 2021.

[69] Achintya Kundu, Pengqian Yu, Laura Wynter, and Shiau Hong Lim. Robustness and personalization in federated learning: A unified approach via regularization. *2022 IEEE International Conference on Edge Computing and Communications (EDGE)*, pages 1–11, 2022.

[70] Ruoyou Wu, Cheng Li, Juan Zou, Qiegen Liu, Hairong Zheng, and Shanshan Wang. Model-based federated learning for accurate mr image reconstruction from undersampled k-space data. *arXiv preprint arXiv:2304.07502*, 2023.

[71] Guang-Ming Li, Wansen Wu, Yan Sun, Li Shen, Baoyuan Wu, and Dacheng Tao. Visual prompt based personalized federated learning. *ArXiv*, abs/2303.08678, 2023.

[72] Canh T. Dinh, Thanh Tung Vu, Nguyen H. Tran, Minh N. Dao, and Hongyu Zhang. A new look and convergence rate of federated multi-task learning with laplacian regularization. *IEEE transactions on neural networks and learning systems*, PP, 2021.

[73] Zhaoping Xiong, Ziqiang Cheng, Xinyuan Lin, Chi Xu, Xiaohong Liu, Dingyan Wang, Xiaomin Luo, Yong Zhang, Hualiang Jiang, Nan Qiao, and Mingyue Zheng. Facing small and biased data dilemma in drug

discovery with enhanced federated learning approaches. *Science China Life Sciences*, 65:529 – 539, 2021.

[74] Xubo Yue, Maher Nouiehed, and Raed Al Kontar. Gifair-fl: A framework for group and individual fairness in federated learning. *INFORMS Journal on Data Science*, 2021.

[75] Devansh Shah, Parijat Dube, Supriyo Chakraborty, and Ashish Verma. Adversarial training in communication constrained federated learning. *ArXiv*, abs/2103.01319, 2021.

[76] Xingjian Cao, Gang Sun, Hongfang Yu, and Mohsen Guizani. Perfedgan: Personalized federated learning via generative adversarial networks. *IEEE Internet of Things Journal*, 10:3749–3762, 2022.

[77] Jingyang Zhang, Yiran Chen, and Hai Helen Li. Privacy leakage of adversarial training models in federated learning systems. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 107–113, 2022.

[78] Yurui Yang and Bo Jiang. Towards group fairness via semi-centralized adversarial training in federated learning. *2022 23rd IEEE International Conference on Mobile Data Management (MDM)*, pages 482–487, 2022.

[79] Taejin Kim, Shubhranshu Singh, Nikhil Madaan, and Carlee Joe-Wong. pfeddef: Defending grey-box attacks for personalized federated learning. *ArXiv*, abs/2209.08412, 2022.

[80] Onat Dalmaz, Usama Mirza, Gokberk Elmas, Muzaffer Ozbey, Salman UH Dar, Emir Ceyani, Salman Avestimehr, and Tolga cCukur. One model to unite them all: Personalized federated learning of multi-contrast mri synthesis. *ArXiv*, abs/2207.06509, 2022.

[81] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021.

[82] Chelsea Finn, P. Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *ArXiv*, abs/1703.03400, 2017.

[83] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *ArXiv*, abs/1803.02999, 2018.

[84] Yujia Gao, Pengfei Wang, L. Liu, Chi Zhang, and Huadong Ma. Configure your federation: Hierarchical attention-enhanced meta-learning network for personalized federated learning. *ACM Transactions on Intelligent Systems and Technology*, 2023.

[85] Yicheng Di and Y. Liu. Mfpcdr: A meta-learning-based model for federated personalized cross-domain recommendation. *Applied Sciences*, 2023.

[86] Lei Yang, Jiaming Huang, Wanyu Lin, and Jiannong Cao. Personalized federated learning on non-iid data via group-based meta-learning. *ACM Transactions on Knowledge Discovery from Data*, 17:1 – 20, 2022.

[87] Alireza Fallah, Aryan Mokhtari, and Asuman E. Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *Neural Information Processing Systems*, 2020.

[88] Canh T. Dinh, Nguyen H. Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. *ArXiv*, abs/2006.08848, 2020.

[89] Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Adaptive gradient-based meta-learning methods. In *Neural Information Processing Systems*, 2019.

[90] Yihan Jiang, Jakub Konecný, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *ArXiv*, abs/1909.12488, 2019.

[91] Mariel Werner, Lie He, Sai Praneeth Karimireddy, Michael Jordan, and Martin Jaggi. Towards provably personalized federated learning via threshold-clustering of similar clients. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

[92] Joo Hun Yoo, Ha Min Son, Hyejun Jeong, Eun-Hye Jang, Ah Young Kim, Han Young Yu, Hong Jin Jeon, and Tai-Myung Chung. Personalized federated learning with clustering: Non-iid heart rate variability data application. *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1046–1051, 2021.

[93] Yae Jee Cho, Jianyu Wang, Tarun Chiruvolu, and Gauri Joshi. Personalized federated learning for heterogeneous clients with clustered knowledge transfer. *ArXiv*, abs/2109.08119, 2021.

[94] Aleksandar Armacki, Dragana Bajović, Duan Jakovetić, and Soummya Kar. Personalized federated learning via convex clustering. *2022 IEEE International Smart Cities Conference (ISC2)*, pages 1–7, 2022.

[95] Zhiyong Jie, Shuhong Chen, Junqiu Lai, Muhammad Arif, and Zongyuan He. Personalized federated recommendation system with historical parameter clustering. *Journal of Ambient Intelligence and Humanized Computing*, 2022.

[96] Zhipeng Gao, Yan Yang, Chen Zhao, and Zijia Mo. Cfedper: Clustered federated learning with two-stages optimization for personalization. *2022 18th International Conference on Mobility, Sensing and Networking (MSN)*, pages 171–177, 2022.

[97] Yae Jee Cho, Jianyu Wang, Tarun Chirvolu, and Gauri Joshi. Communication-efficient and model-heterogeneous personalized federated learning via clustered knowledge transfer. *IEEE Journal of Selected Topics in Signal Processing*, 17:234–247, 2023.

[98] Muhammad Firdaus, Si-Wan Noh, Zhuohao Qian, Harashta Tatimma Larasati, and Kyung Hyune Rhee. Personalized federated learning for heterogeneous data: A distributed edge clustering approach. *Mathematical Biosciences and Engineering*, 2023.

[99] Zijian Li, Zihan Chen, Xiaohui Wei, Shang Gao, Chenghao Ren, and Tony Q. S. Quek. Hpfl-cn: Communication-efficient hierarchical personalized federated edge learning via complex network feature clustering. *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 325–333, 2022.

[100] Sichun Luo, Yuanzhang Xiao, and Linqi Song. Personalized federated recommendation via joint representation learning, user clustering, and model adaptation. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022.

[101] Youpeng Li, Xuyu Wang, and Lingling An. Hierarchical clustering-based personalized federated learning for robust and fair human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7:1 – 38, 2022.

[102] Yixuan Li, Xiaoqi Qin, Hao Chen, Kaifeng Han, and Ping Zhang. Energy-aware edge association for cluster-based personalized federated learning. *IEEE Transactions on Vehicular Technology*, 71:6756–6761, 2022.

[103] Jie Ma, Ming Xie, and Guodong Long. Personalized federated learning with robust clustering against model poisoning. In *International Conference on Advanced Data Mining and Applications*, 2022.

[104] Moming Duan, Duo Liu, Xinyuan Ji, Renping Liu, Liang Liang, Xianzhang Chen, and Yujuan Tan. Fedgroup: Efficient federated learning via decomposed similarity-based clustering. *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pages 228–237, 2021.

[105] Laizhong Cui, Xiaoxin Su, Yipeng Zhou, and Lei Zhang. Clustergrad: Adaptive gradient compression by clustering in federated learning. *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–7, 2020.

[106] Boxiang Lyu, Filip Hanzely, and Mladen Kolar. Personalized federated learning with multiple known clusters. *ArXiv*, abs/2204.13619, 2022.

[107] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 32:3710–3722, 2019.

[108] Christopher Briggs, Zhong Fan, and Péter András. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, 2020.

[109] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *IEEE Transactions on Information Theory*, 68(12):8076–8091, 2022.

[110] Li Huang and Dianbo Liu. Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. *Journal of biomedical informatics*, page 103291, 2019.

[111] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[112] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group

knowledge transfer: Federated learning of large cnns at the edge. *arXiv: Learning*, 2020.

[113] Ilai Bistritz, Ariana J. Mann, and Nicholas Bambos. Distributed distillation for on-device learning. In *Neural Information Processing Systems*, 2020.

[114] Yu Liu, Xuhui Jia, Mingxing Tan, Raviteja Vemulapalli, Yukun Zhu, Bradley Green, and Xiaogang Wang. Search to distill: Pearls are everywhere but not the eyes. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7536–7545, 2019.

[115] Changlin Li, Jiefeng Peng, Liuchun Yuan, Guangrun Wang, Xiaodan Liang, Liang Lin, and Xiaojun Chang. Block-wisely supervised neural architecture search with knowledge distillation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1986–1995, 2019.

[116] Hangyu Zhu, Haoyu Zhang, and Yaochu Jin. From federated learning to federated neural architecture search: a survey. *Complex & Intelligent Systems*, 7:639 – 657, 2021.

[117] Pietro Cassará, Alberto Gotta, and Lorenzo Valerio. Federated feature selection for cyber-physical systems of systems. *IEEE Transactions on Vehicular Technology*, 71(9):9937–9950, 2022.

[118] Jiang Wu, Xuezheng Liu, Jiahao Liu, Miao Hu, and Di Wu. Dpfed: Toward fair personalized federated learning with fast convergence. *2022 18th International Conference on Mobility, Sensing and Networking (MSN)*, pages 510–517, 2022.

[119] Xin Yang and Omid Ardakanian. Blinder: End-to-end privacy protection in sensing systems via personalized federated learning. *ArXiv*, abs/2209.12046, 2022.

[120] Batiste Le Bars, Aurélien Bellet, Marc Tommasi, Erick Lavoie, and Anne-Marie Kermarrec. Refined convergence and topology learning for decentralized sgd with heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 1672–1702. PMLR, 2023.

[121] Yupei Zhang, Shuangshuang Wei, Yifei Wang, Yunan Xu, Yuxin Li, and Xuequn Shang. A personalized federated learning framework using side information for heterogeneous data classification. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 3455–3461, 2022.

[122] Van-Tuan Tran, Huy-Hieu Pham, and Kok-Seng Wong. Personalized privacy-preserving framework for cross-silo federated learning. *ArXiv*, abs/2302.12020, 2023.

[123] Wenzhu Li and Shuang Wang. Federated meta-learning for spatial-temporal prediction. *Neural Computing and Applications*, 34:10355 – 10374, 2022.

[124] Mohamed S. Nafea, Eugine Shin, and Aylin Yener. Proportional fair clustered federated learning. *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 2022–2027, 2022.

[125] Li-Yu Daisy Liu, Shengyuan Hu, Zhiwei Steven Wu, and Virginia Smith. On privacy and personalization in cross-silo federated learning. *ArXiv*, abs/2206.07902, 2022.