IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 July 2025, accepted 9 October 2025, date of publication 20 October 2025, date of current version 27 October 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3623444

SURVEY

Leveraging Unlabeled Data in Federated Learning: A Review

KÜBRA ULUDAG[1], ÇIGDEM EROGLU ERDEM[2], (Member, IEEE), AND ÖMER KORÇAK[1], (Member, IEEE)

[1]Department of Computer Engineering, Faculty of Engineering, Marmara University, 34854 Istanbul, Türkiye
[2]Department of Electrical and Electronics Engineering, Özyeğin University, 34794 Istanbul, Türkiye

Corresponding author: Kübra Uludag (kubra.uludag@marmara.edu.tr)

This work was supported by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) under Grant EEAG-122E025.

ABSTRACT In centralized machine learning, both the data and model to be trained reside on a single server, which may cause problems regarding data privacy as sensitive or personal data need to be transferred from clients to the server. Federated learning has been proposed to provide a solution to this problem by allowing the training of a model without the data leaving the clients. This training takes place between a coordinating server and the clients by continuously exchanging the model parameters instead of exchanging data. In real-life applications, the data on some of the clients or the server may be partially labeled or completely unlabeled, which poses a severe challenge to federated learning. In this paper, we present a survey of recently proposed methods that leverage unlabeled data in a federated learning setting to improve model performance. We also present a novel taxonomy of the methods that leverage unlabeled data based on whether the unlabeled data is assigned a pseudo-label during the process or not. We summarize the datasets, main data modalities, and application areas of federated learning with unlabeled data methods in the literature and highlight future research directions. We believe that this survey will be a useful guide for researchers planning to work on federated learning with partially labeled data.

INDEX TERMS Federated learning, privacy protection, pseudo-labeling, self-supervised learning, semi-supervised learning.

I. INTRODUCTION

With the tremendous increase in both data generation and processing capacity in recent years, machine learning (ML) and deep learning (DL) have become applicable with high success rates in a wide range of applications including healthcare, natural language processing, autonomous vehicles and computer vision (e.g. image classification, object detection, activity recognition). In traditional centralized learning, all data collected by end devices (such as IoT devices, mobile phones, and wearables) to train ML or DL models must be stored centrally, typically in a data center or cloud server. However, this approach presents several challenges [1]. First, growing privacy concerns among data owners arise due to data theft or unauthorized usage, leading to a reluctance to share data. Furthermore, governments have enacted regulations such as the General Data Protection Regulation (GDPR) [2] in the European Union (EU) and the Consumer Privacy Bill of Rights (CPBR) [3] in the United States (US) to protect user personal data, imposing restrictions on data collection and processing by companies or operators. Consequently, these constraints limit the amount of data collected on the central server, which affects the performance of the models trained on a central server. Furthermore, in traditional centralized learning, long propagation delays occur due to remote communication with the central server, causing unacceptable latency, particularly in real-time applications such as self-driving car systems. Moreover, transmitting raw collected data to the server incurs substantial network costs. All of these issues significantly limit the applicability of centralized learning methods.

In traditional distributed learning, data is initially stored on a central server and then distributed to users for processing; therefore, this approach also violates privacy [4]. As user

The associate editor coordinating the review of this manuscript and approving it for publication was Yin Zhang.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

devices have recently been equipped with more powerful sensors and have higher computing power, the distributed on-site ML approach [5] has emerged in which each user builds its ML or DL model and trains it using its private dataset. The on-site ML approach solves the privacy and latency problems of the centralized approach but provides limited performance, as it cannot leverage the data of other users. Since ML and especially DL models need very large datasets to achieve a good performance, the Federated Learning (FL) approach has emerged to address all the problems mentioned above.

The main difference between FL and centralized learning is that the raw data collected is kept locally on each device (also called a client), and only the model updates (that is, the weights or gradients of the model) are sent to a server for the aggregation process. Thus, it preserves data privacy and reduces the communication overhead on the network. In addition, FL has low latency, as the user can use the local model for inferring without waiting for the output from the central server. Unlike on-site distributed learning, each device in FL benefits from the experience (data) of other devices as it uses the aggregated global model and thus has the potential to perform better.

Despite its benefits, FL faces specific challenges. The majority of literature on FL concentrates on supervised settings, primarily tackling four main challenges: communication cost, statistical heterogeneity, system heterogeneity, and security/privacy, which will be discussed in more detail in Section II.

A. RELATED SURVEYS ON FL WITH LABELED DATA

Several survey papers are available in the literature that address current challenges of FL and comprehensively examine FL methods based on labeled data [1], [4], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. Many of these studies have addressed the fundamental concepts of FL, including structural data partitioning in FL (e.g. horizontal and vertical), types of FL (e.g. cross-device and cross-silo), and the historical development of FL [1], [6], [8], [10], [11]. The statistical and system heterogeneity encountered in FL systems is identified as a challenge in almost all existing work [8], [9], [13], [14]. In particular, [13] has extended this concept by focusing on heterogeneity between data types for multimodal FL.

Several surveys have focused more on one of the main challenges in FL: privacy and security. These surveys have covered privacy preservation techniques (e.g., differential privacy, homomorphic encryption, secure aggregation) and security threats (model poisoning, backdoor attacks) in depth [1], [4], [7], [9], [12], [14], [16]. More recent surveys [16], have also introduced new approaches such as hardware-based security solutions (e.g., a trusted executed environment).

The use of FL in different domains such as IoT [1], [7], [15], Industry 4.0 [7], mobile services [4], healthcare [8], [10], [14] and finance [10] has been covered in many studies. Works such as [7], [14], and [16] present techniques (e.g., sparsification, quantization, network-based client selection) to reduce communication cost and optimize resource utilization. Other studies [9], [12], [16] highlight important challenges such as personalization and fairness.

B. FROM LABELED TO UNLABELED DATA IN FL

Although each of the above-mentioned survey papers focuses on various FL challenges and application domains, they are all limited to supervised FL settings. This implies that all data on clients should be labeled. However, in real-world applications, clients may not be willing to participate in the labeling process due to various labeling costs (e.g. time) or because labeling requires domain expertise [4], [8], [14], [17], [18]. The fact that clients have little or no labeled data raises the need for learning paradigms that can take advantage of unlabeled data in FL. The use of different learning paradigms in a FL environment depends on different factors such as data heterogeneity, amount, and distribution of labeled data, and resource constraints (e.g., computation capability).

Semi-supervised learning approaches in the context of FL are often used when all clients have a small amount of labeled data or when only some of the clients have labeled data. For example, in medical image classification applications, some hospitals may have medical images labeled by experts, whereas hospitals without the required expertise may not be able to perform this labeling process. In such cases, the integration of semi-supervised learning methods such as pseudo-labeling, consistency regularization, and mean-teacher into the federated environment has led to new approaches that can exploit both labeled and unlabeled data [19], [20]. Data heterogeneity and label quality play an important role in the performance of semi-supervised FL methods. Therefore, in the process of adapting these methods to federated environments, additional strategies (such as thresholding [21], [22], [23], class balancing [24]) have been developed to deal with inconsistencies in data distribution and improve label accuracy. However, when the amount of labeled data is insufficient and severely unbalanced, semi-supervised learning methods may have limited performance, and in scenarios where there are no labeled data at all, these methods may not be feasible. This has stimulated efforts to integrate self-supervised and unsupervised approaches into the FL framework, enabling direct learning from unlabeled data.

Self-supervised learning methods (e.g. contrastive learning) [25] focus on learning strong representations from unlabeled data through pretext tasks (e.g. masking out a part of an image and trying to predict that part). Then, these learned representations are used in other classification tasks. For example, in the context of FL, we see applications such as image classification [26] or disease diagnosis in healthcare care [27], [28].

Unsupervised learning [26] aims to learn meaningful representations or structures from unlabeled data. In the

context of FL, this approach is particularly valuable when data labeling is not possible. Activity recognition from sensor data [29] or VAE-based content caching applications [30], [31] are prominent examples of unsupervised learning in FL.

C. MOTIVATION AND CONTRIBUTIONS

The integration of these learning paradigms into FL systems has brought both significant benefits and new challenges (e.g., pseudo-label confidence, catastrophic forgetting, and representation shift). This bidirectional impact has led to a large number of studies that either aim to leverage the potential of these methods or address the emerging challenges associated with their use. Therefore, there has been an intense research interest in this field in recent years. However, in-depth analyses that comprehensively evaluate these studies, systematically classify methods, or provide a comparative perspective are very limited in the literature [32]. Hence, a thorough and contemporary survey is needed to holistically evaluate various learning paradigms, scenario configurations, and methodological strategies in this domain.

In this paper, different from previous surveys, we emphasize the issue of label scarcity in FL and concentrate on FL methods that use unlabeled data (FLUD). Specifically, we review recent publications that introduce algorithms that utilize unlabeled data in the model training process. These techniques utilize semi-supervised learning, unsupervised learning, and self-supervised learning. We propose a novel taxonomy that categorizes these methods according to how they incorporate unlabeled data into the training process.

There is a recent review article that also focuses on FL methods without full labels [32]. Our survey differs from [32] in several aspects:

- We analyze the distribution of labeled and unlabeled data on the server and on clients in four different cases according to the locations of labeled and unlabeled data, while [32] focuses only on two cases.
- We present a taxonomy of FLUD methods, classifying them according to their approach to using unlabeled data.
- We categorize FLUD methods according to their data modalities and primary application domains, such as image classification, medical applications, speech recognition, etc.
- Our survey paper is more comprehensive including newer methods published after [32].
- We examine the datasets used in FLUD across multiple data modalities such as image, text, audio, and sensor data (see Table 1).
- To assess the performance trends of FLUD approaches and their applicability in heterogeneous environments, we have analyzed several representative studies in the literature and compared their methodological achievements (see Table 5).
- We outline the challenges and open unresolved issues in this field and provide recommendations to guide researchers (see Section VI).

D. PAPER ORGANIZATION

The paper is structured as follows. Section II provides fundamental insight into FL and discusses the key challenges in more detail. Section III focuses on FL methods that utilize unlabeled data (FLUD) and delineates different scenarios based on the location of unlabeled data. Section IV introduces a novel taxonomy for FLUD methods. Section V examines FLUD methods based on their application domains. Section VI highlights several unresolved issues in FLUD and outlines potential future research directions, which is followed by the conclusions given in Section VII.

II. FEDERATED LEARNING

The FL concept was first proposed by McMahan et al. [33] in 2016, allowing an ML / DL model to be trained locally and collaboratively on edge nodes or end-user devices (also called clients) under the coordination of a central server. In classical FL (known as FedAvg) [33], several communication rounds are conducted between the server and clients to achieve the desired global model performance. The FL training process is depicted in Figure 1, and the traditional (supervised) FL steps are explained below.

Initially, the server generates an initial global model w_0 (this model can be initialized with random weights or trained with a small amount of labeled data stored on the server), and then it starts the FL process which proceeds as follows:

(i) The server selects a subset of clients S_t in round t based on some criteria (such as charge level, idleness, and connectivity to an unmetered network) and sends the (initial, if $t = 0$) global model weights w_g^t to all selected clients.

(ii) Then each selected client k initializes the local model w_k^t with the model parameters w_g^t received from the server and trains the local model using its labeled private data.

(iii) After completing the local training, all active clients (some of the selected clients might dropout due to several reasons such as poor connection or resource constraints) send their model updates (i.e., weights or gradients) back to the server.

(iv) The server aggregates the model updates from the clients by weighting them based on the size of each client's dataset n_k (that is, $w_g^{t+1} = \frac{1}{\sum_{k\in S_t} n_k} \sum_{k\in S_t} n_k w_k^t$) and transmits the new (aggregated) model w_g^{t+1} to the clients to use in the next round $t+1$.

(v) This process continues for a certain number of rounds.

In the following, we briefly review the main challenges in FL: communication cost, statistical heterogeneity, system heterogeneity, privacy, and label scarcity. The main focus of this survey paper is the label scarcity problem, which will be discussed in Section III.

A. COMMUNICATION COST

In FL, compared to cloud-based centralized methods, the amount of data transmitted over the network is expected to be lower, as only model updates or gradient values are

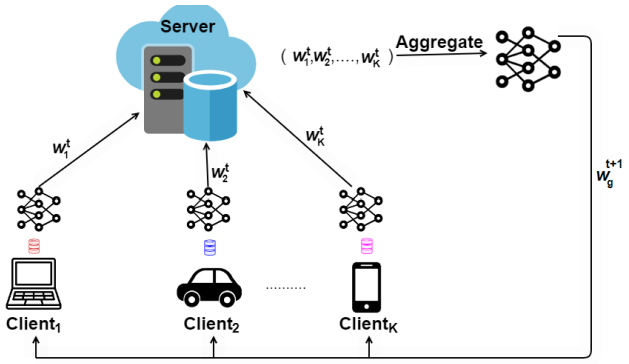


FIGURE 1. Training process in FL.

sent to the server rather than all raw data stored locally on clients. However, a large number of clients can participate in a federated network, and each model update communicated to the server may contain millions of parameters. In addition, many communication rounds can occur between the clients and the server until the desired accuracy is reached [4], [14]. Therefore, communication efficiency is a critical issue in FL. In this context, there have been many studies that can be categorized into two aspects: (i) reducing the total number of communication rounds or (ii) reducing the transmitted message size.

To reduce the total number of communication rounds, Yao et al. [34] propose a two-stream model update in which clients keep the global model fixed during the training process and use it as a reference while training the local model through backpropagation. In other words, the local model is trained not only with local data but also with the help of the global model by trying to reduce the maximum mean discrepancy loss (i.e., the difference between the features extracted by the local and global model). This speeds up the training process and reduces the number of communication rounds. Liu et al. [35] propose a hierarchical FL client-edge-cloud system, HierFAVG, where edge servers act as an intermediary between clients and the cloud server. Clients send their local model updates to edge servers instead of sending them to the cloud server directly. In [36], the authors develop an adaptive sampling method to categorize training data into ''easy'' and ''hard'' samples for each client, along with a client sampling method that selects models with lower losses. This approach aims to accelerate the convergence of the models.

Konečný et al. [37] propose two techniques, called structured updates and sketched updates, to reduce the size of the models before sending them to the server. In [38], the authors introduce two novel schemes that apply Federated Dropout and a lossy compression technique to provide a reduction in local computation and communication costs for both the server and the clients. Sattler et al. [39] introduce a novel compression framework designed to meet specific FL requirements, including reducing communication costs, ensuring robustness against non-IID data, handling a large number of clients and managing partial participation.

In [40], the authors proposed a Communication-Mitigated FL (CMFL) solution where clients only send their updates to the server if their models are identified as relevant. In [41], He et al. present a new framework, Group Knowledge Transfer (FedGKT), which enables the training of large neural networks on devices with limited resources in a federated environment by benefiting from distillation knowledge and split learning.

B. STATISTICAL HETEROGENEITY

In a federated setting, the distribution of data across local datasets can vary between clients due to factors such as geographical location, time, and usage behavior. Consequently, these local datasets may not accurately represent the overall data distribution in the network. Additionally, data dependency can arise from multiple users using the same device. Furthermore, clients may possess varying amounts of data based on their device usage frequency. As a result, data across clients may not exhibit independence and identical distribution (non-IID).

Statistical heterogeneity among clients causes a degradation in the performance of the global model. To address this problem, Zhao et al. [42] propose a data sharing strategy where the server initializes the global model with shared data instead of random weights. In another study [43], an incentive-based Hybrid-FL mechanism is proposed to address the problem of non-IID data. In this approach, some users send their data to the server in exchange for an incentive, and the global model is updated with the IID data that are generated on the server. In [44], statistical challenges are solved using multitask learning, and a new method called MOCHA is presented to overcome the challenges of the system. There are also other aggregation algorithms in the literature that deal with non-IID data distribution, including FedProx [45], FedMA [46], SCAFFOLD [47], and FedOpt [48].

C. SYSTEM HETEROGENEITY

In a federated environment, participating devices can vary in terms of storage capacity, computing and communication capabilities, and battery power. In addition, these characteristics can change dynamically during the training process. The heterogeneity among devices has a detrimental impact on the FL training process [14], [15]. Additionally, resource constraints on participating devices limit the overall system performance as well as their local performance. This means that each FL training round is determined according to the slowest participating device, known as the straggler effect [4], [14], [15]. In the baseline Federated Averaging (FedAvg) algorithm [33], the server awaits local updates from all participating devices to perform synchronous aggregation. For that reason, FedAvg is susceptible to the straggler effect.

To address this challenge, asynchronous FL approaches have been proposed in [49] and [50].

Client selection plays an important role in handling system heterogeneity. In many FL studies, clients are randomly chosen, assuming that they meet basic criteria (such as being charged, idle, and connected to an unmetered network). However, in real-world scenarios, heterogeneous clients, varying in resource capacity and data volume, can face challenges such as prolonged training time or decreased model performance [1]. To address this problem, some studies (e.g. [43], [51]) have emerged that take advantage of information on the client's resources for the selection of the client. Moreover, to accommodate varying computation capabilities among clients, there are approaches that allow clients to use different architectures for local models than for the global model, as demonstrated in [52].

In conventional FL, it is assumed that the devices remain active when selected. However, in real scenarios, some participants may drop out due to issues such as poor connection, computation, or communication limitations, or they may choose not to participate in the training process due to resource costs or constraints [53]. Therefore, several studies [44], [45] have examined the impact of dropout participants on convergence.

D. PRIVACY

In FL, a model is trained locally on clients and aggregated globally on the server. During the training process, only model parameters are exchanged between the server and clients, and private data on the clients never leave the local devices. However, the central server or a hostile third party can infer some sensitive information from the communicated model during the training process. To address this concern, various privacy preservation techniques have been proposed, including differential privacy (DP) [54], secure multiparty computation (SMC) [55], and homomorphic encryption (HE) [56]. In [57], [58], DP mechanisms have been incorporated into FL, while Hao et al. [59] proposed a privacy-preserving FL framework that combined HE and DP mechanisms. In [60], the authors proposed an FL framework based on SMC and DP. In addition, various model aggregation strategies [61], [62] have been suggested to preserve privacy in FL.

E. LABEL SCARCITY

Considering real-world applications, the availability of labeled data on clients is considerably lower than the availability of unlabeled data because of the time and cost requirements of the labeling process. Since the scarcity of labeled data negatively affects model performance, this becomes an important issue to be addressed in the context of FL. To handle this problem, new federated settings have emerged that leverage unlabeled data (FLUD), which are classified and explained in more detail in Section III.

In the rest of the paper, our primary focus is on the label scarcity problem, while we also discuss solutions to other challenges as secondary objectives.

III. FEDERATED LEARNING LEVERAGING UNLABELED DATA (FLUD)

Most existing FL studies are based on supervised learning methods, assuming that all data in the federated network are labeled. However, in real-world scenarios, the data generated by clients might be partially labeled or completely unlabeled, since data annotation is time consuming and, in some cases (e.g., finance and healthcare applications), requires expertise [4], [8], [14], [17], [18]. Despite the scarcity of labels in the federated environment, the abundance of unlabeled data has invoked the *semi-supervised* FL (SSFL) approach. SSFL aims to address label scarcity in a federated environment by using unlabeled data [17], [19]. There are also *self-supervised* methods and *unsupervised* methods, where labeled data is not available to clients (e.g. [26], [27], [30], [63], [64], [65]). We have integrated these methods into a unified taxonomy called FLUD.

In general, we group the FLUD methods in the literature into four different scenarios (cases), as shown in Figure 2, depending on where the labeled and unlabeled data are located. Studies in the literature generally focus on the first two scenarios.

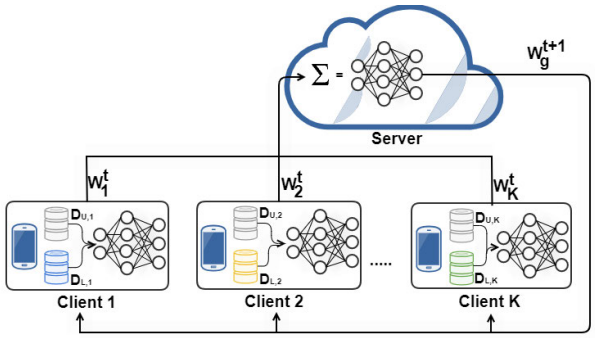
Case 1: In the first scenario (in Figure 2 (a)), each client k possesses both a private labeled dataset $D_{L,k}$ and a private unlabeled dataset $D_{U,k}$, both of which will be utilized in every round. During round t, every client k updates its local model (w_k^t) using the global model received (w_g^t)—a matrix containing the weights of the deep neural network model formed by aggregating local models. Subsequently, it trains the local model w_k^t using $D_{L,k}$ and $D_{U,k}$ to minimize its loss function $\mathcal{L}_k$, as given in:

$$w_k^t \leftarrow w_g^t, \quad w_k^t = w_k^t - \eta \nabla \mathcal{L}_k, \tag{1}$$

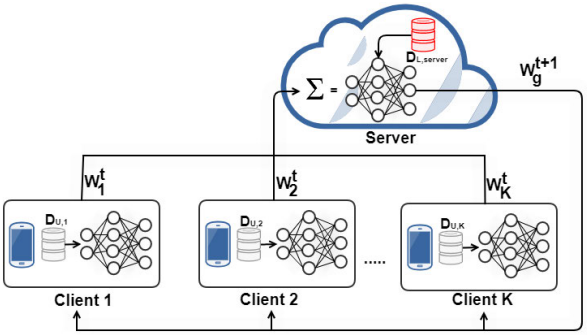
$$\mathcal{L}_k = \mathcal{L}_{s,k}(D_{L,k}) + \lambda \mathcal{L}_{u,k}(D_{U,k}), \tag{2}$$

where ∇ is the gradient of $\mathcal{L}_k$ with respect to the weights w and η is a learning rate. $\mathcal{L}_{s,k}$ is the supervised loss on client k while $\mathcal{L}_{u,k}$ is the unsupervised loss on client k. $\lambda \in [0, 1]$ is a hyper-parameter to control the effect of unlabeled data.

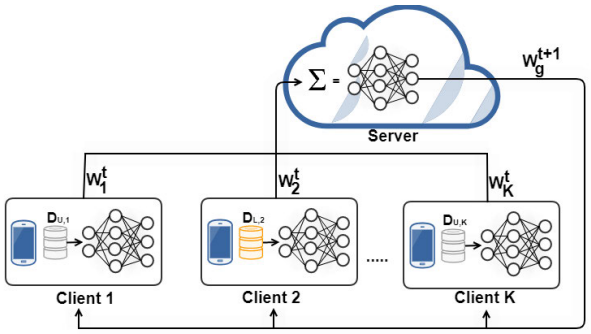
In the first FLUD scenario where both labeled data and unlabeled data are located on the client side (Case 1 in Figure 2(a)), the server acts as a coordinator that performs the distribution and aggregation of models. First, the server initiates the global model w_g^0 with random weights or trains it with a small amount of labeled data (which is only used in the initial step) from scratch. Alternatively, the server can use one of the pre-trained models in the literature as the initial model (e.g., ResNet [66] and VGG [67] for image classification and Tiny-Bert [68] for text classification). In each round t, the server sends the global model w_g^t to the clients. After training the models on the client side, a set of clients S_t upload their local updates to the server. The server then aggregates all



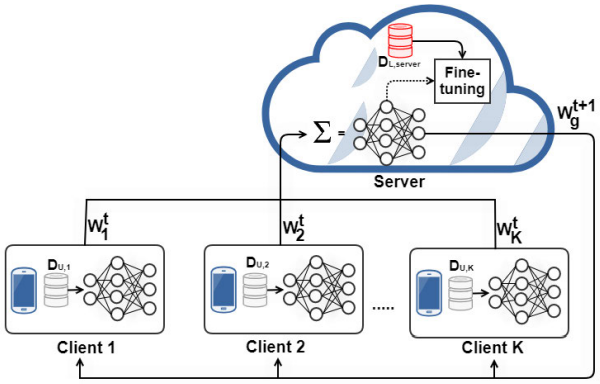
(a) **Case 1:** Both labeled (D_L) and unlabeled data (D_U) reside on the clients and are utilized in each round. A small set of labeled data may exist on the server, which is only used for initialization of the global model.



(b) **Case 2:** Unlabeled data (D_U) is located on the clients and labeled data ($D_{L,server}$) resides on the server. Labeled dataset is utilized during training in each FL round.



(c) **Case 3:** Only a small subset of clients possess labeled data (D_L), while the others have entirely unlabeled data (D_U). Both labeled and unlabeled data are employed in each round. A small set of labeled data may exist on the server, which is only used for initialization of the global model.



(d) **Case 4:** Clients only have unlabeled data (D_U), which is utilized in each round. Labeled data is only available on the server ($D_{L,server}$), which is used to fine-tune the global model after the completion of FL rounds.

FIGURE 2. FLUD scenarios categorized by the locations of labeled and unlabeled data. The colored (blue, yellow, green, and red) datasets represent labeled data (D_L), while the gray dataset depicts unlabeled data (D_U).

updates to the local models by weighting them according to the amount of data (n_k) on the clients or directly averaging them based on the number of clients selected, as given in the following equations:

$$w_g^{t+1} = \frac{1}{\sum_{k\in S_t} n_k} \sum_{k\in S_t} n_k w_k^t \quad \text{or} \quad w_g^{t+1} = \frac{1}{|S_t|} \sum_{k\in S_t} w_k^t. \tag{3}$$

Finally, the server sends the aggregated model (that is, a new global model) w_g^{t+1} to the clients to start the next training round.

Case 2: In the second scenario (in Figure 2 (b)), which is a more complex setting, each client k has only unlabeled data $D_{U,k}$, while the labeled data $D_{L,server}$ are on the server side. Initially, the server sends the global model to the clients. In each round t, each client k leverages the global model weights received w_g^t and refines the parameters of the local model w_t^t using local unlabeled data with an optimization method to minimize the unsupervised loss $\mathcal{L}_{u,k}$, as given in:

$$w_k^t \leftarrow w_g^t, \quad w_k^t = w_k^t - \eta_1 \nabla \mathcal{L}_{u,k}. \tag{4}$$

This local training step is repeated for several epochs E. Following local training, the clients send their local models to the server. On the server side, the received model updates are first aggregated as outlined in (3). Subsequently, the aggregated model w_g^{t+1} is trained using the labeled data on the server ($D_{L,server}$) minimizing the supervised loss $\mathcal{L}_{s,server}$, as given in:

$$w_g^{t+1} = w_g^{t+1} - \eta_2 \nabla \mathcal{L}_{s,server}. \tag{5}$$

Eventually, the updated global model w_g^{t+1} is sent to the clients for the next round of training.

Case 3: In the third scenario (in Figure 2 (c)), a few clients have fully labeled data, while the rest of the clients have only unlabeled data. In this scenario, there is a warm-up period during which only labeled clients are involved in the training process. After the warm-up period, unlabeled clients also participate in the training.

Case 4: In the fourth scenario (in Figure 2 (d)), all clients have only unlabeled data and train their models to learn hidden features of the unlabeled data. After the training

process, the trained model (or a part of the model) is fine-tuned with a small set of labeled data (located on the server) and then it is used for another (e.g. classification) task. Note that Case 4 is different from Case 2. In Case 2, the global model is trained with labeled data from the server in each FL training round, whereas in Case 4 the labeled data from the server are not included in the FL training process. Instead, the labeled data in Case 4 are solely used to fine-tune the global model after the FL process is completed.

A. APPLICATION AREAS

Considering the four FLUD scenarios discussed above, it is not possible to clearly deduce which case is more suitable for which application area since it depends on the ability of the clients to label local data. For example, in medical applications, where each client could be a hospital, local data may or may not be partially labeled, depending on the availability of a particular medical expertise at that hospital. In other applications, where edge devices (e.g., sensors) continuously collect data, it may not be possible to assign ground-truth labels to unlabeled data on local clients (e.g. [29]), or some clients may have limited labeled data ([69], [70]).

Many FL studies in the literature currently employ the Case 1 scenario. There are various applications that include image classification [20], [22], [71], [72], [73], [74], medical image classification [75], [76], [77], traffic sign recognition [78], travel mode identification [23], and human emotional recognition [79], [80] that are handled within the scope of the Case 1 scenario. Currently, Case 2 is used in image classification applications using general image datasets [21], [81], [82], [83], the human activity recognition application [29], and a few more specific real-world applications, such as building damage detection [84]. Case 3, compared to the previous two scenarios, has been used less, but most recent studies focus on this scenario. Among the studies we have reviewed, the Case 3 scenario appears in particular in medical image classification applications [24], [85], [86], [87], [88]. The Case 4 scenario has been addressed in studies using contrastive learning methods, regardless of the specific application area [26], [27], [63], [89].

IV. TAXONOMY OF FLUD METHODS BASED ON UTILIZATION OF UNLABELED DATA

Recently, numerous methods (e.g., [19], [20], [21], [23], [26], [29], [63], [65], [69], [74], [77], [78], [79], [81], [82], [83], [87], [88], [89], [90], [91], [92]) have been proposed to address label scarcity in FL scenarios. These methods explore various learning paradigms such as semi-supervised [93], unsupervised [94], and self-supervised learning [95], to exploit unlabeled data within a FLUD setting.

To organize this diverse body of work, we introduce a taxonomy that specifically addresses FLUD methods, as depicted in Figure 3. Although a taxonomy of deep semi-supervised learning methods is available in [96], to the best of our knowledge, our proposed taxonomy is the first to specifically address FLUD methods in the literature. We divide FLUD studies into three main categories, based on their utilization of unlabeled data. The first category explores *pseudo-labeling* methods to tackle the label scarcity issue, while the second category addresses it *without using pseudo-labeling*. The third category comprises hybrid solutions that combine both approaches. Detailed insights into each category follow below.

A. FLUD METHODS WITH PSEUDO-LABELING

In the first category of FLUD methods, unlabeled data on clients is included in the training process to improve the global model performance by applying the pseudo-labeling methods. These methods employing pseudo-labeling can be further divided into two groups: direct labeling (not utilizing a confidence measure) and confidence-based labeling (utilizing a confidence measure).

1) DIRECT LABELING METHODS

Direct labeling represents a straightforward pseudo-labeling technique in SSFL, bypassing the incorporation of a confidence metric when assigning pseudo-labels to unlabeled data. In methods utilizing direct labeling (e.g. [78]), each client k trains its local model w_k using the training set $D_{T,k}$, which only includes its own labeled data $D_{L,k}$, as described in the following equations:

$$w_k^t = w_k^t - \eta \nabla \mathcal{L}_k, \tag{6}$$

$$\mathcal{L}_k = \frac{1}{|D_{T,k}|} \sum_{(x_i, y_i) \in D_{T,k}} \ell(f_k(x_i, w_k^t), y_i), \tag{7}$$

where $\mathcal{L}_k$ typically represents the loss function for the client k, $f_k(.,.)$ is a function representing the model that predicts the class labels for an input x using weights w, and $\ell(.,.)$ is usually the cross-entropy loss. The trained model is then used to predict labels $\hat{\mathbf{y}}_{u,i}$ for each unlabeled data $x_{u,i}$ in $D_{U,k}$, as given in:

$$\hat{\mathbf{y}}_{u,i} = f_k(x_{u,i}, w_k^t), \quad x_{u,i} \in D_{U,k}. \tag{8}$$

If a client k lacks labeled data, the model trained on the server is used directly for labeling. Here, labeling is performed directly without considering a confidence threshold, e.g., signified by the probability value of a class label predicted by the model. In essence, each prediction $\hat{\mathbf{y}}_{u,i}$ by the classifier model w_k for unlabeled data $x_{u,i}$ in $D_{U,k}$ is transformed into a one-hot label, regardless of a confidence threshold, that is, $y_{u,i} = \text{argmax}(\hat{\mathbf{y}}_{u,i})$. Subsequently, the data $x_{u,i}$ with its one-hot label is directly appended to the training set $D_{T,k}$, forming $D_{T,k} = D_{L,k} \cup (x_{u,i}, y_{u,i})$. Following this labeling process for the entire unlabeled dataset, the classifier model w_k is re-trained using the training dataset $D_{T,k}$, which contains both original labeled data $D_{L,k}$ and pseudo-labeled data $\hat{D}_{U,k}$, where $\hat{D}_{U,k} = \left\{(x_{u,i}, y_{u,i}) \mid x_{u,i} \in D_{U,k},\ y_{u,i} = \arg\max(\hat{\mathbf{y}}_{u,i})\right\}$.

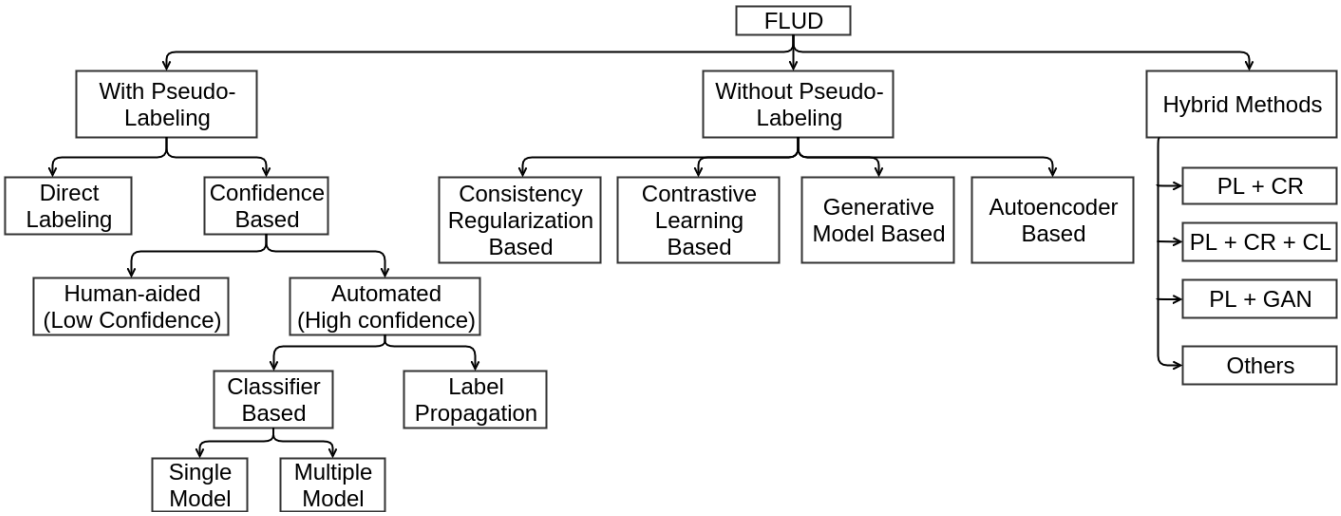


FIGURE 3. The proposed taxonomy of FLUD methods based on utilization of unlabeled data. PL: Pseudo-Labeling, CR: Consistency Regularization, CL: Contrastive Learning.

2) CONFIDENCE BASED PSEUDO-LABELING

In contrast to direct labeling, in confidence-based pseudo-labeling, the label prediction made by a classifier is not directly considered as a pseudo-label. This is because, in situations where the distributions of labeled and unlabeled data significantly differ, a model trained with labeled data tends to produce highly inaccurate label predictions for unlabeled data, leading to potential mislabeling. Accepting these predictions as accurate and incorporating them into model training can compromise model performance. To mitigate this, rather than blindly integrating all predictions into training, employing confidence thresholds to accept predictions above these thresholds as pseudo-labels yields more accurate class predictions for unlabeled data. Consequently, confidence-based approaches are more commonly used within SSFL studies that employ pseudo-labeling [19], [21], [22], [23], [24], [69], [72], [76], [79], [97].

In these studies, unlabeled data is handled in two ways according to its degree of difficulty; low-confidence (those whose labels have low prediction probabilities, i.e., "difficult" data) and high confidence (those whose labels can be easily predicted with high probabilities, i.e., "easy" data). The low-confidence unlabeled data is generally labeled by a human-aided approach (using active learning), while the high-confidence unlabeled data is automatically labeled using classifier-based or label propagation methods, which are explained in more detail below.

a: ACTIVE LEARNING BASED (HUMAN-AIDED) PSEUDO-LABELING

Active learning [98] is an application of machine learning that iteratively queries a human annotator to learn labels for unlabeled data points. In active learning, the unlabeled data sent to a human annotator is selected based on some query strategies (such as confidence scores or information content of data), and then a human annotator assigns labels to these selected data which are then added to the training set.

Similarly, in an SSFL setting, the labels for low-confidence data (which cannot be labeled with high confidence by the classifier trained with the labeled data on a client) are requested from a human annotator [69]. Then, these newly labeled data by the human annotator are added to the training set, and the model is retrained with this training set. In [76], the authors propose a federated active learning with a new sampling strategy, namely LoGo, in which each client determines the querying samples in two steps using local and global models. In this way, they aim to reduce the negative impact of differences between the number of classes in global and local datasets.

b: AUTOMATED PSEUDO-LABELING

Automated pseudo-labeling methods in SSFL are typically applied to high-confidence unlabeled data; this means that a model assigns labels to unlabeled data only when the classifier's confidence level is higher than a predefined threshold τ. Automated pseudo-labeling methods can be divided into two groups as *classifier-based* and *label propagation based* methods.

In *classifier-based methods* [21], [22], [23], [72], [79], [97], each client can use a single model or multiple models to predict the labels of high-confidence unlabeled data. According to the *single model scenario* [21], [23], [79], [97], each client k trains its single classifier model w_k with its labeled data $D_{L,k}$ as in (6), then uses the trained model w_k to predict the labels $\hat{\mathbf{y}}_{u,i}$ for the high-confidence unlabeled data $x^+_{u,i}$. Here, if the predicted probability $\hat{\mathbf{y}}_{u,i}$ for unlabeled data $x_{u,i}$ from the model w_k exceeds or equals a certain threshold τ, i.e., if $\max(\hat{\mathbf{y}}_{u,i}) \geq \tau$, the data is considered high confidence data $x^+_{u,i}$ belonging to the high confidence dataset $D^+_{U,k}$ (i.e., $x^+_{u,i} \in D^+_{U,k}$). In this case, the data is assigned a

label corresponding to the class with the maximum predicted probability, i.e., $y^+_{u,i} = argmax(\hat{\mathbf{y}}_{u,i})$, otherwise it remains unlabeled. The high-confidence pseudo-labeling process is formally defined as follows:

$$\hat{\mathbf{y}}_{u,i} = f_k(x_{u,i}, w^t_k),$$
$$D^+_{U,k} = \left\{(x^+_{u,i}, y^+_{u,i})\, y^+_{u,i} = argmax(\hat{\mathbf{y}}_{u,i}) \text{ if } max(\hat{\mathbf{y}}_{u,i}) \geq \tau\right\}. \quad (9)$$

After labeling all high-confidence data, the current classifier model is re-trained using the training set $D_{T,k}$ that includes the originally labeled data $D_{L,k}$ and the high-confidence pseudo-labeled data $D^+_{U,k}$. Finally, all clients send their local model to the server. Thereby, the performance of the global model is improved by making use of (some of) unlabeled data on the clients.

However, if we consider a scenario in which the distribution of unlabeled data is different from the distribution of labeled data, the single local classifier may not meet the specified confidence level τ when estimating the labels of unlabeled data. Since the local classifier is trained with labeled data, it is more successful in recognizing data containing features of the labeled data distribution. Therefore, the local classifier would label the data containing features of the labeled data distribution with higher confidence, while it would have lower confidence for the data with a different distribution than the labeled data. In this case, data from a rare class might either be disregarded entirely without receiving any label, or it could end up being mislabeled. Moreover, in SSFL, data heterogeneity (non-IID) in clients can lead to a catastrophic forgetting problem, where models may forget knowledge learned from labeled clients while training is being conducted on unlabeled clients [19], [24]. To solve these problems, Li et al. [24] propose a class-balanced adaptive pseudo-labeling (CBAFed) approach for SSFL. In CBAFed, some of the clients have fully labeled data, while others have completely unlabeled data. Each unlabeled client updates its local model with the received global model and generates a fixed pseudo-labeled dataset using the local model and the adaptive thresholds calculated for each class by the server using the client data statistics. In addition, on each unlabeled client, unlabeled data belonging to tail (rare) classes are labeled using a mask function. The pseudo-labeled dataset (fixed and tail) obtained for each unlabeled client is used in the training process. Thus, they aimed to solve the catastrophic forgetting problem. To enhance model performance, they used a residual weight connection method during the training of labeled clients and in aggregating the global model. This method involves skipping connections of the model's parameters between training epochs.

In another study [72], Jiang et al. propose an alternating collaboration for the SSFL method (FedAC) based on two-stage clustering where training is performed alternately between similar and dissimilar clusters of clients in order to produce high quality pseudo-labels and reduce the negative impact of heterogeneous data distribution on global model performance. On the server side, all clients are clustered into two stages according to both their similarity (i.e., clients with similar data distribution are assigned to the same class) and dissimilarity (i.e., clients with different data distributions are assigned to the same class) using the affinity propagation clustering algorithm [99]. In the similarity clustering stage, local models trained with labeled data are sent to cluster heads identified by the server. Each cluster head then aggregates these local models and creates a single cluster model, which is then sent to the clients within the cluster. Each client uses this (similarity) cluster model to label its unlabeled data with higher quality. In the dissimilarity clustering stage, the goal is to learn the global data distribution by clustering clients with dissimilar data distributions and thus mitigate the negative impact of non-IID data distribution on model performance. For tuning iterations of similarity and dissimilarity clustering, they used a multiarmed bandit-based algorithm [100].

Several researchers have proposed using multiple models to decide on labels for unlabeled data. In the scenario of *multiple models* [19], [22], each client can use two or more classifier models to make label predictions with higher confidence for unlabeled data. Consequently, clients can benefit from local models of other clients when labeling their unlabeled data, or they can use multiple models that they keep locally. Here, the intuition is that models trained with different sets of labeled data can capture different patterns and that models with different architectures or different convergence levels may also be more successful in recognizing different features. Therefore, in SSFL studies that use this approach, the labeling process is carried out based on predictions from multiple models using ensemble methods (such as averaging of scores, majority voting, etc.), and thus unlabeled data that cannot be labeled with high enough confidence by a single local model is expected to be labeled with higher confidence.

Label propagation [101] is one of the semi-supervised graph-based methods which propagates the prediction of labels for unlabeled data in the dataset based on a similarity metric between labeled and unlabeled data. In SSFL methods with label propagation [69], [102], instead of training a classification model, a graph is constructed in which each data point (labeled or unlabeled) represents a node and the nodes are connected according to similarity measures, and a set of predictions for the unlabeled data is inferred using this graph.

B. FLUD METHODS WITHOUT PSEUDO-LABELING

FLUD methods without pseudo-labeling aim to directly use unlabeled data without the need for intermediate steps (such as pretraining the client model with labeled data, and then labeling unlabeled data using that model). We categorize these FLUD methods into four main categories considering the techniques and deep learning architectures used: consistency regularization-based, contrastive learning-based, generative method-based, and auto-encoder-based methods.

1) CONSISTENCY REGULARIZATION BASED METHODS

Consistency regularization [103], [104], [105] is a popular approach that benefits from unlabeled data in a semi-supervised setting that aims to make model predictions invariant against perturbations in input data or model function. The consistency regularization method enforces the model to produce similar results for perturbed versions of the same input data. Similarly, when some perturbations occur on the network architecture, it regularizes the original and perturbed models to output similar predictions. There exist some consistency regularization methods (such as π-model [104], Temporal Ensembling [104], and Mean-Teacher [106]), which can be directly integrated or adapted into FLUD settings.

In some FLUD studies based on consistency regularization (e.g. [20], [71]), for labels in the client scenario (Case 1 in Figure 2 (a)), each client k updates its local model simultaneously, minimizing both supervised loss ℓ_s (e.g. cross entropy) and unsupervised loss (e.g. Kullback-Leibler (KL) divergence or mean squared error), as given in:

$$\mathcal{L}_k = \frac{1}{|D_{L,k}|} \sum_{x_i, y_i \in D_{L,k}} \ell_s(f(x_i, w_k), y_i) + \frac{1}{|D_{U,k}|} \sum_{x_{u,i} \in D_{U,k}} \left|\left| f(x_{u,i}, w_k),\ f(\pi(x_{u,i}), w_k) \right|\right|^2, \quad (10)$$

where $f(x_{u,i}, w_k)$ is a model architecture that outputs predictions for an unlabeled input $x_{u,i}$ using the model parameters w_k of the client k, and $\pi(.)$ is a perturbation function that applies data augmentation methods (e.g., cropping, rotation, and flipping) on the unlabeled data.

In some FLUD studies (such as [20], [81]), for labels in the server scenario (case 2 in Figure 2 (b)), each client k updates its local model with only its unlabeled data by minimizing the unsupervised consistency loss given in (11) and the server updates the global model with its labeled data $D_{L,server}$ by minimizing the supervised loss $\mathcal{L}_{server}$, as given in:

$$\mathcal{L}_k = \frac{1}{|D_{U,k}|} \sum_{x_{u,i} \in D_{U,k}} \left|\left| f(x_{u,i}, w_k),\ f(\pi(x_{u,i}), w_k) \right|\right|^2, \quad (11)$$

$$\mathcal{L}_{server} = \frac{1}{D_{L,server}} \sum_{x_i, y_i \in D_{L,server}} \ell_s(f(x_i, w_g), y_i). \quad (12)$$

In [19], instead of simultaneously minimizing the losses as mentioned above, the model parameters are represented with two different sets: one for supervised learning (σ) and the other for unsupervised learning (ψ). During supervised learning, parameters in σ are updated while the parameters in ψ are fixed. Similarly, during unsupervised learning, the parameters in ψ are updated while the others are fixed. In this way, they aim to obtain a better model accuracy by preventing knowledge forgetting from labeled data.

In [86], the authors aimed to overcome the uneven reliability of local client models due to the non-IID distribution and different amounts of labeled data. For this, they propose a federated random sampling consensus (RSCFed) framework in which labeled clients apply supervised learning while unlabeled clients use the mean-teacher-based consistency regularization method. In this framework, a number of subsets is generated by random sampling over clients. In each subset, clients perform their local training based on the supervised learning or consistency regularization method, depending on the type of client. After training, a subconsensus model is obtained using distance-reweighted model aggregation (DMA), which aggregates the updated local models by weighting them based on closeness to the average of the local models in the subset. In this study, they also present the ability of RSCFed in the case where all clients have partially labeled data. In [107], a consistency regularization method based on the mean teacher is used to leverage data from unlabeled clients. Additionally, to achieve a fairer model aggregation, a novel approach based on multi-sampling is proposed instead of the standard FedAvg. In this approach, dynamic weighting is applied based on the distance between local models and the global model, and the generated compromise models are aggregated to form the final global model. In FedCD [87], they demonstrate a new SSFL approach based on dual teacher distillation and class awareness balance to deal with class imbalance and knowledge gap between clients. The dual teacher distillation aims to make the predictions of the student model consistent using both global and local teachers. Erroneous information from teachers is filtered through the knowledge purification approach, so that the student model learns more accurate and pure information. In addition, by identifying rare class examples through a proxy model, these samples are given more weight in the loss function, aiming for the local model to learn all classes equally.

2) CONTRASTIVE LEARNING-BASED METHODS

Contrastive learning [25] is a self-supervised learning method that allows learning good representations from data without labels, by teaching the model how to distinguish data pairs from each other. In contrastive learning, the aim is to differentiate similar samples from different samples by minimizing the distance between the representations of the similar samples while maximizing the distance between the representations of dissimilar samples. In a basic contrastive learning scheme, a sample is chosen as the anchor (query sample), and samples with a distribution similar to this anchor are called positive samples and samples with different distributions are called negative samples. If the data is unlabeled, positive samples are obtained by applying augmentation to the anchor, while negative samples are selected from the remaining samples. The model (consisting of encoder and projection networks) tries to minimize the distance between the anchor and positive samples in the latent

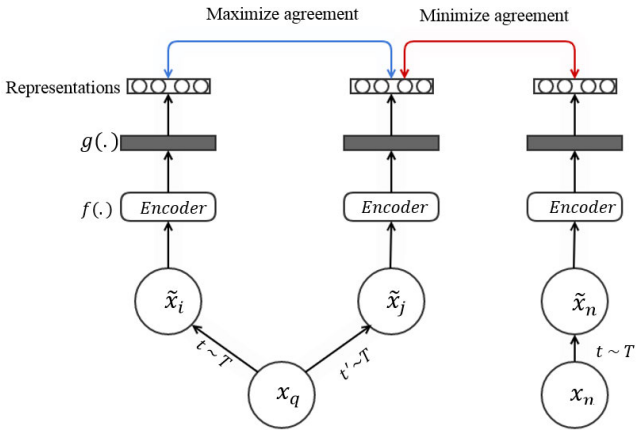


FIGURE 4. A simple illustration of contrastive learning. t and t' are two augmentation operators which are sampled from the augmentations T. $\tilde{x}_i$ and $\tilde{x}_j$ are the positive pair of a query sample x_q while x_n is a negative sample. $f(.)$ is a encoder network and $g(.)$ is a projection network.

space, which belong to the same distribution, and at the same time maximize the distance between the anchor and negative samples, as shown in Figure 4.

There are three types of contrastive learning methods: instance-based, cluster-based, and multiview coding [25]. The FL methods using contrastive learning techniques we reviewed focused more on instance-based contrastive learning techniques (e.g. PIRL [108], Moco [109], SimCLR [110]). PIRL is an instance-based contrastive learning method using a memory bank which aims to learn image representations that do not covary with the image transformations. MoCo is another contrastive learning method that uses a dynamic dictionary instead of a memory bank. SimCLR is one of the most popular contrastive learning techniques, which does not use a memory bank or an alternative architecture. It attempts to learn useful representations from the samples by minimizing the loss between their non-linear representations, which are applied to different augmentation techniques.

Contrastive learning methods have been integrated into the FLUD environment in some studies [26], [63]. However, directly combining these methods with federated learning can pose some challenges [26]. For example, each client may have a different data distribution, and local models trained with these data will then learn different representations from each other. This causes the global model to fail to learn the common representations in the clients. In addition, the representations learned by the local models may not be similar even if the data distribution on the clients is the same due to the random selection of the training data. In this case, it negatively affects the performance of the global model. To handle these problems, Zhang et al. [26] propose a federated contrastive averaging method (FedCA) in which they use dictionary and alignment models. In this study, each client has a local dictionary (generated by temporal ensembling) and two models (local model and alignment model). The local model is trained with the client's local data and the global dictionary using contrastive learning. At the same time, the difference between the representations of the local and aligned models is minimized. After training, each client sends its local model and local dictionary to the server. The experiments show that this method yields better results than the methods that directly combined unsupervised and self-supervised methods (e.g. SimCLR) with FL. However, there may be privacy issues due to the sharing of data representations on clients.

In [63], the authors propose a two-stage federated contrastive learning framework for the diagnosis of dermatological diseases; where in the first stage the main local model is trained using contrastive learning, and then the trained model is used as an initialization model for fine-tuning in the second stage. In this framework, each client has a momentum contrast (MoCo) model, which is used to extract features of the data in the clients, along with the main local model. In each round, every client extracts the features of the data in their current mini-batch by using the MoCo model and sends them to the server for anonymization, then it downloads these anonymized features as remote features. After that, it trains its own main model with these remote features and its local features, considering as negatives (i.e., dissimilar) and positives (i.e., similar), respectively, and uploads the updated main local model and local features to the server for the initialization of the next round. They demonstrate that their framework improves the results compared to state-of-the-art methods. They also reduce privacy leakage due to the anonymization of features. In [89], the authors use a Siamese network and also apply contrastive learning to learn meaningful features from unlabeled data for a downstream task.

3) AUTOENCODER BASED METHODS

An autoencoder [111] is an unsupervised neural network model that can automatically learn the latent features of the data. It consists of three parts, encoder, code, and decoder, illustrated in Figure 5. In the encoder part, the encoder model e_θ learns how to encode the input data x into a low-dimensional representation containing the latent features and the representation of the encoded input data z resides in the code layer. In the decoder part, the decoder model d_ϕ tries to reconstruct the input data $\hat{x}$ from the encoded representation z. The autoencoder aims to reconstruct the input data as an output with the least possible loss as given in the following equation:

$$\mathcal{L} = ||x - \hat{x}||^2 = ||x - d_\phi(z)||^2 = ||x - d_\phi(e_\theta(x))||^2. \quad (13)$$

In [29], the authors propose an SSFL method based on auto-encoder, where each client keeps a local auto-encoder model to learn representations of its own data, and a classifier model (which can be on the server or client) is trained with data encoded by the auto-encoder model. In [112], Cha et al. propose a FL method based on autoencoders where data is vertically partitioned into different sites (i.e., hospitals).

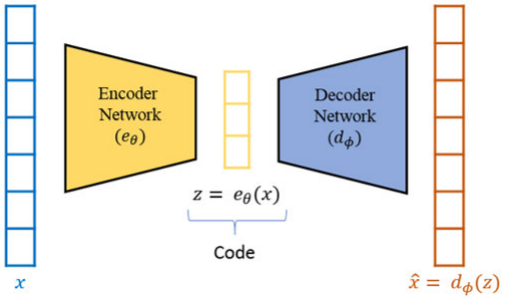


FIGURE 5. An illustration of autoencoder model.

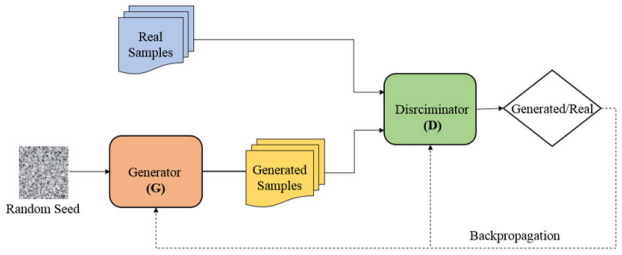


FIGURE 6. An illustration of Generative Adversarial Network.

At each site, an overcomplete autoencoder model is trained in which the dimension of the latent space is equal to or greater than the input space, and the latent data is sent to the server. The data received are aggregated at the server side and they are used for training of the global model. In [113], the authors present a novel FL approach for intrusion detection in IoT networks using the Shrink Autoencoder [114] as a feature representation method to obtain a new feature space, where the Centroid algorithm [115] is used for anomaly detection.

4) GENERATIVE METHODS

Generative models [96], which are divided into two categories such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), try to learn the distribution of data points in a training dataset and generate new data points with this distribution rather than classifying the data points.

GANs [116] are deep learning-based generative models which have a similar idea to the Min-Max two-player game. In GANs, there exist two competing networks named generator and discriminator, where the generator produces fake samples to deceive the discriminator, while the discriminator tries to distinguish the real samples from the generated ones. The purpose of the GAN model is to allow the generator to learn the distribution of the training data without accessing them. The GAN training procedure is illustrated in Figure 6. Consequently, the generator G and discriminator D networks are trained simultaneously and are updated by backpropagation. In each round, the generator tries to produce samples with a distribution closer to the real data distribution until the discriminator cannot distinguish the generated samples from the real ones. Thus, at the end of the training procedure, the generator learns the real data distribution of the training samples.

In some studies in FL, GAN models have only been used to augment labeled data. For example, in [65], the authors proposed a GAN based FL framework for COVID-19 diagnosis, and the generator and discriminator model parameters are shared with a cloud server for aggregation. In another study [64], Zhang et al. introduced a federated differentially private GAN framework to solve the problem of non-IID data distribution in FL by augmenting COVID-19 data. Unlike [65], here only the parameters of the generator model are alternated between the server and the clients. After the FL process finishes, the augmented data is used to classify Covid, Normal, and Pneumonia.

However, there are also GAN-based studies in FL that aim to make more use of unlabeled data. Xu et al. [117] introduce a federated traffic classification framework based on GAN, called FGAN-AC, which can generate synthetic traffic data (unlabeled) in a decentralized manner to capture the overall distribution of data. Synthetic traffic data is automatically labeled using the deep clustering method (i.e., DEC [118]) and used to train a global classification model. Thus, they solve both the limited labeled data problem and the data privacy issue. In [119], the authors propose a synthetic data-aided FL (SDA-FL) approach in which each client generates unlabeled synthetic data locally and sends them to the server to form a global synthetic dataset before starting training. During the FL rounds, the global (unlabeled) synthetic data is pseudo-labeled by the corresponding local models uploaded to the server side, and then the pseudo-labeled synthetic dataset is used to train both the global model and local models. In this framework, the pseudo-labels are improved in each round using the updated local models. In [73], Wijesinghe et al. propose a FL framework based on GAN (UFed-GAN) with unlabeled data on wireless sensor networks. Instead of training a full GAN model at each sensor node, UFed-GAN trains a discriminator model to capture the data distribution at the sensor side, then at the server side, the full GAN model training is performed using the obtained discriminator model, and unlabeled synthetic data is generated with the generator part of the full GAN model.

VAEs [120] are deep generative models that provide regularized latent space representations that are different from standard AEs. In VAEs, latent space is continuous and complete, which ensures generativity. This means that each latent variable in the latent space produces a meaningful output when decoded, and that the latent variables close to each other in the latent space do not produce completely different outputs. In VAEs, the output of the encoder is not deterministic as in AE; on the contrary, it is stochastic. As seen in Figure 7, the encoder e_θ encodes every input x as a distribution $N(\mu_x, \sigma_x)$ instead of a latent vector z. The latent vector is sampled from the encoded distribution, then

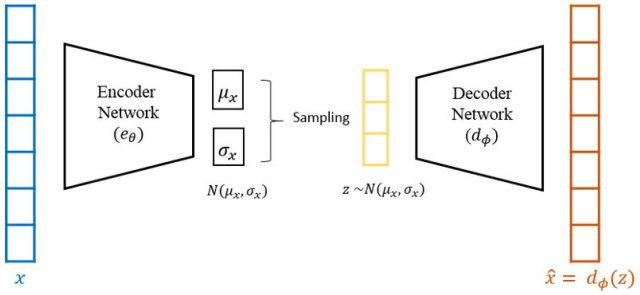


FIGURE 7. An illustration of variational autoencoder.

the decoder d_ϕ decodes the latent vector to reconstruct the input.

In VAEs, the loss function consists of a reconstruction loss term and a regularization loss term, as given in the following equation:

$$\begin{aligned}\mathcal{L} &= \left|\left|x - \hat{x}\right|\right|^2 + KL\left[N\left(\mu_x, \sigma_x\right),\ N\left(0, 1\right)\right] \\ &= \left|\left|x - d_\phi\left(z\right)\right|\right|^2 + KL\left[N\left(\mu_x, \sigma_x\right),\ N\left(0, 1\right)\right].\end{aligned} \tag{14}$$

The reconstruction loss is the same as in AE, which is the difference between the input x and reconstructed output $\hat{x}$. The regularization loss is the KL divergence between the encoded distribution and the standard normal distribution, which aims to regularize the latent space by forcing the encoded distribution to be close to a standard normal distribution.

In [31], the authors propose a framework to predict the popularity of content for caching decisions based on peer-to-peer FL using data such as user request histories and contextual information. In this approach, they use a collaborative filtering-based VAE (CF-VAE) model that learns latent representations of user preferences and content popularity to predict missing values (unrequested content) in the user by content-request matrix. Unlike [31], which is focused on peer-to-peer FL, this study [30] proposes a federated caching scheme based on a mobility-aware caching replacement policy. Furthermore, more emphasis was placed on the sensitivity to contextual information by using the contextual adversarial autoencoder (C-AAE) model instead of CF-VAE in the construction of the user content request matrix.

C. HYBRID METHODS

In numerous FLUD studies (e.g., [19], [30], [69], [71], [75], [77], [82], [83], [85], [88], [90], [91], [92], [121]) the aforementioned methods or other learning paradigms are used in combination to make more effective use of unlabeled data. Thus, it aims to improve the performance of the global model by integrating multiple strategies. Specifically, in these studies, the pseudo-labeling and consistency regularization methods are often combined [19], [71], [74], [75], [77], [82], [83], [85], [90], [91], [121]. Among them, [19], [71], [74], [77] employ diverse models for pseudo-labeling, while others (e.g., [82], [90], [91]) rely on a single model. In one approach [83], predictions are made for various augmentations of unlabeled data, and the highest class prediction obtained by aggregating these predictions is considered as the label.

Lee et al. [92] focused on the scenario in which the labeled data reside only on the server and the clients have only the unlabeled data. In this study, client-specific thresholding is applied for pseudo-labeling along with sharpness-aware consistency regularization. The client-specific threshold is computed by averaging the predictions of the global model on unlabeled data for each class. Predictions that meet this threshold are then converted into pseudo-labels for model training. Additionally, high-confident pseudo-labels are identified using a sharpness-aware approach. In this process, a high fixed threshold is applied to obtain trustworthy labels, and strong data augmentation techniques are used. These augmented samples are then fed into both the original local model and an aggressively perturbed version to enforce consistency in model predictions. Finally, model aggregation is performed based on the learning levels of local models.

In [74], Zhu et al. propose a new SSFL approach (TGAFed) based on the FedMatch method; however, instead of using a fixed threshold for weighting pseudo-labels, it employs a truncated Gaussian distribution. Consequently, a weighting function is defined that assigns higher weights to samples whose maximum prediction probability exceeds the Gaussian mean, considering the maximum probability value and the Gaussian distribution parameters. In this way, the aim is to improve the pseudo-labels in terms of both quantity and quality. Furthermore, a distribution alignment approach has been applied to improve the consistency of the model's predictions with the labeled data.

Cho et al. [77] introduce a new SSFL framework called FedLABEL, which includes pseudo-labeling and consistency regularization approaches based on local and global model selection. In the pseudo-labeling process, a model is first selected by comparing the confidence scores of the local and global models. The logit with higher value that exceeds the predefined threshold in the selected model is assigned as the pseudo-label. Consistency regularization is applied when the non-selected model also predicts the same pseudo-label. During this process, the KL divergence between the output of the updated local model and the output of the non-selected model is minimized. This ensures that the local model also learns from the unselected model.

In [121], pseudo-labeling and consistency regularization techniques are also used together to overcome the class imbalance problem and improve model performance. They introduce an exploitation maximization method of unlabeled data for federated semi-supervised learning by integrating three loss functions into a mean teacher model proposed in FixMatch [122]. In the proposed method, em-FSSL, there are two types of clients: labeled and unlabeled. In labeled clients, training is performed using only labeled data. In unlabeled

clients, the teacher and student models make predictions on the weakly augmented and strong versions of the unlabeled data, respectively. The predictions made by the teacher model are also converted into pseudo-labels. Using high-confidence samples (those larger than a fixed threshold), a focal loss and label distribution-aware marginal loss are computed to increase the weight of difficult samples and give larger margins to minority classes. Thus, they aim to reduce the class-imbalance problem. On the other hand, low-confidence samples are used to improve the generalization of the model by comparing the diversity among them.

In [88], the authors propose an SSFL framework called FedRVR, with data- and model-guided learning on the client side and relation-guided directional aggregation on the server side. Unlabeled clients use unlabeled data through superior and inferior global models. The superior model generates pseudo-labels, which are used to train the local model. Additionally, the representations of local models are regularized using those from the inferior global model. In labeled clients, in addition to updating the local model, a relation predictor model is trained using pairs of local models from different time steps. On the server side, an undirected intrinsic graph is first constructed from the received client models. The relation predictor models from the labeled clients are then combined to create a global predictor model that determines the relationships between the models. Based on these relations, better models are aggregated with higher weights to obtain a superior global model. An aggregation in the opposite direction is also performed to generate an inferior model.

Another study [69] uses active learning and label propagation together while utilizing unlabeled data. In [75], the concept of peer learning is integrated with FL. This involves aggregating similar peers' models at the server side to ensure anonymity and sending them to the target client for labeling support. On the client side, local training minimizes both semi-supervised loss and consistency loss. Moreover, in [85], labeled clients train their models with labeled data, while unlabeled clients use distilled soft labels from labeled clients to generate labels for their data. They optimize their models by minimizing both the loss-based consistency regularization and the KL divergence between the distilled soft labels and the produced soft labels.

Although many studies have combined pseudo-labeling and consistency regularization techniques, several others have explored different approaches simultaneously. For example, in [30], the VAE and GAN models are merged for a content caching scheme. In another study, [117], GAN and pseudo-labeling methods are used for the synthesis and classification of traffic data. In [73], pseudo-labeling is used for labeling of synthetic data generated by the GAN model. Furthermore, in [81], consistency regularization and pseudo-labeling are integrated with contrastive learning concepts. Moreover, in [123], the local model is optimized by minimizing supervised loss alongside contrastive loss. Furthermore, [27] uses contrastive learning to extract meaningful features from unlabeled data, subsequently transferring them to a downstream task to reduce reliance on labeled data.

V. APPLICATIONS OF FLUD

Methods utilizing unlabeled data in FL are used in many areas such as computer vision, speech processing, IoT (sensor data), networking, and healthcare. In this section, we first list the most commonly used datasets for different data modalities. Then, we give a table of representative FLUD methods in the literature for various application areas. Finally, we give a brief quantitative comparison of FLUD methods for the most widely used image dataset (CIFAR-10).

A. DATASETS

Some of the datasets used for different applications are listed by data modality in Table 1. We can observe from the datasets used in FLUD that the image modality has been studied the most. Not only general-purpose image datasets (e.g., MNIST and CIFAR-10), but also medical image datasets (e.g., CheXpert, HAM10000 and ISIC-2018) have been widely utilized. The main reason for the widespread use of general purpose image datasets in particular is that they are publicly available, large, and standardized, making it easier to make performance comparisons between different studies.

In addition, the availability of a large number of deep learning models that are already pre-trained on visual data encourages the adaptation of these models to FL environments, which in turn increases the use of image datasets in FLUD studies. In addition, the need for data privacy in medical applications has emphasized the applicability of FLUD methods in this field, which has further increased interest in the image modality.

However, the use of other modalities such as sensors, audio, or network traffic in FL is currently relatively limited. This suggests that there is still a significant imbalance in FL research in terms of modality diversity and that the work in this area needs to expand to a wider variety of data modalities.

B. METHODS

Representative lists of FLUD methods in the literature, grouped by data modalities and application areas, are given in Table 2, Table 3 and Table 4. In these tables, we also provide information on the location of unlabeled data and highlight the main contributions of each method.

We can observe from Table 2, which lists methods for image modality, that there are many FLUD-based studies for computer vision applications, especially for image classification and object detection. For image classification, in [19], [20], [21], [22], [71], [81], [82], [83], [89], and [91], different FLUD approaches were proposed for different scenarios (i.e., in terms of the location of the labeled data or different data distributions) using general image datasets (e.g. MNIST, CIFAR-10, etc.) as benchmarks. In [78], a simple FLUD method is used for image recognition. On the other hand, Chiu et al. [161] proposed a new FLUD method for object detection (traffic signs, signboards, and bus stations), where

TABLE 1. Datasets and application areas used in FLUD methods in the literature, grouped by modality.

Modality	Application	Datasets
Audio	Audio Recognition	Ambient Context [124], Speech Commands [125], VoxForge [126]
	Speech Emotion Recognition	IEMOCAP (audio data only) [127], MSP-IMPROV (audio data only) [128]
GPS/Geo	Travel Mode Identification	GeoLife [23]
	Building Damage Assessment	2018 Lower Puna Eruption Satellite Imagery [129]
Image	Image Classification	CIFAR-10/100 [130], CINIC-10 [131], EMNIST [132], FASHION-MNIST [133], FEMNIST [134], MNIST [135], Mini-ImageNet [136], SVHN [137]
	Traffic Sign Recognition	GTSDB [138]
Medical Image	Medical Image Classification	AtlasDerm [139], BloodMNIST [140], ChestCovid [141], CheXpert [142], ChestX-ray8 [143], Covid-19 [144], DarkCovid [145], DarkDerm [63], DermaMNIST [140], DermNet [146], Derm7pt [147], HAM10000 [148], ISIC-2018/2019/2020 [148]–[150], OrganAMNIST [140], PAD-UFES [151], PathMNIST [140], RSNA ICH [152], VinDr-CXR [153]
Network	Network Traffic Classification	ISCXVPN2016 [154]
Sensor	Activity Recognition	DG [155], MobiAct [156], Opp [157], PAMAP2 [158], WISDOM [159]
Tabular	Content Caching	MovieLens [160]

they periodically swap client models among themselves instead of aggregating client models in each round.

In another study [84], Zhang et al. proposed an SSFL framework, called Semi-FedDA, that automatically assesses building damage using satellite imagery. In this framework, low Earth orbit (LEO) satellites have only unlabeled data, while the ground base station (BS) has a small amount of labeled data. To train the models in satellites, they used the approach proposed in [91]. In order to reduce the communication cost between the BS and the satellites, an intra-orbit aggregation method is proposed, where only a source satellite from each orbit communicates with the BS for global aggregation.

In Table 3, which lists methods for the medical image modality, we can see that different FLUD methods have been proposed for healthcare applications mainly to diagnose diseases using medical image classification [24], [27], [63], [64], [65], [75], [76], [85], [86], [87], [88].

In Table 4, we can see a list of representative methods for audio, network traffic, sensor, and tabular data modalities, where the application areas are also indicated. For speech recognition tasks, various classifier-based automated pseudo-labeling methods have been proposed. For example, in [162], a single-model approach is used for pseudo-labeling, while [80] utilizes multiple models for the same purpose. Although IEMOCAP and MSP-IMPROV are multimodal datasets containing audio, video, and textual modalities, the studies [80], [162] focus exclusively on audio-derived features for training speech emotion recognition models.

In [117], a GAN-based FLUD method is used for the synthesis and classification of network traffic. There are also several IoT-oriented applications that utilize sensor data [29], [69].

C. QUANTITATIVE COMPARISON

Quantitative comparison of FLUD methods is significantly challenging due to the diversity of data modalities used, differences in model architectures, and heterogeneity in data distribution. However, despite these challenges, a systematic comparison was carried out in order to analyze the methods in detail. To make the comparison fair and meaningful, we tried to evaluate the methods under as similar conditions as possible. For this purpose, we chose the image modality that has been the most studied in the literature and compared the performance of the methods by analyzing the results on the CIFAR-10 dataset (under different non-IID distributions), which is the most commonly used dataset in this modality. The compared methods include techniques based on pseudo-labeling, consistency regularization, and self-supervised learning approaches. The comparison results are detailed in Table 5. The results of the analysis show that methods based on self-supervised learning, e.g., FedCA and FedU, can achieve higher accuracy levels than other approaches without using any labeled data (except for the fine-tuning phase) and perform quite well in this respect. The observed performance differences among pseudo-labeling approaches are related to the auxiliary techniques used, structural differences in model architectures, and especially the degree of non-IID data distribution commonly encountered in distributed systems. The diversity in the model architectures used and especially the degree of distribution of the data away from IID are among the main constraints in the comparative performance analysis of the methods. In this context, it is of great importance to establish a common and standardized evaluation protocol to evaluate the performance of FLUD methods in a fair and reproducible way.

VI. DISCUSSION: CHALLENGES AND FUTURE RESEARCH DIRECTIONS

FL, which uses unlabeled data, has found application in numerous machine learning challenges. The scarcity of labeled data poses a significant challenge, particularly in domains that demand specialized knowledge for data labeling, such as medical applications. Examining Table 2

TABLE 2. Overview of FLUD methods in the literature focusing on the *image modality*, organised by major application areas. The table also specifies the placement of labelled (D_L) and unlabelled (D_U) data, and whether a novel aggregation strategy is introduced.

Application	Acronym	Dataset(s)	D_L/D_U location	New Agg.	Main Contributions / Methods
Building Damage Assessment	Semi-FedDA [84]	2018-Lower Puna Eruption Satellite Imagery	Case 2	Yes	– Proposes intra-orbit aggregation scheme to reduce communication cost
Image Classification	FedMatch [19]	CIFAR-10 Fashion-MNIST	Case 1, 2	No	– Client/data-level **consistency regularization** – Parameter decomposition – **Agreement-based** pseudo-labelling (helper agents)
	FedSiam [20]	MNIST CIFAR-10 SVHN	Case 1, 2	No	– Uses a Siamese network and momentum update – **Consistency regularization** for unlabelled data and **cross-entropy** loss for labelled data – Layer selection to reduce communication cost
	RC-SSL [21]	MNIST CIFAR-10	Case 2	Yes	– **Two-level pseudo-labelling** – Min–max client selection (malicious filtering) – Symmetric quantisation; **geometric-median** aggregation
	FedTriNet [22]	MNIST Fashion-MNIST SVHN	Case 1	No	– **Multiview pseudo-labeling** with three networks – Dynamic high-confidence data selection
	SemiFed [71]	CIFAR-10 SVHN	Case 1	No	– **Consistency regularization + pseudo-labelling** – Pseudo-labels via maximum model agreement
	FedCon [81]	MNIST CIFAR-10 SVHN	Case 2	No	– Online/target dual-network design – Minimizes **supervised** and **consistency** loss at server
	Zhang et al. [82]	CIFAR-10 SVHN EMNIST	Case 2	Yes	– **Consistency regularization** with group normalization – **Pseudo-labelling** using local model – **Grouping-based** averaging
	FedU [89]	CIFAR-10 CIFAR-100 Mini-ImageNet	Case 4	No	– **Siamese network** (online/target) with **contrastive loss** – Divergence-aware predictor update
	FedMix [83]	CIFAR-10	Case 2	Yes	– Combines multiple global models – **Consistency regularization + pseudo-labeling** on clients – Proposes **FedFreq** aggregation (weights by training frequency)
	SemiFL [91]	CIFAR-10 SVHN	Case 2	No	– Generates **pseudo-labels** (based on global model) to build high-confidence set – Mixup sampling from high/low-confidence data
	FedCA [26]	CIFAR-10 CIFAR-100 Mini-ImageNet	Case 4	No	– Applies **contrastive learning** (dictionary/alignment) – Uses **temporal ensembling**; **transfer learning** – Linear evaluation + semi-supervised protocol
	FedAC [72]	Fashion-MNIST SVHN CIFAR-10 FEMNIST	Case 1	No	– **Similarity clustering** for pseudo-labels – **Dissimilarity clustering** to mitigate non-IID
	UFed-GAN [73]	CIFAR-10 SVHN Fashion-MNIST	Case 1	No	– **Discriminator** learns node data distribution – **Generator** synthesises per-node data – Pseudo-labelling of synthetic data
	em-FSSL [121]	CIFAR-10 CIFAR-100	Case 3	No	– Mean-teacher (FixMatch) backbone – **High-confidence** samples via pseudo-labels; **low-confidence** samples via consistency regularization
	FL^2 [92]	CIFAR-10 CIFAR-100 SVHN	Case 2	Yes	– **Client-specific adaptive thresholds** from global model – **Sharpness-aware consistency** (robustness) – Adaptive aggregation reflecting client learning level
	TGAFed [74]	CIFAR-10 CIFAR-100 CINIC-10	Case 1	No	– **Pseudo-labelling + consistency regularisation** (helper models) – Pseudo-labels filtered by truncated Gaussian weighting
Traffic Sign Recognition	FedSem [78]	GTSDB	Case 1	No	– Uses **direct labelling** method

TABLE 3. Overview of FLUD methods in the literature focusing on the *medical image modality*, organised by major application areas. The table also specifies the placement of labelled (D_L) and unlabelled (D_U) data, and whether a novel aggregation strategy is introduced.

Application	Acronym	Dataset(s)	D_L/D_U location	New Agg.	Main Contributions / Methods
Medical Image Classification	FedCL [63]	ISIC-2019 AtlasDerm DermNet DarkDerm	Case 4	No	– **Contrastive learning** with momentum contrast model to build memory banks of local and remote features during pre-training – Fine-tuning with limited labeled data for classification
	FedIRM [85]	RSNA ICH ISIC 2018	Case 3	No	– **Inter-client relation matching**: labeled clients assist unlabeled clients' training
	FedMoCo [27]	**Pre-training** CheXpert ChestX-ray8 VinDr-CXR **Downstream task** COVID-19	Case 4	No	– Uses MoCo model for intra-node **contrastive learning** – Proposes **metadata transfer** for inter-node augmentation – Presents **self-adaptive aggregation** with representational similarity – Uses linear classification protocol to evaluate model performance
	FedPerl [75]	ISIC19 HAMM10000 Derm7pt PAD-UFES ISIC20 for testing	Case 1, 2	No	– Clusters similar clients into one community – Introduces **dynamic peer learning** with anonymized peer models to generate **pseudo-labels** – Minimizes **semi-supervised** and **consistency regularization** loss during the training
	RSCFed [86]	SVHN CIFAR-100 ISIC2018	Case 1, 3	Yes	– Uses **mean-teacher based consistency regularization** – Propose a client sampling strategy to generate sub-consensus models – Propose a new aggregation method for sub-consensus models
	FedLABEL [77]	EMNIST CIFAR-10/100 OrganAMNIST BloodMNIST	Case 1	No	– Applies **pseudo-labeling** based on model selection (local or global) with confidence score – Uses **consistency regularization** to utilize knowledge from both the local and non-selected models.
	LoGo [76]	CIFAR-10 SVHN PathMNIST DermaMNIST OrganAMNIST	Case 1	No	– Uses **active learning** – Each client has his own annotator – Proposes a **cluster-based sampling** with two models to annotate unlabeled data
	CBAFed [24]	SVHN CIFAR-10/100 FASHION-MNIST ISIC2018	Case 3	Yes	– Proposes a class-balanced **adaptive pseudo-labeling** method – Applies a **residual weight connection method** in supervised training and aggregation of global model
	FedCD [87]	HAM10000 RSNA-ICH	Case 3	No	– Introduces **dual teacher distillation**, an advanced consistency regularization method – Mitigates non-IID distribution issues in SSFL using a **class awareness balance** module
	FedRVR [88]	CIFAR-10/100 ISIC-2018	Case 1, 3	Yes	– Applies two different aggregation approaches based on the relation graph of local models – Uses both **pseudo-labeling** and **feature-based regularization** on unlabeled clients using two different global models

reveals a growing number of articles focused on medical image classification using FLUD techniques. This surge is attributed to the sensitivity of medical data, leading to regulatory limitations on its transfer. An interesting observation is that existing FLUD methods in the literature predominantly fall into the Case 1 and Case 2 scenarios. In Case 1, all clients possess both labeled and unlabeled data, while in Case 2, clients only have access to unlabeled data. Another observation is that while Case 1 and Case 3 are more suitable for pseudo-labeling methods, Case 4 is more appropriate for methods that do not use pseudo-labeling (especially contrastive learning). On the other hand, Case 2 scenario appears in studies where hybrid methods combining pseudo-labeling and consistency regularization are used. We have also observed that recent SSFL studies in the medical field have focused more on the Case 3 scenario. Additionally, we believe that it would not be fair to directly compare the reported experimental results of the methods

TABLE 4. FLUD methods for non-image modalities, organised by application areas. The table also specifies the placement of labelled (D_L) and unlabelled (D_U) data, and whether a novel aggregation strategy is introduced.

Modality	Application	Acronym	Dataset(s)	D_L/D_U location	New Agg.	Main Contributions / Methods
Audio	Audio Recognition	FedSTAR [79]	Ambient Context Speech Commands VoxForge	Case 1	No	– Applies **self-training** with pseudo-labeling based on adaptive confidence threshold
	Speech Emotion Recognition	Tsouvalas et al. [162]	IEMOCAP	Case 1	No	– Applies **self-training** strategy to generate pseudo-labels for unlabeled data based on a constant threshold
	Speech Emotion Recognition	Semi-FedSER [80]	IEMOCAP MSP-IMPROV	Case 1	Yes	– Augmentation (FixMatch) – **Multiview pseudo-labeling**; uncertainty-aware selection – Uses **SCAFFOLD** aggregation
Network traffic	Network Traffic Classification	FGAN-AC [117]	ISCXVPN2016	Case 1 or Case 4	No	– Generates network traffic data using **GAN**, and proposes automatic classification framework for the generated data
Sensor (time-series)	Activity Recognition	FedHAR [69]	MobiAct WISDOM	Case 1	No	– Applies **active learning** for low confidence unlabeled data – Applies **label propagation** for remaining unlabeled data
	Activity Recognition	Zhao et al. [29]	Opp DG PAMAP2	Case 2	No	– Applies **unsupervised learning** using autoencoder model at client side
Tabular	Content Caching	MPCF [30]	MovieLens	Case 1 or Case 4	Yes	– Uses **Context-aware Adversarial Autoencoder** to predict content popularity – Modifies FedAvg that weights models based on the location of connected vehicles
	Content Caching	PPFC [31]	MovieLens	Case 1 or Case 4	Yes	– A vehicle is selected as a parameter server instead of a roadside unit – Uses a Collaborative Filtering based **VAE** model – **Dual-weighted aggregation** scheme
Trajectory (GPS)	Travel Mode Identification	Zhu et al. [23]	GeoLife	Case 1	Yes	– Designs a new DNN by combining CNN, GRU, and FNC – **Pseudo-labeling** based on fixed threshold – Proposes **grouping-based aggregation**

given in the FLUD taxonomy with each other because different test environments were used in each work.

We also observe that integrating unlabeled data into the FL training process can improve the performance of the global model; however, it can also introduce certain challenges. Especially when the data distribution is non-IID, approaches that use pseudo-labeling tend to bias the labels produced by clients towards the dominant class of the locally available labeled data. This can lead to error-prone label generation. Moreover, even if highly accurate pseudo-labels are obtained, these labels can lead to overfitting of local models to certain classes. Thus, the generalization capacity of the global model, which is formed by aggregating these biased local models, may decrease.

Below we provide some insights about potential future research directions for FL which leverage unlabeled data:

- **Applications using Biometric Data:** The applicability of FLUD methods is expected to extend to a wide range of machine learning problems previously addressed in a centralized manner. However, in scenarios that involve sensitive biometric data, the development of new FLUD algorithms becomes imperative. For example, applications such as face recognition [162], [163] and face recognition presentation attack detection [164] rely heavily on facial data, which are considered personal and subject to increasingly stringent data protection regulations. Thus, the sharing of facial data requires explicit consent from the individuals represented in the datasets. Currently, there are a limited number of studies in the literature investigating the use of unlabeled data in federated face recognition.
  Another use case is the recognition of emotions from speech [80], [162] and facial expressions. Speech data on edge devices are predominantly unlabeled due to the high cost of annotation and limited incentives or expertise among users. Consequently, SSFL emerges as a suitable approach to address speech-related challenges, including speech emotion recognition, speech recognition [79], and speaker identification/verification.
- **Unsupervised Methods in FL:** As direct labeling and pseudo-labeling methods may introduce undesired labeling noise to the data, FLUD methods that abstain from pseudo-labeling appear more promising in harnessing unlabeled data effectively. Consequently, there is a need for the development of innovative auto-encoder or self-supervised learning-based approaches that leverage unlabeled data without the assignment of pseudo-labels [28]. In particular, the

TABLE 5. Summary of quantitative results reported in selected FLUD studies on the CIFAR-10 dataset under different labeled data ratios. The acronmys used are as follows. PL: Pseudo-Labeling, CR: Consistency Regularization, RL: Reinforcement Learning, VR: Versatile Regularization, AL: Active Learning, CL: Contrastive Learning, TL: Transfer Learning.

Method	Labeled Data	L/U Clients	Labeled Data Usage	Architecture	Techniques	Accuracy (%)	Gain (%)[+]
CBAFed [24]	10%	10 (1L + 9U)	Training	ResNet18	PL	67.08	10.01
FedAC [72]	10%	30 (Partially labeled)	Training	AlexNet	PL + CR + RL	63.66	2.39
FedCA [26]	10%	5 (0L + 5U)	Fine-tuning (semi-sup.)	ResNet-50 + MLP	CL + TL	94.16	0.1
FedLABEL [77]	20%	100 (Partially labeled)	Training	ResNet34	PL	46.05	14.06
FedRVR [88]	10%	10 (1L + 9U)	Training	Simple CNN	VR	58.50	1.33
FedU [89]	10%	5 (0L + 5U)	Fine-tuning (semi-sup.)	ResNet-50 + MLP	CL	80.08	5.78
LoGo [76]	20%	10 (Partially labeled)	Training	Simple CNN	AL	66.50	0.18
TGAFed [74]	∼9%	100 (Partially labeled)	Training	ResNet-9	PL + CR	54.35	2.67

[+] The gain for each method was calculated with respect to the best-performing baseline reported under the same experimental conditions within its respective study. Therefore, direct comparisons of these gains across different methods may not be entirely objective.

exploration of self-supervised methods that incorporate pretext tasks on unlabeled data within a federated setting has been relatively limited in the literature, which presents a promising direction for future research. In particular for image data, the design of novel pretext tasks warrants investigation. In scenarios where active learning, involving human-in-the-loop interactions, is feasible [165], [166], a potential research direction involves minimizing human labeling effort by judiciously selecting unlabeled data sent to human annotators at the clients. Moreover, the server's inherent limitation in directly observing data poses challenges in identifying data samples that would offer maximal training value through labeling, thereby opening up another avenue for further research exploration in the context of active learning.

- **Data Scarcity:** The challenge of statistical heterogeneity and data scarcity among certain clients or classes is a significant issue for FLUD methods. To address data scarcity, generative methods have proven effective [65], [117]. By generating synthetic unlabeled data, it becomes possible to alleviate data scarcity to a certain extent. These synthetic data can then be combined with labeled data in later stages, contributing to better classification performance. Additionally, synthetic data generation offers a potential solution for overcoming the challenges associated with non-IID data distribution by promoting greater balance among classes at the client level.
- **Anomaly Detection:** Another promising direction for research could be anomaly detection, which can be used to detect abnormal (or rare) data in clients based on the application area, which may also lead to robustness in the client's anomaly or detection of intrusion/malicious clients [167]. Most studies on FL-based anomaly detection assume that local data is fully labeled with ground truths [168]. Some studies (e.g. [169]) do not require labeling but assume that the data in the training phase are benign and overlook the fact that anomalies may also take place during the training phase. Investigating semi-supervised FL approaches for anomaly detection based on the location of the labeled data requires further exploration.
- **Personalized Models:** Since the distribution of data between clients is inherently different in some applications, client-specific (personalized) models are also trained along with a global model. For example, in federated face recognition, each client can contain images of different identities and personalized models have been shown to improve the performance on local data [170], which can alleviate the mismatch between the data distributions on the server and in the clients. Although model personalization approaches have been proposed for fully supervised FL [171], [172], [173], personalized models have not been fully investigated for FLUD settings in terms of their usefulness for domain adaptation. This approach might also enable an easier generalization to unseen clients.
- **Multimodal (Multiview) methods:** In some real-world applications data at the clients may consist of multiple modalities and may have unlabeled data in some or all of these modalities, which is a topic that has not been sufficiently explored in the literature. Novel methods need to be developed to utilize multiple modalities for pseudo-labeling or self-supervised learning scenarios in FL. The distribution of modalities and missing modalities are factors that should be taken into account for multimodal FLUD methods [13]. Some examples of multimodal FL applications are emotion recognition from audio-visual and other data; human activity recognition using multiple sensors; and healthcare applications that use images, physiological measurements, and text data.
- **Curriculum Learning:** Curriculum learning is based on the idea of training a machine learning model from easier to harder data, which resembles the way humans learn starting from easier concepts and then moving on to harder concepts [174], [175]. Curriculum learning can be applied not only at the data level, but also at the model level, which is based on starting with an easier machine learning model and then proceeding with model complex models. Model-based curriculum

learning may be suitable for FL when clients are heterogeneous and differ in computing capacities [52]. Data-level and model-level curriculum learning has not been sufficiently explored in FL, especially for semi-supervised and self-supervised settings. Moreover, a client-based curriculum could also be developed in FL during the aggregation stage on the server [21].

- **Asynchronous FLUD:** In synchronous FL, the server updates the global model only after receiving model updates from all participating clients. Hence, the server waits for the latest client to send its model every time. This scenario is referred to as the straggler effect and can lead to a significant increase in delay, particularly in scenarios where clients exhibit high heterogeneity in terms of bandwidth, memory, processing speed, and data volume. To address such situations, asynchronous FL emerges as a solution, allowing the server to continuously update the global model as new local updates arrive from the clients. Additionally, to reduce the communication and computing costs associated with frequent global model updates, a semi-asynchronous setting can be employed. In this setup, model aggregation occurs after receiving parameters from a subset of clients.

  Although asynchronous FL has been extensively explored in supervised settings [176], there is a limited body of research investigating asynchronous FL in the absence of labeled data. These studies mainly focus on Case 2, where clients possess unlabeled data and only the server has labeled data [177], [178]. Further exploration is required, particularly in the case where some clients have labeled data, to advance our understanding of asynchronous and semi-asynchronous FL methods in such scenarios.
- **Incentive Mechanisms for FLUD:** The effectiveness of FL depends on increased client participation with valuable data. However, in practical situations, clients are hesitant to participate unless they receive compensation. To address this issue, various incentive mechanisms have been proposed for supervised FL, driven by factors such as the contribution of client data, reputation, or allocation of resources [179]. In a semi-supervised context, the amount of labeled data varies among clients, and in some instances, labeled data is concentrated in specific clients, making their participation crucial for model performance. Therefore, a promising avenue for future research involves designing incentive mechanisms that offer greater rewards to clients with more labeled data, while ensuring the authenticity and utility of these labels.
- **Designing Benchmark Tests for FLUD:** The papers in the literature report experimental results compared to a few relevant previous work. However, it is difficult to make a comprehensive comparison based on the reported experimental results, since their experimental protocols may be different. First, these methods are typically tested on different datasets using different settings. For instance, pseudo-labeling methods are commonly used on general image datasets (e.g. CIFAR-10, MNIST), while methods that do not use pseudo-labeling are often tested on datasets specific to particular applications. Furthermore, even in studies using the same dataset, there may be variations in data distribution and the amount of labeled data, making direct comparisons between these methods challenging. Therefore, benchmark test scenarios should be designed to allow for a more fair and consistent comparison of the FLUD methods in the literature.
- **Federated Self-Supervised Learning under Extreme Heterogeneity:** One promising avenue for future research is the development of federated self-supervised learning techniques that are robust to extreme heterogeneity. In practical FL settings, clients vary in terms of data distribution (e.g., class imbalance, domain shifts), computational power, and bandwidth. Traditional self-supervised methods assume IID data and uniform model capacity, which is rarely realistic. More research is needed to design adaptive pretext tasks that are personalized to the client's data, enable partial participation, and prevent representation collapse. Methods that incorporate client-specific embedding spaces or hierarchical federated architectures may offer solutions to these challenges.
- **Unsupervised and Utility-Aware Client Selection:** Client selection remains a largely underexplored topic in FLUD. Since the server lacks access to the raw data or even labels, it cannot directly evaluate the utility of each client's contribution. This makes random client sampling inefficient in settings where only a subset of clients have useful or high-quality data. Future work can focus on unsupervised utility estimation techniques, such as similarity clustering based on model divergence, update sparsity, or confidence scores derived from self-supervised embeddings. Reinforcement learning could also be employed to learn optimal selection policies based on reward signals aggregated over time.
- **Privacy Risks and Differential Privacy Guarantees:** Although FLUD avoids sharing raw data, privacy risks still persist through gradient leakage and indirect exposure of sensitive patterns in embeddings or pseudo-labels. Differential privacy (DP) has been widely used in supervised FL, but its integration in FLUD remains underdeveloped. New approaches are needed to ensure formal privacy guarantees, especially for semi-supervised and contrastive learning scenarios. For example, bounding the influence of pseudo-labeled instances on gradient updates or applying DP mechanisms to self-supervised representations may help mitigate privacy leakage without severely degrading model performance.

VII. CONCLUSION

In this paper, we conducted a comprehensive survey of recent FL methods that leverage unlabeled data (FLUD) during the model training process. The utilization of unlabeled data is achieved through various techniques, including semi-supervised learning, unsupervised learning, or self-supervised learning. To improve clarity, we introduced a novel taxonomy that categorizes these methods based on how they integrate unlabeled data into the training process. Additionally, we have presented a representative table (non-exhaustive) of FLUD methods along with the datasets commonly employed in the literature, categorized by their application areas. Our survey also highlights the challenges encountered in this emerging field and outlines potential avenues for future research directions.

ACKNOWLEDGMENT

The authors acknowledge the use of AI-assisted tools based on LLMs for grammar and writing refinement in certain parts of this manuscript. The sentences rephrased by the AI were reviewed and edited by the authors to ensure their accuracy and relevance.

REFERENCES

[1] S. Abdulrahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5476–5497, Apr. 2021.

[2] X. Liang and N. Ye, "Bart custers et al. (2019): EU personal data protection in policy and practice," *Int. J. Legal Discourse*, vol. 5, no. 1, pp. 119–125, May 2020.

[3] A. Kaal, J. Klosek, and B. Waleski, "U.S. consumer privacy bill of rights: Principles and impact," *Comput. Law Rev. Int.*, vol. 13, no. 3, pp. 65–72, Jan. 2012.

[4] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.

[5] B. K. Mathew, J. C. Ng, and J. L. Zerbe, "Using proxies to enable on-device machine learning," Patent 11 210 583, Dec. 28, 2021.

[6] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowl. Based Syst.*, vol. 216, Mar. 2021, Art. no. 106775.

[7] J. Zhou, S. Zhang, Q. Lu, W. Dai, M. Chen, X. Liu, S. Pirttikangas, Y. Shi, W. Zhang, and E. Herrera-Viedma, "A survey on federated learning and its applications for accelerating industrial Internet of Things," 2021, *arXiv:2104.10501*.

[8] Z. Iqbal and H. Y. Chan, "Concepts, key challenges and open problems of federated learning," *Int. J. Eng.*, vol. 34, no. 7, pp. 1667–1683, 2021.

[9] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, 2021, doi: 10.1561/2200000083.

[10] D. H. Mahlool and M. H. Abed, "A comprehensive survey on federated learning: Concept and applications," in *Proc. Mobile Comput. Sustain. Informat.*, 2022, pp. 539–553.

[11] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3347–3366, Apr. 2023.

[12] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, and W. Zhang, "A survey on federated learning: Challenges and applications," *Int. J. Mach. Learn. Cybern.*, vol. 14, no. 2, pp. 513–535, Feb. 2022.

[13] L. Che, J. Wang, Y. Zhou, and F. Ma, "Multimodal federated learning: A survey," *Sensors*, vol. 23, no. 15, p. 6986, Aug. 2023.

[14] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[15] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1759–1799, 3rd Quart., 2021.

[16] B. Liu, N. Lv, Y. Guo, and Y. Li, "Recent advances on federated learning: A systematic survey," *Neurocomputing*, vol. 597, Sep. 2024, Art. no. 128019.

[17] Y. Jin, X. Wei, Y. Liu, and Q. Yang, "Towards utilizing unlabeled data in federated learning: A survey and prospective," 2020, *arXiv:2006.06882*.

[18] S. Ji, Y. Tan, T. Saravirta, Z. Yang, Y. Liu, L. Vasankari, S. Pan, G. Long, and A. Walid, "Emerging trends in federated learning: From model fusion to federated x learning," *Int. J. Mach. Learn. Cybern.*, vol. 15, no. 9, pp. 3769–3790, Sep. 2024.

[19] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, "Federated semi-supervised learning with inter-client consistency & disjoint learning," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–11. [Online]. Available: https://openreview.net/forum?id=ce6CFXBh30h

[20] Z. Long, L. Che, Y. Wang, M. Ye, J. Luo, J. Wu, H. Xiao, and F. Ma, "FedSiam: Towards adaptive federated semi-supervised learning," 2020, *arXiv:2012.03292*.

[21] Y. Liu, X. Yuan, R. Zhao, Y. Zheng, and Y. Zheng, "RC-SSFL: Towards robust and communication-efficient semi-supervised federated learning system," 2020, *arXiv:2012.04432*.

[22] L. Che, Z. Long, J. Wang, Y. Wang, H. Xiao, and F. Ma, "FedTriNet: A pseudo labeling method with three players for federated semi-supervised learning," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 715–724.

[23] Y. Zhu, Y. Liu, J. J. Q. Yu, and X. Yuan, "Semi-supervised federated learning for travel mode identification from GPS trajectories," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2380–2391, Mar. 2022.

[24] M. Li, Q. Li, and Y. Wang, "Class balanced adaptive pseudo labeling for federated semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 16292–16301.

[25] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, Dec. 2020.

[26] F. Zhang, K. Kuang, L. Chen, Z. You, T. Shen, J. Xiao, Y. Zhang, C. Wu, F. Wu, Y. Zhuang, and X. Li, "Federated unsupervised representation learning," *Frontiers Inf. Technol. Electron. Eng.*, vol. 24, no. 8, pp. 1181–1193, 2023.

[27] N. Dong and I. Voiculescu, "Federated contrastive learning for decentralized unlabeled medical images," in *Proc. 24th Int. Conf. Med. Image Comput. Comput. Assist. Intervention–MICCAI*, 2021, pp. 378–387.

[28] R. Yan, L. Qu, Q. Wei, S.-C. Huang, L. Shen, D. L. Rubin, L. Xing, and Y. Zhou, "Label-efficient self-supervised federated learning for tackling data heterogeneity in medical imaging," *IEEE Trans. Med. Imag.*, vol. 42, no. 7, pp. 1932–1943, Jul. 2023.

[29] Y. Zhao, H. Liu, H. Li, P. Barnaghi, and H. Haddadi, "Semi-supervised federated learning for activity recognition," 2020, *arXiv:2011.00851*.

[30] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, and M. S. Hossain, "Mobility-aware proactive edge caching for connected vehicles using federated learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5341–5351, Aug. 2021.

[31] Z. Yu, J. Hu, G. Min, H. Xu, and J. Mills, "Proactive content caching for Internet-of-Vehicles based on peer-to-peer federated learning," in *Proc. IEEE 26th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2020, pp. 601–608.

[32] Y. Jin, Y. Liu, K. Chen, and Q. Yang, "Federated learning without full labels: A survey," *Bull. IEEE Comput. Soc. Tech. Committee Data Eng.*, vol. 46, no. 1, pp. 27–51, 2023.

[33] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2016, pp. 1273–1282.

[34] X. Yao, C. Huang, and L. Sun, "Two-stream federated learning: Reduce the communication costs," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2018, pp. 1–4.

[35] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[36] J. Shu, W. Zhang, Y. Zhou, Z. Cheng, and L. T. Yang, "FLAS: Computation and communication efficient federated learning via adaptive sampling," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 4, pp. 2003–2014, Jul. 2022.

[37] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.

[38] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," 2018, *arXiv:1812.07210*.

[39] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.I.d. data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[40] L. Wang, W. Wang, and B. Li, "CMFL: Mitigating communication overhead for federated learning," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 954–964.

[41] C. He, M. Annavaram, and S. Avestimehr, "Group knowledge transfer: Federated learning of large CNNs at the edge," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 14068–14080.

[42] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, *arXiv:1806.00582*.

[43] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, and R. Yonetani, "Hybrid-FL for wireless networks: Cooperative learning mechanism using non-IID data," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–7.

[44] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 4427–4437.

[45] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, 2018, pp. 429–450.

[46] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–14.

[47] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5132–5143.

[48] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–8.

[49] M. Sprague, A. Jalalirad, M. Scavuzzo, C. Capota, M. Neun, L. Do, and M. Kopp, "Asynchronous federated learning for geospatial applications," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2019, pp. 21–28.

[50] C. Xie, O. Koyejo, and I. Gupta, "Asynchronous federated optimization," in *Proc. 12th Annu. Workshop Optim. Mach. Learn.*, 2019, pp. 1–6. [Online]. Available: https://opt-ml.org/oldopt/papers/2020/paper28.pdf

[51] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[52] E. Diao, J. Ding, and V. Tarokh, "HeteroFL: Computation and communication efficient federated learning for heterogeneous clients," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–12. [Online]. Available: https://openreview.net/forum?id=TNkPBBYFkXg

[53] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," in *Proc. Mach. Learn. Syst.*, 2019, pp. 374–388.

[54] C. Dwork, "Differential privacy," in *Proc. Int. Colloq. Automata, Lang. Program.*, 2006, pp. 1–12.

[55] R. Cramer, I. B. Damgård, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*. Cambridge Univ. Press, 2015.

[56] X. Yi, R. Paulet, and E. Bertino, "Homomorphic encryption," in *Homomorphic Encryption and Applications*. Cham, Switzerland: Springer, 2014, pp. 27–46.

[57] M. Noble, A. Bellet, and A. Dieuleveut, "Differentially private federated learning on heterogeneous data," in *Proc. 25th Int. Conf. Artif. Intell. Statist.*, 2021, pp. 10110–10145.

[58] T. Zhu and P. S. Yu, "Applying differential privacy mechanism in artificial intelligence," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 1601–1609.

[59] M. Hao, H. Li, G. Xu, S. Liu, and H. Yang, "Towards efficient and privacy-preserving federated deep learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[60] W.-H. Mou, C. Fu, L. Yan, and C. Hu, "A verifiable federated learning scheme based on secure multi-party computation," in *Proc. Wireless Algorithms, Syst., Appl.*, 2021, pp. 198–209.

[61] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor, and C. S. Hong, "A crowdsourcing framework for on-device federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3241–3256, May 2020.

[62] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, T. N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7252–7261.

[63] Y. Wu, D. Zeng, Z. Wang, Y. Sheng, L. Yang, A. J. James, Y. Shi, and J. Hu, "Federated contrastive learning for dermatological disease diagnosis via on-device learning," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 2021, pp. 1–7.

[64] L. Zhang, B. Shen, A. Barnawi, S. Xi, N. Kumar, and Y. Wu, "FedDPGAN: Federated differentially private generative adversarial networks framework for the detection of COVID-19 pneumonia," *Inf. Syst. Frontiers*, vol. 23, no. 6, pp. 1403–1415, Dec. 2021.

[65] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, and A. Y. Zomaya, "Federated learning for COVID-19 detection with generative adversarial networks in edge cloud computing," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 10257–10271, Jun. 2022.

[66] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[67] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[68] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "TinyBERT: Distilling BERT for natural language understanding," in *Proc. Findings Assoc. Comput. Linguistics, EMNLP*, Nov. 2020, pp. 4163–4174.

[69] C. Bettini, G. Civitarese, and R. Presotto, "Personalized semi-supervised federated learning for human activity recognition," 2021, *arXiv:2104.08094*.

[70] H. Yu, Z. Chen, X. Zhang, X. Chen, F. Zhuang, H. Xiong, and X. Cheng, "FedHAR: Semi-supervised online learning for personalized federated human activity recognition," *IEEE Trans. Mobile Comput.*, vol. 22, no. 6, pp. 3318–3332, Jun. 2023.

[71] H. Lin, J. Lou, L. Xiong, and C. Shahabi, "SemiFed: Semi-supervised federated learning with consistency and pseudo-labeling," 2021, *arXiv:2108.09412*.

[72] Z. Jiang, Y. Xu, H. Xu, Z. Wang, and C. Qiao, "Clients help clients: Alternating collaboration for semi-supervised federated learning," in *Proc. IEEE 40th Int. Conf. Data Eng. (ICDE)*, May 2024, pp. 1847–1860.

[73] A. Wijesinghe, S. Zhang, S. Qi, and Z. Ding, "UFed-GAN: Secure federated learning over wireless sensor networks with unlabeled data," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2024, pp. 1048–1053.

[74] S. Zhu, Y. Wang, and G. Sun, "Federated semi-supervised learning based on truncated Gaussian aggregation," *J. Supercomput.*, vol. 81, no. 1, p. 269, Jan. 2025, doi: 10.1007/s11227-024-06798-z.

[75] T. Bdair, N. Navab, and S. Albarqouni, "Semi-supervised federated peer learning for skin lesion classification," *Mach. Learn. for Biomed. Imag.*, vol. 1, no. April 2022, pp. 1–37, Apr. 2022.

[76] S. Kim, S. Bae, H. Song, and S.-Y. Yun, "Re-thinking federated active learning based on inter-class diversity," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 3944–3953.

[77] Y. J. Cho, G. Joshi, and D. Dimitriadis, "Local or global: Selective knowledge assimilation for federated learning with limited labels," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 17041–17050.

[78] A. Albaseer, B. S. Ciftler, M. Abdallah, and A. Al-Fuqaha, "Exploiting unlabeled data in smart cities using federated edge learning," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2020, pp. 1666–1671.

[79] V. Tsouvalas, A. Saeed, and T. Ozcelebi, "Federated self-training for semi-supervised audio recognition," *ACM Trans. Embedded Comput. Syst.*, vol. 21, no. 6, pp. 1–26, Nov. 2022.

[80] T. Feng and S. Narayanan, "Semi-FedSER: Semi-supervised learning for speech emotion recognition on federated learning using multiview pseudo-labeling," in *Proc. INTERSPEECH*, Sep. 2022, pp. 5050–5054. [Online]. Available: https://www.isca-archive.org/interspeech2022/feng22interspeech.pdf

[81] Z. Long, J. Wang, Y. Wang, H. Xiao, and F. Ma, "FedCon: A contrastive framework for federated semi-supervised learning," 2021, *arXiv:2109.04533*.

[82] Z. Zhang, Y. Yang, Z. Yao, Y. Yan, J. E. Gonzalez, K. Ramchandran, and M. W. Mahoney, "Improving semi-supervised federated learning by reducing the gradient diversity of models," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 1214–1225.

[83] Z. Zhang, S. Ma, J. Nie, Y. Wu, Q. Yan, X. Xu, and D. Niyato, "Semi-supervised federated learning with non-IID data: Algorithm and system design," in *Proc. IEEE 23rd Int. Conf. High Perform. Comput. Commun., 7th Int. Conf. Data Sci. Syst., 19th Int. Conf. Smart City, 7th Int. Conf. Dependability Sensor, Cloud Big Data Syst. Appl. (HPCC/DSS/SmartCity/DependSys)*, Dec. 2021, pp. 157–164.

[84] Y. Zhang, Y. Gong, and Y. Guo, "Semi-supervised federated learning for assessing building damage from satellite imagery," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2024, pp. 3821–3826.

[85] Q. Liu, H. Yang, Q. Dou, and P. Heng, "Federated semi-supervised medical image classification via inter-client relation matching," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2021, pp. 325–335.

[86] X. Liang, Y. Lin, H. Fu, L. Zhu, and X. Li, "RSCFed: Random sampling consensus federated semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10144–10153.

[87] Y. Liu, H. Wu, and J. Qin, "FedCD: Federated semi-supervised learning with class awareness balance via dual teachers," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, no. 4, pp. 3837–3845.

[88] Q. Yang, Z. Chen, Z. Peng, and Y. Yuan, "Relation-guided versatile regularization for federated semi-supervised learning," *Int. J. Comput. Vis.*, vol. 133, no. 6, pp. 3312–3326, Jun. 2025, doi: 10.1007/s11263-024-02330-1.

[89] W. Zhuang, X. Gan, Y. Wen, S. Zhang, and S. Yi, "Collaborative unsupervised visual representation learning from decentralized data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4892–4901.

[90] D. Yang, Z. Xu, W. Li, A. Myronenko, H. R. Roth, S. Harmon, S. Xu, B. Turkbey, E. Turkbey, X. Wang, W. Zhu, G. Carrafiello, F. Patella, M. Cariati, H. Obinata, H. Mori, K. Tamura, P. An, B. J. Wood, and D. Xu, "Federated semi-supervised learning for COVID region segmentation in chest CT using multi-national data from China, Italy, Japan," *Med. Image Anal.*, vol. 70, May 2021, Art. no. 101992.

[91] E. Diao, J. Ding, and V. Tarokh, "SemiFL: Semi-supervised federated learning for unlabeled clients with alternate training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 17871–17884.

[92] S. Lee, T.-L. V. Le, J. Shin, and S.-J. Lee, "$(FL)^2$: Overcoming few labels in federated semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 43693–43714.

[93] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (Chapelle, O. Et al., Eds.; 2006) [book reviews]," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, p. 542, Mar. 2009.

[94] H. B. Barlow, "Unsupervised learning," *Neural Comput.*, vol. 1, no. 3, pp. 295–311, Sep. 1989.

[95] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2021.

[96] X. Yang, Z. Song, I. King, and Z. Xu, "A survey on deep semi-supervised learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 8934–8954, Sep. 2023.

[97] W. Li, C. Zhang, and Y. Tanaka, "Pseudo label-driven federated learning-based decentralized indoor localization via mobile crowdsourcing," *IEEE Sensors J.*, vol. 20, no. 19, pp. 11556–11565, Oct. 2020.

[98] B. Settles, "Active learning literature survey," Dept. Comput. Sci., University of Wisconsin-Madison, Madison, WI, USA, Tech. Rep. TR1648, 2009.

[99] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1136800

[100] S. Agrawal and N. Goyal, "Analysis of Thompson sampling for the multi-armed bandit problem," in *Proc. 25th Annu. Conf. Learn. Theory (COLT)*, Jun. 2011, pp. 39.1–39.26.

[101] M. Stikic, D. Larlus, and B. Schiele, "Multi-graph based semi-supervised learning for activity recognition," in *Proc. Int. Symp. Wearable Comput.*, Sep. 2009, pp. 85–92.

[102] D. Caldarola, M. Mancini, F. Galasso, M. Ciccone, E. Rodolà, and B. Caputo, "Cluster-driven graph federated learning over multiple domains," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 2743–2752.

[103] P. Bachman, O. Alsharif, and D. Precup, "Learning with pseudo-ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 3365–3373.

[104] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–11.

[105] M. S. M. Sajjadi, M. Javanmardi, and T. Taşdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1171–1179.

[106] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–8.

[107] D. Kim, K. Lee, Y. Lee, and H. Woo, "ACMFed: Fair semi-supervised federated learning with additional compromise model," *IEEE Access*, vol. 13, pp. 47734–47747, 2025.

[108] I. Misra and L. Van Der Maaten, "Self-supervised learning of pretext-invariant representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6706–6716.

[109] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9726–9735.

[110] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[111] P. Baldi, "Autoencoders, unsupervised learning and deep architectures," in *Proc. ICML Workshop Unsupervised Transf. Learn.*, 2011, pp. 37–50.

[112] D. Cha, M. Sung, and Y.-R. Park, "Implementing vertical federated learning using autoencoders: Practical application, generalizability, and utility study," *JMIR Med. Informat.*, vol. 9, no. 6, Jun. 2021, Art. no. e26598, doi: 10.2196/26598.

[113] V. T. Nguyen and R. Beuran, "FedMSE: Semi-supervised federated learning approach for IoT network intrusion detection," *Comput. Secur.*, vol. 151, Apr. 2025, Art. no. 104337.

[114] V. L. Cao, M. Nicolau, and J. McDermott, "Learning neural representations for network anomaly detection," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 3074–3087, Aug. 2019.

[115] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009.

[116] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[117] C. Xu, R. Xia, Y. Xiao, Y. Li, G. Shi, and K.-C. Chen, "Federated traffic synthesizing and classification using generative adversarial networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2021, pp. 1–6.

[118] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 478–487.

[119] Z. Li, J. Shao, Y. Mao, J. H. Wang, and J. Zhang, "Federated learning with GAN-based data synthesis for non-IID clients," in *Proc. Int. Workshop Trustworthy Federated Learn.*, 2022, pp. 17–32.

[120] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019.

[121] S. Chen and J. Shen, "Exploitation maximization of unlabeled data for federated semi-supervised learning," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 9, no. 2, pp. 2039–2044, Apr. 2025.

[122] K. Sohn, D. Berthelot, C. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 596–608.

[123] X. Mu, Y. Shen, K. Cheng, X. Geng, J. Fu, T. Zhang, and Z. Zhang, "FedProc: Prototypical contrastive federated learning on non-IID data," *Future Gener. Comput. Syst.*, vol. 143, pp. 93–104, Jun. 2023.

[124] C. Park, C. Min, S. Bhattacharya, and F. Kawsar, "Augmenting conversational agents with ambient acoustic contexts," in *Proc. 22nd Int. Conf. Hum.-Comput. Interact. with Mobile Devices Services*, Oct. 2020, pp. 1–9.

[125] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018, *arXiv:1804.03209*.
[126] voxforge.org. (2006). *VoxForge Dataset*. Accessed: Mar. 28, 2025. [Online]. Available: https://www.voxforge.org/home
[127] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Lang. Resour. Eval.*, vol. 42, no. 4, pp. 335–359, Dec. 2008.
[128] C. Busso, S. Parthasarathy, A. Burmania, M. AbdelWahab, N. Sadoughi, and E. M. Provost, "MSP-IMPROV: An acted corpus of dyadic interactions to study emotion perception," *IEEE Trans. Affect. Comput.*, vol. 8, no. 1, pp. 67–80, Jan. 2017.
[129] R. Gupta, B. Goodman, N. Patel, R. Hosfelt, S. Sajeev, E. Heim, J. Doshi, K. Lucas, H. Choset, and M. E. Gaston, "Creating xBD: A dataset for assessing building damage from satellite imagery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. workshops*, Jun. 2019, pp. 10–17.
[130] A. Krizhevsky, "Learning multiple layers of features from tiny images," Master's thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.
[131] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey, "CINIC-10 is not ImageNet or CIFAR-10," 2018, *arXiv:1810.03505*.
[132] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "EMNIST: Extending MNIST to handwritten letters," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2921–2926.
[133] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[134] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar, "LEAF: A benchmark for federated settings," in *Proc. 33rd Conf. Neural Inf. Process. Syst. (NeurIPS) Workshop Federated Learn. Data Privacy Confidentiality*, 2018, pp. 1–5.
[135] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the Web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
[136] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–8.
[137] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2011, pp. 1–7.
[138] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: A multi-class classification competition," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 1453–1460.
[139] D. B. Calheiros and S. F. Da Silva. (2025). *Atlasderm Dermatology Diseases*. Accessed: Mar. 28, 2025. [Online]. Available: http://www.atlasdermatologico.com.br
[140] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "MedMNIST v2—A large-scale lightweight benchmark for 2D and 3D biomedical image classification," *Sci. Data*, vol. 10, no. 1, p. 41, Jan. 2023.
[141] P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, and A. Mohammadi, "COVID-CAPS: A capsule network-based framework for identification of COVID-19 cases from X-ray images," *Pattern Recognit. Lett.*, vol. 138, pp. 638–643, Oct. 2020.
[142] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. L. Ball, K. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. H. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng, "CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 590–597.
[143] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-ray8: hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3462–3471.
[144] M. E. H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, K. R. Islam, M. S. Khan, A. Iqbal, N. A. Emadi, M. B. I. Reaz, and M. T. Islam, "Can AI help in screening viral and COVID-19 pneumonia?" *IEEE Access*, vol. 8, pp. 132665–132676, 2020.
[145] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, "Automated detection of COVID-19 cases using deep neural networks with X-ray images," *Comput. Biol. Med.*, vol. 121, Jun. 2020, Art. no. 103792.
[146] D. N. Zealand. (2025). *Skin Disease Atlas*. Accessed: Mar. 28, 2025. [Online]. Available: http://www.dermnet.com
[147] J. Kawahara, S. Daneshvar, G. Argenziano, and G. Hamarneh, "Seven-point checklist and skin lesion classification using multitask multimodal neural nets," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 2, pp. 538–546, Mar. 2019.
[148] P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific Data*, vol. 5, no. 1, Aug. 2018, Art. no. 180161, doi: 10.1038/sdata.2018.161.
[149] challenge.isic archive.com. (2019). *ISIC-2019 Dataset*. Accessed: Mar. 28, 2025. [Online]. Available: https://challenge.isic-archive.com/landing/2019/
[150] challenge2020.isic archive.com. (2020). *ISIC-2020 Dataset*. Accessed: Mar. 28, 2025. [Online]. Available: https://challenge2020.isic-archive.com/
[151] A. G. C. Pacheco, G. R. Lima, A. S. Salomão, B. Krohling, I. P. Biral, G. G. De Angelo, F. C. R. Alves Jr., J. G. M. Esgario, A. C. Simora, P. B. C. Castro, F. B. Rodrigues, P. H. L. Frasson, R. A. Krohling, H. Knidel, M. C. S. Santos, R. B. do Espírito Santo, T. L. S. G. Macedo, T. R. P. Canuto, and L. F. S. De Barros, "PAD-UFES-20: A skin lesion dataset composed of patient data and clinical images collected from smartphones," *Data Brief*, vol. 32, Oct. 2020, Art. no. 106221.
[152] A. Stein, C. Wu, and C. Carr. (2019). *RSNA Intracranial Hemorrhage Detection*. Accessed: Mar. 28, 2025. [Online]. Available: https://www.kaggle.com/competitions/rsna-intracranial-hemorrhage-detection
[153] H. Q. Nguyen et al., "VinDr-CXR: An open dataset of chest X-rays with radiologist's annotations," *Sci. Data*, vol. 9, no. 1, p. 429, Jul. 2022.
[154] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features," in *Proc. 2nd Int. Conf. Inf. Syst. Secur. Privacy*, 2016, pp. 407–414.
[155] M. Bachlin, D. Roggen, G. Troster, M. Plotnik, N. Inbar, I. Meidan, T. Herman, M. Brozgol, E. Shaviv, N. Giladi, and J. M. Hausdorff, "Potentials of enhanced context awareness in wearable assistants for Parkinson's disease patients with the freezing of gait syndrome," in *Proc. Int. Symp. Wearable Comput.*, Sep. 2009, pp. 123–130.
[156] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis, "The MobiAct dataset: Recognition of activities of daily living using smartphones," in *Proc. Int. Conf. Inf. Commun. Technol. Ageing Well e-Health*, 2016, pp. 143–151.
[157] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. D. R. Millán, and D. Roggen, "The opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognit. Lett.*, vol. 34, no. 15, pp. 2033–2042, Nov. 2013.
[158] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput.*, Jun. 2012, pp. 108–109.
[159] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explor. Newslett.*, vol. 12, no. 2, pp. 74–82, Mar. 2011.
[160] G. Research. (2021). *MovieLens Dataset*. Accessed: Mar. 28, 2025. [Online]. Available: https://grouplens.org/datasets/movielens/
[161] T.-C. Chiu, Y.-Y. Shih, A.-C. Pang, C.-S. Wang, W. Weng, and C.-T. Chou, "Semisupervised distributed learning with non-IID data for AIoT service platform," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9266–9277, Oct. 2020.
[162] V. Tsouvalas, T. Ozcelebi, and N. Meratnia, "Privacy-preserving speech emotion recognition through semi-supervised federated learning," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops other Affiliated Events (PerCom Workshops)*, Mar. 2022, pp. 359–364.
[163] D. Aggarwal, J. Zhou, and A. K. Jain, "FedFace: Collaborative learning of face recognition model," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Aug. 2021, pp. 1–8.
[164] R. Shao, P. Perera, P. C. Yuen, and V. M. Patel, "Federated generalized face presentation attack detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 1–14, Jan. 2024.
[165] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM Comput. Surveys*, vol. 54, no. 9, pp. 1–40, 2021.
[166] B. Büyüktaş, Ç. E. Erdem, and T. Erdem, "More learning with less labeling for face recognition," *Digit. Signal Process.*, vol. 136, May 2023, Art. no. 103915. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200423000106

[167] R. Zhao, Y. Wang, Z. Xue, T. Ohtsuki, B. Adebisi, and G. Gui, "Semisupervised federated-learning-based intrusion detection method for Internet of Things," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8645–8657, May 2023.
[168] E. M. Campos, P. F. Saura, A. González-Vidal, J. L. Hernández-Ramos, J. B. Bernabé, G. Baldini, and A. Skarmeta, "Evaluating federated learning for intrusion detection in Internet of Things: Review and challenges," *Comput. Netw.*, vol. 203, Feb. 2022, Art. no. 108661.
[169] P. Tian, Z. Chen, W. Yu, and W. Liao, "Towards asynchronous federated learning based threat detection: A DC-adam approach," *Comput. Secur.*, vol. 108, Sep. 2021, Art. no. 102344.
[170] C. Liu, C.-Y. Wang, S. Chien, and S. Lai, "FedFR: Joint optimization federated framework for generic and personalized face recognition," in *Proc. 36th AAAI Conf. Artif. Intell. (AAAI-22)*, vol. 36, 2022, pp. 1656–1664.
[171] F.-E. Yang, C.-Y. Wang, and Y.-C.-F. Wang, "Efficient model personalization in federated learning via client-specific prompt generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 19159–19168.
[172] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, J. Cao, and H. Guan, "GPFL: Simultaneously learning global and personalized feature information for personalized federated learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 5018–5028.
[173] G. Sun, M. Mendieta, J. Luo, S. Wu, and C. Chen, "FedPerfix: Towards partial model personalization of vision transformers in federated learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 4965–4975.
[174] X. Wang, Y. Chen, and W. Zhu, "A survey on curriculum learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 4555–4576, Sep. 2022.
[175] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe, "Curriculum learning: A survey," *Int. J. Comput. Vis.*, vol. 130, no. 6, pp. 1526–1565, 2022.
[176] C. Xu, Y. Qu, Y. Xiang, and L. Gao, "Asynchronous federated learning on heterogeneous devices: A survey," *Comput. Sci. Rev.*, vol. 50, Nov. 2023, Art. no. 100595.
[177] N. Yang, D. Yuan, Y. Zhang, Y. Deng, and W. Bao, "Asynchronous semi-supervised federated learning with provable convergence in edge computing," *IEEE Netw.*, vol. 36, no. 5, pp. 136–143, Sep. 2022.
[178] L. Liu, W. Zhai, F. Wang, Y. Ding, W. Lu, and W. Meng, "Federated semi-supervised and semi-asynchronous learning for anomaly detection in IoT networks," 2023, *arXiv:2308.11981*.
[179] Y. Zhan, J. Zhang, Z. Hong, L. Wu, P. Li, and S. Guo, "A survey of incentive mechanism design for federated learning," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 2, pp. 1035–1044, Apr. 2022.

KÜBRA ULUDAG received the B.S. and M.S. degrees in computer engineering from Marmara University, Turkey, in 2014 and 2018, respectively, where she is pursuing the Ph.D. degree in computer engineering. She also completed a double major program in electrical and electronics engineering in 2015. She is currently a Research Assistant at Marmara University. Her research interests include user-centric networks, wireless networks, machine learning, and federated learning.



ÇIGDEM EROGLU ERDEM (Member, IEEE) received the B.S. and M.Sc. degrees (Hons.) in electrical and electronics engineering from Bilkent University, Ankara, Türkiye, in 1995 and 1997, respectively, and the Ph.D. degree in electrical and electronics engineering from Boğaziçi University, Istanbul, Türkiye, in 2002.

From September 2000 to June 2001, she was a Visiting Researcher with the Department of Electrical and Computer Engineering, University of Rochester, NY, USA. From 2003 to 2004, she was a Postdoctoral Fellow with the Faculty of Electrical Engineering, Delft University of Technology, The Netherlands, where she was also affiliated with the Video Processing Group, Philips Research Laboratories, Eindhoven. From 2002 to 2009, she was the Director of Research at Momentum Digital Media Technologies Inc., a technology SME located in Istanbul. From 2009 to 2016, she was a Faculty Member with the Department of Electrical and Electronics Engineering, Bahçeşehir University, Istanbul. From 2016 to 2024, she was a Full Professor with the Department of Computer Engineering, Marmara University, Istanbul. Since 2024, she has been a Full Professor with the Department of Electrical and Electronics Engineering, Özyeğin University, Istanbul. Her current research interests include digital image and video processing, computer vision, and pattern recognition with applications in biometrics and federated learning. She served as a referee for numerous technical journals. She also served as an independent expert and the vice chair during project evaluations for European Commission.



ÖMER KORÇAK (Member, IEEE) received the B.S. (Hons.), M.S., and Ph.D. degrees in computer engineering from Boğaziçi University, in 2002, 2004, and 2009, respectively. In 2011, he was a Visiting Researcher with Deutsche Telekom Laboratories, Berlin. He is currently an Associate Professor with Marmara University, Istanbul. His research interests include distributed learning, edge intelligence, smart IoT, game theory and incentive mechanism design for networked systems, and security and privacy-preserving mechanisms for intelligent systems.