Discriminative Feature Learning-Based Federated Lightweight Distillation Against Multiple Attacks

Haijiao Chen, *Graduate Student Member, IEEE*, Huan Zhao, *Member, IEEE*, Zixing Zhang, *Senior Member, IEEE*, and Keqin Li, *Fellow, IEEE*

***Abstract*—Thanks to the advantages of cloud and edge computing, federated learning (FL)–based speech emotion recognition (SER) tasks can be well-scaled to cloud–edge–terminal ecosystems. It aims to characterize emotions while protecting data privacy. However, catastrophic forgetting caused by data heterogeneity, potential system attacks, and possible privacy leakage and communication overhead from parameter sharing have constrained its breakthrough. Some schemes that attempt to tackle the FL bottleneck do not consider these issues comprehensively. We propose a federated distillation-based multiple defense approach (FedMud), which simultaneously considers how to balance system performance, privacy security, and communication overhead. First, it employs a server-side lightweight generator to learn global view knowledge and guides client-side updates through distillation, further mitigating catastrophic forgetting and improving system performance. In addition, we design a multipath integrated defense paradigm to counter potential system attacks, with a data perturbation technique based on gradient modification, a dynamically weighted selection method, and a privacy-enhanced strategy by capturing discriminative features. Moreover, to minimize parameter leakage, the parameter-decoupled hierarchical sharing mechanism is utilized, which also significantly reduces the communication overhead. The experimental results show that our approach is effective, with gender predictions down to chance levels while maintaining SER performance enhancements.**

***Index Terms*—Anti-attribute inference attacks, cloud computing, edge computing, federated learning (FL), knowledge distillation (KD), knowledge selection, privacy protection.**

I. Introduction

SPEECH emotion recognition (SER), a critical component of human–computer interaction applications, is responsible for recognizing the emotional states expressed by voices. It is extensively utilized in smart homes [1], medical diagnostics [2], advanced driver assistance systems [3], voice assistants, and others. The data explosion in multiple fields facilitates the development of Artificial Intelligence of Things (AIoT). Integrating multisource data, offering real-time analysis, and achieving assisted assistant decisions have emerged as a new paradigm for AIoT applications. As technology evolves, cloud computing has gained popularity thanks to its ability to effectively overcome the limitations of terminal devices in terms of computation, communication, storage, etc., but it suffers from system performance degradation due to the long-link connection and the limited bandwidth between the terminal and the cloud, e.g., service delay and network overload and even privacy leakage. Edge computing can make up for the shortcomings of cloud computing but has to consider the limited resources at the edge. Deepening the cloud–edge–terminal (C–E–T) collaboration to promote higher levels of AIoT development.

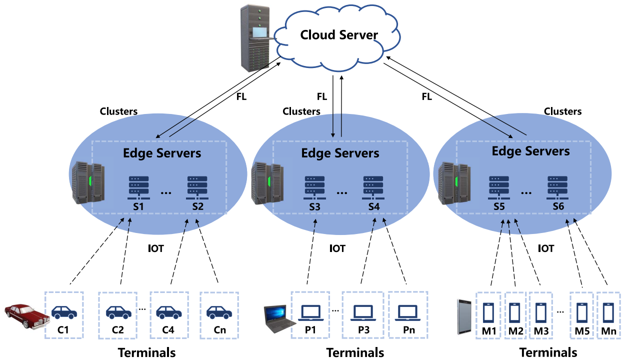

Fig. 1. FL-based cloud–edge–terminal framework.

As a distributed machine learning (ML) collaboration paradigm, federated learning (FL) [4] was first proposed to address challenges, such as data silos and data privacy. It allows participants (i.e., data owners) to collaboratively train shared models (gradients or parameters) without accessing local data, leveraging its advantages in privacy preservation and reduced transmission overhead [5]. The AIoT-based C–E–T collaboration framework, depicted in Fig. 1, includes terminal devices, edge server clusters, and cloud servers. To capitalize on the confluence of cloud computing and edge computing, we employ edge servers as data storage servers, which are usually deployed in the last kilometer between the

Manuscript received 24 October 2023; revised 14 December 2023 and 5 January 2024; accepted 22 January 2024. Date of publication 30 January 2024; date of current version 9 May 2024. This work was supported by the National Natural Science Foundation of China under Grant 62076092; in part by the National Key Research and Development Program of China under Grant 2020YFB1713400; and in part by the Special Project of Foshan Science and Technology Innovation Team, China, under Grant FS0AA-KJ919-4402-0069. *(Corresponding authors: Huan Zhao; Zixing Zhang.)*

Haijiao Chen, Huan Zhao, and Zixing Zhang are with the College of Information Science and Engineering, Hunan University, Changsha 410082, China (e-mail: chenhaijiao@hnu.edu.cn; hzhao@hnu.edu.cn; zixingzhang@hnu.edu.cn).

Keqin Li is with the Department of Computer Science, State University of New York at New Paltz, New Paltz, NY 12561 USA, and also with the College of Information Science and Engineering, Hunan University, Changsha 410082, China (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/JIOT.2024.3360094

user and the data source, and have advantages in computing power, storage volume, and network bandwidth compared with the terminal devices. Matched by device type and the nearest base station, it collects terminal device data in real time through IoT encryption. In addition, the cloud server is responsible for model aggregation and other large-scale computations in collaborative training. The practical AIoT systems integrating real-time data collection and multiparty collaborative training are constructed.

Data statistical heterogeneity [i.e., nonindependently identically distributed (non-iid)] induced by device variability is one of the primary challenges of C–E–T-based FL systems [6] and [7] demonstrated that non-iid greatly hurts FL performance and impairs system convergence. Some enhanced federated averaging (FedAvg)-based methods [8], [9] address heterogeneity by minimizing variations between local training parameters, but it remains inadequate when deep neural network architectures are used [10], [11]. The statistical variability of SER tasks has received less attention. Furthermore, the model in FL local updating is optimized for private data, which is prone to overfitting the present knowledge and forgetting what was learned from other clients during the collaborative updating phase, which we refer to as catastrophic forgetting [12]. Typically, the condition is resolved by fine-tuning such as meticulously adjusting hyperparameters [13], which takes time and does not fix the problem consistently. Knowledge distillation (KD) [14], [15] has emerged as a solution that utilizes the integrated knowledge of local models to enrich the global model more efficiently than parameter averaging and fine-tuning, however, utilizing additional proxy data sets makes it limited.

The majority of present work on FL-based SER focuses on system performance while ignoring potential system attacks and privacy concerns. It has been demonstrated that attackers can recover local training data from uploaded gradients [16], [17], [18]. More specifically, gradient updates can infer eigenvalues, and the virtual data recovered using eigenvalues is infinitely close to the real data, allowing for data recovery, which is a gradient inversion attack. FL is vulnerable to attribute inference attacks, which is also a gradient-based attack that a curious server analyzes the statistical characteristics (e.g., gender, age, or identity) of the client's data using aggregated shared model parameters. Feng et al. [19] simulated a white-box attack, in which the model architecture and hyper-parameters are almost transparent to the attackers and successfully inferred the gender attribute of the client. Therefore, the shared gradient protection becomes particularly important. When an attacker injects malicious content into training data, the mislabeled data can readily degrade model performance, and such data poisoning attacks are frequently undetected. Many more attacks exist in real-world settings, and a minor well-designed perturbation can lead a model to forecast a class inaccurately. A strong defensive model is required, and adversarial training has shown promise in increasing model robustness [20], [21].

Meanwhile, the traditional approach to enhancing privacy is cryptography, specifically differential privacy [22], which prevents leakage by modifying client parameters before they are uploaded to the server, but the effectiveness of the protection is greatly reduced when an attacker observes multiple model updates. Homomorphic encryption [23] and multiparty secure computing [24] are further options. Due to bandwidth and distributed logic restrictions, some encryption techniques are less appropriate for FL frameworks. Most early defense techniques concentrated on one specific attack or privacy leakage. For the complexities of C–E–T collaboration, we urgently want a security solution that fights multiple attacks and prevents privacy leakages.

To overcome these challenges, in this article, we propose a discriminative feature learning-based federated lightweight generation method for countering multiple attacks and preserving privacy in the C–E–T environment. On the one hand, we learn client-side knowledge using a lightweight generator without using any proxy data, which avoids catastrophic forgetting owing to model averaging. The extracted knowledge is then utilized to guide local updates, resulting in improved generalization performance on non-iid data distributions. On the other hand, some protection mechanisms are proposed. The local data perturbation (DP) achieves data enhancement while defending against gradient inversion attacks and adversarial attacks. With the addition of the privacy-enhanced (PE) module, we aim to focus on emotion-related features and filter irrelevant redundant features to prevent attribute inference attacks. The client selection can avoid malicious tampering of data or labels by actively removing unreliable information, preventing data poisoning attacks, and helping positive prediction. Furthermore, a parameter hierarchical sharing mechanism is used to satisfy the system privacy and communication constraints by sharing the prediction layer. Our main contributions are summarized as follows.

1) A federated lightweight distillation scheme based on discriminative feature learning is proposed, which comprehensively considers system performance, privacy security, and communication overhead, and it also provides excellent cross-domain scalability and generalization.
2) It learns reliable knowledge with the global view, and guides local updates through distillation, to achieve improved model generalization performance under non-iid distribution. Besides, theoretically analyzing the integration performance of cross-domain global distribution. A robust multipath integration defense mechanism is constructed to actively protect against gradient inversion attacks, adversarial attacks, data poisoning attacks, and attribute inference attacks. Furthermore, sharing the local model prediction layer helps reduce privacy leakage and decrease the communication overhead.
3) For the first practice in the SER task, we simulate the attribute inference attacks and defense under the cloud–edge–terminal settings, and the results demonstrate that our scheme effectively defends against attacks while maintaining superior SER performance.

The remainder of this article is organized as follows. Section II reviews the related work. Section III proposes a federated multiple defense approach, and provides the related theoretical analysis. Section IV describes the experimental results and evaluation. Conclusions and future works are drawn in Section V. Finally, Section Appendix adds scalability and generalizability analysis.

II. Related Work

In recent years, FL has gained popularity in more fields, which can extend the underlying model data availability, and enable computational sharing, and data localization during collaboration can protect privacy. Data non-iid constrains system performance in FL setting [25]. Furthermore, the FL system is vulnerable to both inferencing and adversarial attacks.

Several works have attempted to address the issue of non-iid to improve system performance. FedProx [8] optimizes based on FedAvg, which adds approximation terms to the local model to restrict the local updates to be closer to the global model. To account for local update drift, SCAFFOLD [9] applies variance reduction strategies. With the successful deployment of KD in FL tasks, new ways are developing. With the help of unlabeled data sets, FedDF [14] presents integrated distillation for model fusion, which trains the global model using locally averaged logits. FedDistill [26] creates a global KD by improving the user data logics gained via model forward propagation to decrease global drift. FedAux [27] finds a local model initialization that weights local model logits using differential privacy deterministic scoring. All of the preceding work is based on unlabeled proxy data sets, and it is unclear how closely the proxy data sets are linked to the training data sets to guarantee good KD. Furthermore, the various data distributions of the proxy data sets may influence the result of KD. Following this, data-free KD emerged, in which information is taken from a pretrained teacher model (pseudo-data) and migrated to another student model without proxy data. DeepImpression [28] models the output of the teacher model by fitting it to recover real data. The work [29] extracted metadata from the activation layer of the teacher model. Mao et al. [30] learned a conditional generator that creates samples by maximizing the teacher's predicted probability on the target label. Inspired by the same, FedGen [31] learns a generative model that integrates local model knowledge over the latent space for lightweighting.

Unlike previous work, we deliver performance gains by optimizing the knowledge structure to generate reliable knowledge from global views by first extracting relevant and reliable information from the clients, which further guides local updates. In this regard, acquiring reliable knowledge is contributed by our PE module and client selection module. Client-side knowledge augmentation benefits from extracted locally relevant knowledge, sampled globally reliable knowledge, and enhanced knowledge from DPs.

Regarding the related research on privacy preservation, a noisy representation for protecting prediction privacy [32] was developed as a gradient-based perturbation maximization technique that removes irrelevant features by introducing noise into the input. Feng et al. [33] then presented theoretical user-level differential privacy guarantees via privacy parameters. Gradually, other schemes for homomorphic encryption [34] and multiparty secure computation [35] were proposed. These above noise additions or encryption to ensure privacy have significant limitations in FL.

Furthermore, adversarial attacks seem to have become a research staple in recent years. Ren et al. [21] proposed an adversarial approach with embedded constraints to limit the similarity between the original samples and their opponent's samples when performing adversarial training. To effectively improve model robustness, Chang et al. [36] presented a two-stage technique of adversarial training and randomized testing. These single defenses, however, are incapable of dealing with complicated attacks, and generation-based adversarial network-related measures demand a large computing overhead. Currently, robust defense schemes against multiple attacks have not been developed in the FL field.

III. Federal Multiple Defense

Our goals are to:

1) build an AIoT C–E–T collaboration framework based on FL;
2) analyze multiple attacks caused by parameter sharing, incorrect samples, and gradient updating, further propose a set of targeted defense schemes, and deductively interpret the feasibility and effectiveness of the schemes;
3) balance the system performance degradation caused by data heterogeneity and the communication overhead caused by frequent interaction of parameters in SER tasks while ensuring system privacy.

Before introducing the approaches, various conceivable attack situations will be listed briefly.

As shown in Fig. 2, the general framework of a lightweight federated PE scheme under the SER task is given, which mainly consists of two parts: 1) the FL privacy-protected module with a C–E–T collaboration (the upper, i.e., SER/Shadow module) and 2) the Attack module (the lower, we concentrate on modeling gender attribute inference attack, all attacks also include data poisoning and gradient inversion).

A. Data Sets

The commonly used public data sets IEMOCAP and MSP-Improv were employed in the SER task. Due to the unbalanced distribution, we utilize the four most frequent emotions (happy, sad, angry, and neutral), and the four labels are included in both data sets, as detailed below.

The IEMOCAP database [37] is a multimodal sensor database collected by the University of California, ten actors (five males and five females) spontaneously interacted orally with selected emotional scripts, resulting in 10 039 utterances over 12 h. Detailed motion, audio, and video of the interactions were captured from the face, hand, and head-tagged sessions. In addition, the improvised scenes aim at real emotional interactions to stimulate specific types of emotions. Therefore, the scripts are divided into script-conditioned scenes and improvised-conditioned scenes according to whether the scripts are from a script or not. We chose only the latter, which samples 2943 emotion labels, including 2415 training sets and 528 test sets.

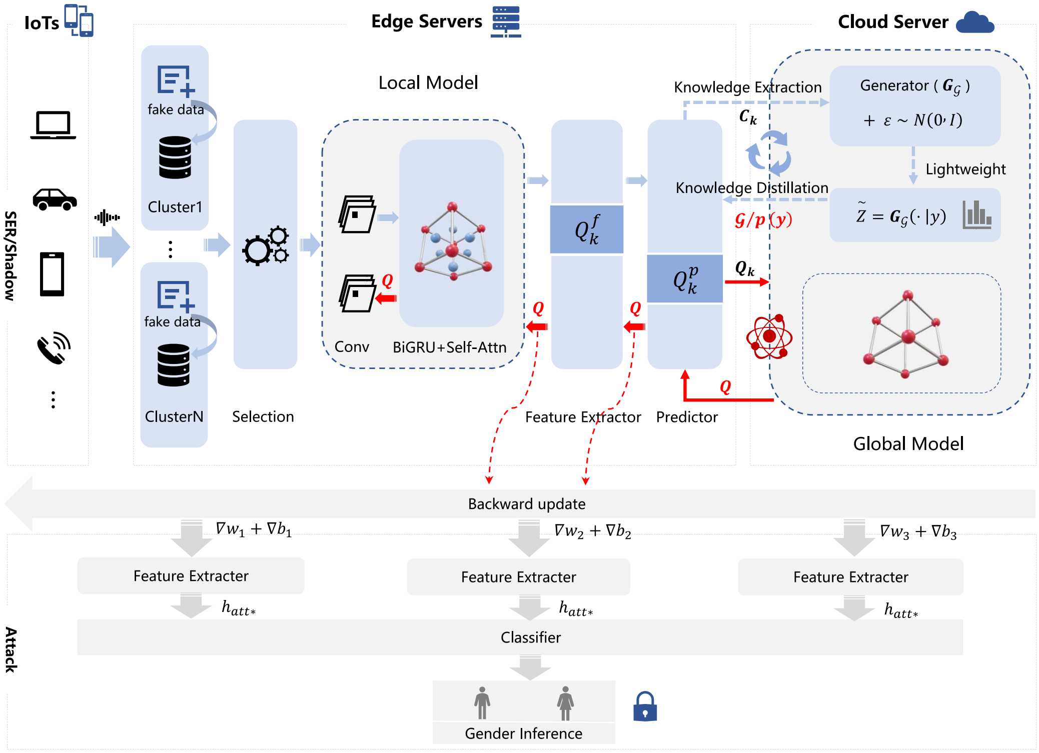

Fig. 2. PE scheme (FedMud) against multiple attacks based on the FL framework.

The MSP-Improv database [38] collected naturalistic interactional emotions from 12 (six male and six female) English major participants who engaged in six conversations totaling over 9 h. The corpus recorded audio and video data, including 4381 utterances in the improvisation condition, 2785 utterances in the naturalistic condition, 652 utterances in the target condition, and 620 utterances in the reading speech condition. We used only the improvisation condition data, where 4580 emotion labels were sampled with 3583 training sets and 997 test sets.

B. Attack Design Philosophy

In an FL setting, the attacker tries to predict the gender label from a client by the shared model updates of the main task SER and other public data. D_p is defined as a private data set, including the feature set X, the accompanying sentiment label Y, and the gender label z from multiple clients. Assume that the attacker does not have access to the private data set, but rather to a public data set D_h with a distribution similar to D_p. Following that, we will focus on attribute inference attacks and briefly discuss other attacks.

1) Attribute Inference Attack: Similar to [19], we design an attribute inference attack in three stages, privacy training, shadow training, and inference attack.

The private training aims to train collaboratively to produce shared model update g^t by the private data set D_p. Model-shared updates can indirectly expose privacy even if an attacker does not have access to D_p.

Shadow training mimics membership inference attacks, which was first proposed in [17]. Shadowing models M_s trained to mimic privacy training models M_p, aiming at recognizing emotions from speech features. An attacker typically employs D_h that is similar to the privacy training data sets in both format and distribution and ensures that they do not overlap. Attribute inference attacks are white-box attacks, so the architecture and related parameters are the same regarding shadow and privacy models.

Finally, we collect the shared model update g_k^t and labeled gender z_k generated from the kth client's shadow training as the attack data set D_a. D_a is then used to train attack model M_a to infer the gender z. The attacker can access the global model parameter θ^t as well as the model parameter θ_k^t of the kth client's update, but not the original gradient. Therefore, we derive a pseudo-gradient $g{*}_k^t$ similar to that in [18] as an attack input

$$g{*}_k^t = \frac{1}{T\eta}\left(\theta^t - \theta_k^t\right) \tag{1}$$

where η is the learning rate, and the kth client iterates T times. Our goal is to train attack models for parameterized ϑ to minimize cross-entropy loss

$$\min_{\vartheta} \mathcal{L}\left(M_a\left(g{*}_k^t, \vartheta\right), z_k\right). \tag{2}$$

2) Data Poisoning Attack: The attackers hope to harm the model or impair its performance by adding partially modified or malicious data into the training data set. The data poisoning may be divided into clean label poisoning and dirty label poisoning. Label flipping, as a typical dirty label poisoning attack, maliciously flips an original label $l_x \in l$ to another wrong one, where the set of training labels l, then $F_{\text{invert}} : l_x \rightarrow l_{x'}$, and we define the attack process by $l_{x'} =$

Algorithm 1 FedMud for Cloud–Edge–Terminals

```
Input: $k$ clients (FL clients) by private data $D_i$, task $\mathcal{T}_k$; local
    parameters $\left\{\theta_k = \left[\theta_k^e; \theta_k^p\right]\right\}_{k=1}^{K}$, $p(y)$ initialized, local label
    counter $C_k$; global parameters $\theta^p$, generator parameters $\mathcal{G}$,
    gaussian noise $\varepsilon \sim \mathcal{N}(0, I)$; learning rate $\alpha$, $\beta$, local step
    $T$, batch size $B$;
Output: Average global accuracy, server model;
Initialize server model, and broadcast $\theta^p$, $\mathcal{G}$, $p(y)$ to each
    active client;
Select the reliable clients with a dynamic threshold;
repeat
    for selected reliable $\mathcal{A}$ clients in parallel do
        $\theta_k \leftarrow \theta^p$;
        Generate pseudo-data $D_{fake}$;
        Extract the knowledge;
        for t = 1,..., $T$ do
            Sample from the generator;
            Update label counter $C_k$;
            $\theta_k \leftarrow \theta_k - \beta \nabla_{\theta_k} J(\theta_k)$;
            ▷ Optimize Equation (21);
        end for
        Send $\theta_k^p$, $C_k$ to server;
    end for
    Update server-side $\theta^p$ and $p(y)$ based $\{C_k\}_{k \in \mathcal{A}}$,
    $\theta^p \leftarrow \frac{1}{\mathcal{N}} \sum_{k=1\ k \in \mathcal{N}}^{k} \theta_k^p$;
    Generate feature distribution with lightweight,
    ▷ Optimize Equation (18);
    $\mathcal{G} \leftarrow \mathcal{G} - \alpha \nabla_{\mathcal{G}} J(\mathcal{G})$;
    Distill knowledge and guide local updates,
    ▷ Optimize Equation (22);
until training stop or converge
```

$F_{\text{invert}}(l_x)$. This attack does not require a priori knowledge, which is a poisoning attack method based on the data itself.

3) Gradient Inversion Attack: Similar to the work [16], we can recover utterance or emotion labels from the gradient. Following FedSGD, a single batch gradient computation is performed in each iteration. Each client samples the smallest batch $(x_i, y_i) \in D_i$ from the local private data set D_i, whose gradient is

$$\nabla w_{t,i} = \frac{\partial \mathcal{L}(F(x_i, w_t), y_i)}{\partial w_t} \tag{3}$$

the attack is as follows, the attacker creates virtual discourse and virtual sentiment labels, which are randomly initialized and then the model classifies to get the virtual gradient. Optimizing the virtual gradient goes close to the original gradient to make the virtual data close to the real data.

To defend against the multiple attacks mentioned above while maintaining the system performance, we propose a Federal Multiple Defense (FedMud) approach, the details of which are given in Algorithm 1. The approach consists of several important parts.

1) DP, to change the gradient by perturbed data, avoiding gradient inversion attacks, while training together with adversarial samples to make the model more robust.
2) Client Selection, effectively eliminating unreliable clients and preventing malicious data injection.
3) PE Model, which reduces information leakage by emphasizing relevant features and filtering irrelevant secondary features.
4) Federated KD under Security Mode, which utilizes a server-side lightweight generator to generate consensus knowledge to direct client-side modeling in a distilled way, and additionally, layered parameter sharing to ensure data privacy. The details are as follows.

C. Data Perturbation

To address the challenge of gradient inversion attack, inspired by [39], we try to make efforts at the data source. The fast gradient sign method (FGSM) becomes a crucial tool and it [40] was first proposed to defend against adversarial sample attacks. We discover that the neural network misclassifies when the data has been significantly changed. According to Goodfellow et al. [40], one speculation for generating an adversarial attack is that the linear nature of deep neural networks in high-latitude space can cause such an attack. Normally, stochastic gradient descent (SGD) makes the loss smaller to predict correctly. However, as the loss is added to the input, the output loss grows and the network begins to predict inaccurately.

Denote the raw data as x (here are the preprocessed speech features), the label as y, the emotion classification model as M, and the model parameters as θ. A perturbation η is applied to the FGSM, ε regulates the perturbation amplitude, and M is forward propagated for x to get the loss $\nabla_x \mathcal{L}(x, y, \theta)$. To manage the unlimited number of paradigms of the loss (the maximum value of each loss), the direction of the gradient is obtained using the symbolic function sign$(\cdot)$, rather than the gradient

$$\eta = \varepsilon \text{sign}(\nabla_x \mathcal{L}(x, y, \theta)) \tag{4}$$

where $||\eta||_\infty < \varepsilon$, the perturbation injected into the raw samples to obtain the adversarial samples

$$x_{\text{adv}} = x + \eta. \tag{5}$$

The perturbation of FGSM is effective because the adversarial samples go through the network as follows:

$$w^T x_{\text{adv}} = w^T(x + \eta) = w^T x + w^T \eta. \tag{6}$$

Let the weight vector have n dimensions and the average value of the weight vector elements be m. From $||\eta||_\infty < \varepsilon$, we obtain $w^T \eta < \varepsilon mn$, which has an effect due to η. $w^T \eta$ increases with the dimension of the weight vector, i.e., $n \uparrow$ and $w^T \eta \uparrow$, thus proving that the perturbation is valid.

Following that, the gradient is perturbed to improve privacy. There are data points in D_i^* that are comparable to those in D_i but with entirely different labels, $D_i^* = \{(x, (1/n) \sum_{j=1}^{n} e^j) | (x, y) \in D_i\}$, where e^j is the standard basis vector with 1 at position j. Assume $g(D_i)$ and $g(D_i^*)$ have comparable semantic but distinct categorization information. Because categorical information mainly affects classification performance, we remove the semantic information in $g(D_i)$

without significantly reducing model performance. Then, the $g(D_i)$ component with the same (or essentially the same) direction of $g(D_i^*)$ contains semantic information, and the other (orthogonal) components contain categorization information, and the corrected gradient is computed

$$\widetilde{g}(D_i) = g(D_i) - \mu g\left(D_i^*\right), \mu = \frac{g(D_i)^T g\left(D_i^*\right)}{g\left(D_i^*\right)^T g\left(D_i^*\right)}. \tag{7}$$

To combine the adversarial with the raw samples for training, which realizes data enhancement and improves adversarial robustness. Thanks to the large storage, high-speed computation, and high bandwidth of the edge servers (FL data owners), FGSM adversarial samples are prepared in advance at the edge side, i.e., preprocessing generation and caching mechanisms. The inclusion of adversarial samples eliminates adversarial sample attacks to some extent.

D. Clients Selection

Attackers maliciously inject toxic data in the FL client or flip labels to mislead the FL server-side discriminating, which can be readily overlooked or not easily discovered. A selection technique is suggested that requires screening for positive client knowledge. Unlike others that rely heavily on supplementary information about clients, we refer to client test accuracy values and select well-performing clients for error reduction and fast convergence.

Formally, let denote $K = \{1, \ldots, k\}$ clients, each has data set $D_i = \{D_i^{\text{train}} \cup D_i^{\text{test}}\}$ consisting of training and test sets. To select reliable (no data poisoning attack) clients, we should evaluate all clients according to the formula Accuracy(%) $=$ $(\sum D_i^{\text{true}}/D_i^{\text{test}}) \times 100$ (i.e., the quotient between the total number of samples tested correctly and the total number of samples tested) to obtain ascending test accuracy set ACC $= \{\text{acc}_1, \ldots, \text{acc}_k\}$. However, we must consider the number of unreliable to be deleted, as well as thresholds for distinguishing between correct and bad forecasts. *Deletion Number Setting*: referring to [41], a higher proportion of client participation in each round can save time for the global model to achieve the expected performance. The upper limit of deletion minima from ACC is $\text{Max}_{\text{del}} = {}^{1}/_{10} \times K$. *Dynamic Threshold Setting*: as the loop iterates, the global accuracy glob improves, as does the poisoned sample's latent ability. It is challenging to locate latently poisoned samples when the threshold remains constant. Here, the threshold is dynamically adjusted every ten global iterations, when it is the average of the accuracy of the previous ten global iterations.

E. Privacy-Enhanced Model

Curious servers can infer sensitive attributes since the network forward propagation process contains them that are finally collected. We investigate ways to filter these sensitive or redundant features such that only SER task-relevant features are retained, hence improving SER performance while keeping privacy. To protect against attribute inference attacks, an FL PE model is presented, with main components, including bi-directional gated recurrent unit (Bi-GRU) and Multihead Self-attention, as mentioned below.

Bi-GRU: Considering that emotion expression is time-dependent throughout the utterance, the Bi-GRU network [42] captures this temporal dependency well. The "gate" structure (update gate and reset gate) allows information to be selectively delivered in the hidden layer, remembering critical information while preventing gradient vanishing or gradient explosion. Unidirectional GRU is formulated in the following:

$$\begin{aligned} z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z) \\ r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r) \\ \widetilde{h_t} &= \tanh(W_h x_t + U_h(r_t \circ h_{t-1})) \\ h_t &= (1 - z_t) \circ h_{t-1} + z_t \circ \widetilde{h_t} \end{aligned} \tag{8}$$

where x_t is the input at time t, z_t and r_t denote the update gate and reset gate at time t, $h_t^{\sim}$ and h_t denote the candidate state and the hidden state at time t, the weight parameters W_z, W_r, W_h, U_z, U_r, and U_h, the bias parameters b_z and b_r, σ is the *sigmoid* nonlinear activation function, tanh is the nonlinear mapping function, and $\circ$ corresponds to the Hadamard product.

The Bi-GRU calculates the time series' hidden states in the forward and backward directions, respectively, and then combines the results from each time step. This ensures that the output at each time step contains both past and future contextual information.

Multihead Self-Attention: Previous research [43] indicates that not all parts of an utterance are emotionally relevant, with the main features having a crucial role, and that secondary irrelevant features also carry much sensitive information. We utilize the self-attention mechanism to focus on the main features (salient periods) in the utterance segments that are relevant to the emotions, to see through the autocorrelation, and to filter out the irrelevant parts thus preserving privacy. First, the hidden representation $H^e = [h_e^1, \ldots, h_e^T]$ is output through the Bi-GRU encoding context, where $h_e^i \in \mathbb{R}^{i \times 2d_e}$ denotes the forward-propagated and back-propagated hidden state splice. Next, the multihead self-attention network extracts salient features from the hidden output. Here, the input sequence features $[h_e^1, \ldots, h_e^T]$ represents T equal-dimensional vectors, and $[h_e^1]$ is the vector corresponding to the first segment in the discourse. Using the learnable weight matrix w, we progressively construct the query matrix q, key matrix k, and value matrix v. The dot product of q and k yields the attention score, further calculating attention distribution $d_i(d_i \in \mathbb{R}^T \quad \forall_i)$ using softmax, which is then normalized to discover which vectors are most connected with $[h_e^1]$. We can then determine the emotion-related traits. The formula is as follows:

$$\begin{aligned} q &= H^e w^q \\ k &= H^e w^k \\ v &= H^e w^v \\ d_i &= \text{Soft}\max\left(q_i k_i^T / \sqrt{T}\right) \times v_i. \end{aligned} \tag{9}$$

The multihead refers to the linear transformation of a query using distinct weight matrices to obtain multiple queries, all of which essentially require different types of relevant information, allowing the attention model to introduce more information into the context vector computation. To produce the attention model's output, the context vectors created by each attention header are stitched together and linearly processed using a weight matrix.

F. Federal Knowledge Distillation Under Security Mode

Consider the significance of FL client-side knowledge, where parameter/data sharing causes privacy leakage and communication overhead. Inspired by [31], we employ a server-side generator to learn the global view knowledge. On the one hand, given the goal label, a generator learned from user predictions produces feature representations that are consistent with the set of user predictions. The generator is then distributed to the users, who sample and augment the samples guide local training, and facilitate KD from other users. On the other hand, feature extraction and model prediction with client data on which shared model updates follow the FL protocol.

Next, we discuss the general problem of knowledge transfer, let denote the instance space as $x \subset \mathbb{R}^x$, the output space as $y \subset \mathbb{R}^y$, $\mathcal{T}$ denotes a domain consisting of a data distribution D over x and a truth labeling function τ, i.e., $\mathcal{T} : = \{ \text{D}, \tau\}$, $\tau : x \to y$. The model is parameterized by θ: $= [\theta^e; \theta^p]$, where the feature extractor $e : x \to z$ is parameterized by θ^e, $z \subset \mathbb{R}^z (z < x)$ is the latent feature space, the predictive classifier $p : z \to \Delta^y$ is denoted by the parameterization θ^p, where Δ^y is simplex over y. Then, the risk of the model parameterized θ over the domain $\mathcal{T}$ is defined as follows:

$$\mathcal{L}_{\mathcal{T}}(\theta) := \mathbb{E}_{x\sim D}\Big[\mathcal{L}\big(p(e(x;\theta^e);\theta^p),\tau(x)\big)\Big]. \tag{10}$$

1) Knowledge Extraction (Extract Discriminative Features and Filter Secondary Features): Because Zhang et al. [44] demonstrated that the importance of knowledge varies across local models, we aim to extract discriminative features X_{crucial} while hiding other irrelevant features $X_{\text{unrelated}}$. Bi-GRU captures the contextual dependencies of the utterance after convolutional neural networks (CNNs) to get the hidden representation

$$H_e = \text{BiGRU}(\text{CNN}(x)) = \left[h_e^1,\ldots,h_e^T\right]^{T\times 2d_e} \tag{11}$$

the multihead self-attention makes it possible for the end-to-end structure to learn emotion-related salient features of the utterance, indirectly filtering emotion-irrelevant redundant information, which often contains sensitive attributes, as shown below in the single-head attentional output representation

$$X^{\text{i}}_{\text{crucial}} = \text{Attention}\left(\left[h_e^{\text{i}}\right]^{2d_e}\right) = \left[h^{\text{i}}_{\text{attn}}\right]^{d_e} \tag{12}$$

the predictor MLP makes classification predictions for the spliced attention, where w is a linearly transformed weight matrix

$$y = \text{MLP}\Big(\text{concat}\big(h^1_{\text{attn}},\ldots,h^{\text{i}}_{\text{attn}},h^T_{\text{attn}}\big)\times w\Big). \tag{13}$$

To optimize the local parameters θ by minimizing the following loss, when the accuracy of the feature extraction influences the classification

$$\min_{\theta} \mathbb{E}_{x\sim D}\Big[\mathcal{L}\big(p(e(x;\theta^e);\theta^p),\tau(x)\big)\Big] \tag{14}$$

on the one hand, it affects global loss computation and updating as well as local model personalization. On the other hand, when server-side models or parameters are aggregated, if the extracted knowledge is inaccurate and the server-side generator obtains incorrect predictive labels, it directly affects the generated knowledge distribution. Our knowledge model can extract important features, filter sensitive information, and prevent attribute inference attacks.

2) Flexible Sharing of Layered Parameters: Deep model parameters are enormous and sharing the entire model in FL imposes a communication burden, we consider sharing some of the parameters. Parameter decoupling was proposed in [31] and [45], and Zhu et al. [31] proved that not sharing the feature extraction layer benefited local users significantly. We share only the prediction layer parameters θ^p_k, keep the feature extractor parameters θ^e_k localized, follow the FL protocol (FedAvg), and aggregate the updates after t iteration:

$$\theta^p \leftarrow \frac{1}{\mathcal{N}}\sum_{k=1}^{k}{}_{k\in\mathcal{N}}\theta^p_k \tag{15}$$

where $\mathcal{N}$ denotes the number of clients. This partial parameter sharing is less likely to leak privacy. Client updates will be given in the following section.

3) Avoid Being Misled by False Labels: In addition to extracting higher-level features, prediction labels from individual clients are required for knowledge generation of the server-side global view. Some malicious attackers have attempted data poisoning attacks by purposely flipping labels to drive the model in the direction of established predictions. It has been shown in [46] that deep neural networks prefer to learn most classes. To circumvent pseudo-label misdirection and defend against such undetectable attacks, the client selection method (Section III-B) robustly determines client knowledge and ensures that the correct predictive labels are transmitted to the server.

4) Lightweight Feature Distribution Generation: Catastrophic forgetting in deep networks affects system performance. To address the challenge, a lightweight generative scheme that balances performance and privacy has been proposed, which aims to extract global view knowledge of user data distributions and distill them into local models to guide their learning and mitigate knowledge forgetting. We consider first learning a conditional distribution that is consistent with the true data distribution

$$P^* = \arg\max_{P:y\to x} \mathbb{E}_{y\sim p(y)}\mathbb{E}_{x\sim p(x|y)}\big[\log p(y|x)\big] \tag{16}$$

where $p(y)$ and $p(y|x)$ are the target labels' ground-truth prior and posterior distributions, respectively. We further empirically approximate $p(y)$ and $p(y|x)$ to make the parameter p optimizable. First, the distribution of user training labels can be roughly represented as the prior distribution $p(y)$, which is obtained from the statistical distribution of the training labels when the model is uploaded. The posterior distribution comes from the integration knowledge of the user model and thus can be approximated as follows:

$$\log \widetilde{p}(y|x) \approx \frac{1}{K}\sum_{k=1}^{K} \log p(y|x; \theta_k). \tag{17}$$

When x is high dimensional, optimizing the above formulas causes computational overload, which is why we do not generate pseudo-data directly and cause privacy leakage by integrating θ_k directly. We investigate utilizing the conditional generator G of the parameterization $\mathcal{G}$ to recover a more compact distribution in the potential space z than the original data space, while optimizing the following objective:

$$\min_{\mathcal{G}} J(\mathcal{G}) := \mathbb{E}_{y\sim p(y)}\mathbb{E}_{z\sim G(z|y)}\left[\mathcal{L}\left(\sigma\left(\frac{1}{K}\sum_{k=1}^{K}\phi(z;\theta_k^p)\right), y\right)\right] \tag{18}$$

where $\phi(\cdot)$ is the predictor's logits output and $\sigma(\cdot)$ is a nonlinear activation function, optimizing the above equation only requires access to the predictive layer parameters θ_k^p of the user model. To satisfy the sample diversity, the Gaussian noise is added to the generator to obtain $z \sim G_{\mathcal{G}}(y, \varepsilon | \varepsilon \sim \mathcal{N}(0, I))$. In summary, the loss $\mathcal{L}_{\text{server}}$ of the server-side training generator consists of the teacher loss, the student loss, and the diversity loss, i.e.,

$$\mathcal{L}_{\text{server}} = \mathcal{L}_{\text{teacher}} + \mathcal{L}_{\text{student}} + \mathcal{L}_{\text{diversity}}. \tag{19}$$

Knowledge integration of global distributions is associated with cross-domain analysis, and we explore generalized boundaries to develop theoretical relationships, which build on existing techniques for domain adaptation [47], [48]. Denote $h : z \to y$ a prediction hypothesis, a hypothesis class $\mathcal{H} \subseteq \{h : z \to y\}$, and two domain distributions $\mathcal{D}'$ and $\mathcal{D}''$, [48] evaluates the distance between the two distributions on the hypothesis space $\mathcal{H}-$divergence

$$d_{\mathcal{H}}(\mathcal{D}', \mathcal{D}'') := 2 \sup_{A \in A_{\mathcal{H}}} |\Pr_{\mathcal{D}'}(A) - \Pr_{\mathcal{D}''}(A)| \tag{20}$$

where $A_{\mathcal{H}}$ is a measurable subset satisfying $h \in \mathcal{H}$ under domain distributions $\mathcal{D}'$ and $\mathcal{D}''$. Moreover, $\mathcal{H}\Delta\mathcal{H} := \{h(z) \oplus h'(z), h, h' \in \mathcal{H}\}$ is defined as the symmetrically distinct hypothesis spaces, where $\oplus$ denotes the heterodyne operation. Theorem 1 illustrates insights into the performance of integration on global distributions. See section Appendix A for proofs of Theorem 1.

5) Knowledge Distillation: After broadcasting the learned lightweight generator $G_{\mathcal{G}}$ to FL clients, each client model can sample from $G_{\mathcal{G}}$ to obtain an augmented representation of the feature space. Hence, the objective of the local model θ_k is optimized to increase the probability of ideal prediction for the augmented samples

$$\begin{aligned}\min_{\theta_k} \; J(\theta_k) &:= \frac{1}{\widehat{\mathcal{D}}_k} \sum\nolimits_{x_i \in \widehat{\mathcal{D}}_k} \left[\mathcal{L}\left(p(e(x_i; \theta_k^e); \theta_k^p), \tau(x_i)\right)\right] \\ &+ \widehat{\mathbb{E}}_{y\sim p(y)}\widehat{\mathbb{E}}_{z\sim G(z|y)}\left[\mathcal{L}\left(p(z; \theta_k^p); y\right)\right]\end{aligned} \tag{21}$$

where the former term in (23) denotes the empirical risk of local data $\widehat{\mathcal{D}}_k$.

Instead of the global model, knowledge is distilled to the local model. To enhance the generalization performance by matching distributions over the latent space z, transferring the inductive bias and directly guiding the local learning. The prior distribution of labels is denoted as $p(y)$ and the generator-derived conditional distribution is denoted as $g(z|y) : y \to z$, the local model θ_k is optimized by minimizing conditional KL-divergence from the generator and client distributions

$$\begin{aligned}\max_{\theta_k} \; & \mathbb{E}_{y\sim p(y)}\mathbb{E}_{z\sim g(z|y)}\left[\log p(z|y; \theta_k)\right] \\ & \equiv \min_{\theta_k} D_{KL}\left[g(z|y) || p(z|y; \theta_k)\right]\end{aligned} \tag{22}$$

then, the local model is updated as follows:

$$\theta_k \leftarrow \theta_k - \beta \nabla_{\theta_k} J(\theta_k) \tag{23}$$

where β is the local learning rate. To summarize, the client-side loss consists of three components: 1) the local prediction loss $\mathcal{L}_{\text{local}}$; 2) the potential loss of generator-augmented sample prediction $\mathcal{L}_{\text{samples}}$; and 3) the teacher loss $\mathcal{L}_{\text{teacher}}$ (using unduplicated labels), i.e.,

$$\mathcal{L}_{\text{client}} = \mathcal{L}_{\text{local}} + \mathcal{L}_{\text{samples}} + \mathcal{L}_{\text{teacher}}. \tag{24}$$

To summarize, a generator is utilized to generate consensus knowledge that guides client learning through KD. The effectiveness and robustness of our approach to data heterogeneity will be demonstrated in Section IV.

IV. Experiment Evaluation and Results

A. Model and Configurations

In this work, three main models are involved, the PE model, the attack model, and the generator model. For the PE model (emotion recognition model), a 2-layer CNN is first employed to extract high-level features from the EmoBase utterance feature set provided by opensmile toolkit [49]. The BiGRU follows, which performs better in capturing the context [50], and to avoid overfitting. Let us set the latent dimension to 128 (128 × 2 for bidirectionality), and set the dropout as 0.2. Then, a transformer encoder follows [43] based on multihead self-attention for focusing on emotion-related discriminative features and filtering irrelevant information, where the multihead is set to 8. The final MLP layer serves as the predictor θ_k^p, and the preceding layers serve as the feature extractor θ_k^e. The FL server and clients started with the same model, and the individual clients got a localized heterogeneous model structure with various model parameters along with the training iterations.

For the attack model, which consists of a 3-layer CNN feature extractor and classifier, the ith layer weight Δw_i of the gradient g from backpropagation is injected into the CNN to compute the hidden representation h_i. After flattening, it is connected with the bias Δb_i of the corresponding layer and fed into the classifier for gender prediction. Finally, the performance of the attack model is evaluated using the shared model update generated in FL.

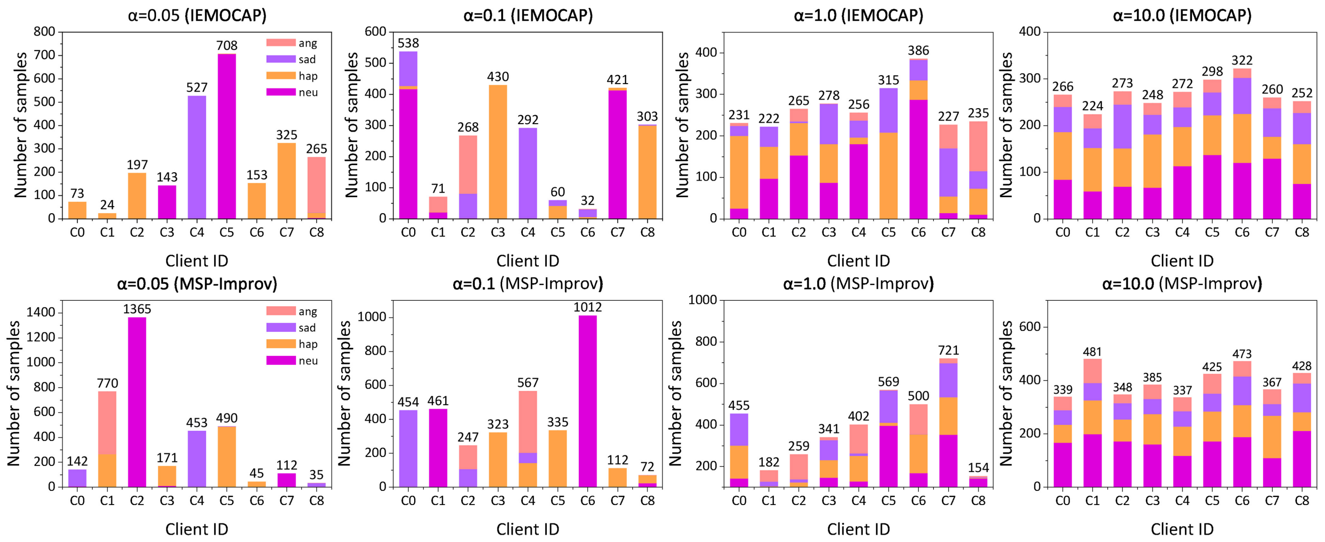

Fig. 3. Data dividing with the Dirichlet(α) distribution on the IEMOCAP and MSP-Improv data sets, visualizing the non-iid distribution from each client, with the x-axis denoting the client ID and the y-axis denoting the number of training samples corresponding to the four emotion classification labels (i.e., angry, sad, happy, and neutral). The heterogeneity coefficient α is set to {0.05, 0.1, 1.0, 10.0}, with smaller values indicating more heterogeneity.

The generator model is also designed based on the MLP network, which requires a noise vector and a one-hot labeling vector as inputs, and we add diversity loss to train the model to enrich the variety of generations. Other settings are as follows, we set local epoch as 10, global iterations as 100, batch size as 64, and generator nosie as 32. We have implemented the proposed approach on a Linux server with an NVIDIA GeForce RTX 2080Ti GPU and 64-GB RAM.

B. Data Preprocessing (Non-IID)

To achieve speaker independence (IEMOCAP has 10 speakers, MSP-Improv has 12 speakers), and to allow more data to participate, we divided the training and test data sets by 8:2, set up nine clients, and for the IEMOCAP data set, one more speaker data (duplicates) needed to be added. We modeled the data in two ways to investigate the influence of non-iid: 1) speaker independent and 2) nonspeaker independent.

For non-speaker-independent division, following the technique [14] to model non-iid data distributions utilizing Dirichlet(α), where a smaller α means more heterogeneous. Figs. 3 and 4 visualize the training set partitioning (based on emotion category and number of utterances) for various heterogeneity coefficients $\alpha = \{0.05, 0.1, 1.0, 10.0, \infty\}$, which correspond to the test data set partitioning in two ways.

1) Divide the test data set by Dirichlet(α) using the same heterogeneity coefficients as $\alpha = \{0.05, 0.1, 1.0, 10.0, \infty\}$.
2) Divide without heterogeneous operations and each client has the same collection of test data sets.

For speaker-independent division, as illustrated in Fig. 5, each client has an independent speaker training data set, and the test data set is modeled uniformly by heterogeneity coefficient $\{\infty\}$, resulting in the test data set close to the independently identically distributed (iid).

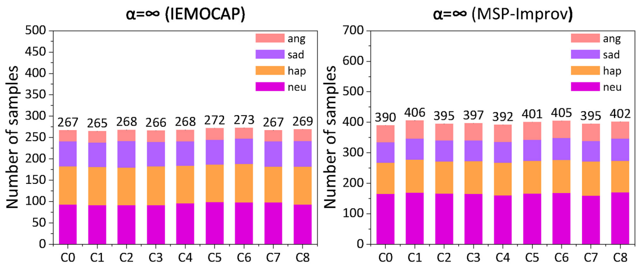

Fig. 4. Referring to the division in Fig. 3, the heterogeneity coefficient α is set to $\{\infty\}$ to visualize the distribution of training samples across clients on the IEMOCAP and MSP-Improv data sets, where the x-axis denotes the client ID, and the y-axis denotes the number of training samples corresponding to the four emotion classification labels.

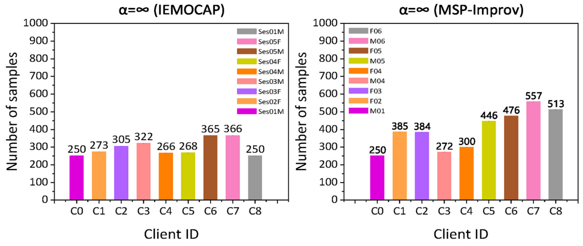

Fig. 5. Visualize the distribution of training samples for each client by dividing in a speaker-independent manner, with the x-axis denoting the client ID and the y-axis denoting the number of training samples, each client has only one speaker-independent data, at which point the test data set from each client is divided according to a heterogeneity coefficient of $\{\alpha = \infty\}$.

C. Baselines and Evaluation Metrics

Baselines: FedAvg [41] is a classical FL algorithm that utilizes parameter averaging for aggregation. FedProx [8] enhances FedAvg-based local objectives by providing regularization of proximal terms for local training. FedDistill+ is an enhanced data-free KD approach based on FedDistill [26], which shares model parameters and labeled logic vectors for fair comparisons. FedGen [31] develops a global generator

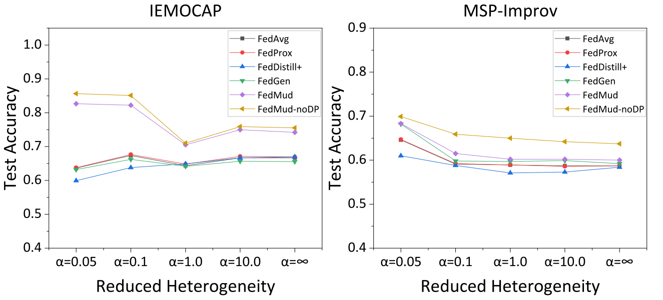

Fig. 6. Performance w.r.t data heterogeneity according to the Dirichlet(α) partition of Section IV-B1 .

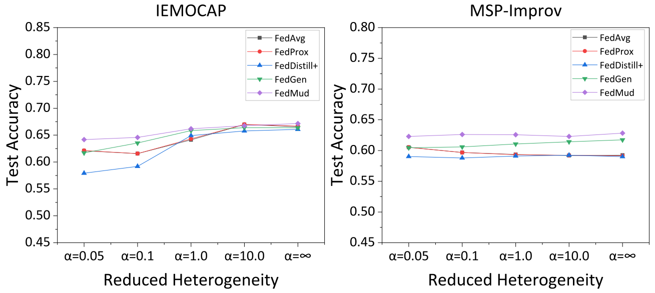

Fig. 7. Performance w.r.t data heterogeneity according to the Dirichlet(α) partition of Section IV-B2 .

TABLE I

Test Accuracy Comparison for Speaker-Independent Data Division (Section IV-B) using Four Models

Dataset	FedAvg	FedProx	FedDistill+	FedGen	FedMud
IEMOCAP	0.671	0.670	0.580	0.674	**0.723**
MSP-Improv	0.585	0.585	0.570	0.589	**0.635**

by sharing local labels and distilling consensus knowledge to guide local learning.

Evaluation Metrics: The common metrics global test accuracy (Test acc) is employed to measure system performance, while gender prediction accuracy (Attack acc) and Unweighted Average Recall (Attack uar) are used to evaluate defense capabilities.

D. Impact of Data Heterogeneity

To demonstrate algorithmic robustness, we analyze the performance of several algorithms under data heterogeneity in three scenarios, and we find that FedMud outperforms the other baselines by a considerable margin.

1) When both train and test data sets are divided by heterogeneity coefficients, all results are shown in Fig. 6. FedMud-noDP shows the corresponding performance when there is no DP. As a DP method, FGSM defends against gradient reversal and adversarial attacks by modifying the gradient and generating adversarial samples, but it always reduces performance.

   For IEMOCAP, FedMud consistently maintains a performance advantage, which confirms our motivation to mitigate distributional differences across clients by selectively aggregating local knowledge in a data-free manner to guide local learning. When the data distribution is highly heterogeneous ($\alpha \in \{0.05 \rightarrow 1.0\}$), FedMud is up to 82.7% ($\alpha = 0.05$), achieving a performance difference of around 12.0% (compared to $\alpha = 1.0$). With this distribution, the other baselines show an increasing and then decreasing trend, while FedMud continues to decrease. On the one hand, since the test data set also adopts a heterogeneous division, which may appear to be heavily skewed in terms of emotion categories. When random predictions are made across clients, it greatly increases the likelihood that some clients will predict only a single emotion category, while the skewed majority of emotions are more likely to be hit correctly, resulting in a higher test accuracy. On the other hand, during training, the perturbed data brings sample enhancement while exacerbating the data imbalance, and the neural network tends to learn the majority class while ignoring the minority class [26]. FedMud maintains about a 9.0% advantage as the data distribution approaches the iid ($\alpha > 1$). The conclusion is the same for MSP-Improv. However, due to the various distributions, FedMud does not obtain as significant a boost as IEMOCAP, but still outperforms the other algorithms for different heterogeneous distributions.

2) FedMud gets the best performance on both data sets while keeping the training data set heterogeneous and the test data set the same. Fig. 7 reports the differences across baselines. When highly heterogeneous ($\alpha \in \{0.05 \rightarrow 1.0\}$), FedGen performs stably thanks to its knowledge refinement, attenuating distributional differences through distillation. FedMud further improves performance through knowledge selection and discriminative feature extraction, while also defending against multiple attacks. These filtered and shared knowledge are not accessible via baselines, such as FedAvg and FedProx. FedDistill+, as an improvement over the data-free KD baseline, is susceptible to heterogeneity, with performance lower than FedAvg when $\alpha < 1$ and slightly better or about the same when $\alpha > 1$, indicating that FedDistill+ shared logits and some model parameters are insufficient to handle user heterogeneity.

3) When the training data set is speaker independent and the test data set is uniformly divided by the heterogeneity coefficient ∞, as shown in Table I, FedMud maintains robust and good performance despite cross-domain and feature heterogeneity. There is an improvement of around 4.9% (IEMOCAP) and 4.6% (MSP-Improv) compared to the suboptimal FedGen.

In conclusion, we find that FedMud has superior performance and algorithmic robustness, even when the devices and data are heterogeneous. It can weaken the heterogeneous distribution among each other, alleviate knowledge forgetting, and achieve performance gain through selective knowledge fusion and KD.

E. Ablation Experiments

We conducted ablation examinations, as shown in Fig. 8, to evaluate the performance increase and privacy

TABLE II

Parameter Sensitivity Analysis on Different Indicators (Dropout, Learning Rate, FGSM Eps, and Generator Noise) Using Data Set D_p for Emotion Recognition and D_h for Attribute Inference Attack. Test Accuracy and the Unweighted Average Recall (UAR) Scores of Gender Prediction Are Reported

		IEMOCAP(D_p) – MSP-Improv(D_h)			**MSP-Improv(D_p) – IEMOCAP(D_h)**		
Sensitivity	**Factor**	**Test acc**	**Attack acc**	**Attack uar**	**Test acc**	**Attack acc**	**Attack uar**
Dropout	0.2	**0.723**	0.503	0.503	**0.635**	0.531	0.496
	0.4	0.705	0.501	0.501	0.624	0.552	0.502
	0.6	0.703	0.500	0.500	0.620	0.550	0.501
LR	0.01	0.723	/	/	**0.635**	/	/
	0.001	**0.732**	/	/	0.582	/	/
	0.0001	0.684	/	/	0.580	/	/
FGSM_eps	0.1	0.720	0.502	0.502	**0.644**	0.535	0.505
	0.25	**0.723**	0.503	0.503	0.635	0.531	0.496
	0.35	0.715	0.505	0.505	0.578	0.556	0.503
Noise	2	**0.735**	0.501	0.501	0.627	0.553	0.499
	8	0.719	0.497	0.497	0.622	0.552	0.498
	32	0.723	0.503	0.503	**0.635**	0.531	0.496

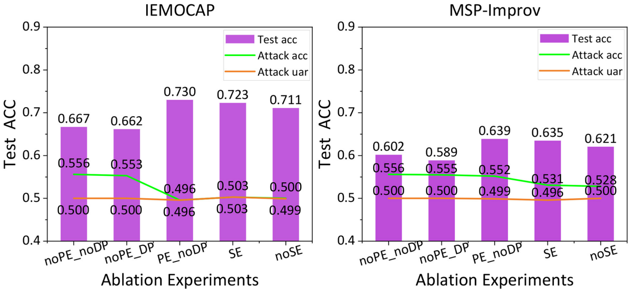

Fig. 8. Impact of different modules on system performance and gender prediction attacks, where noPE_noDP indicates the absence of PE, DP, and SE modules, noPE_DP indicates the absence of PE and SE modules, PE_noDP indicates the absence of DP and SE modules, SE indicates the presence of SE, PE, and DP modules, and no_SE indicates the absence of SE only.

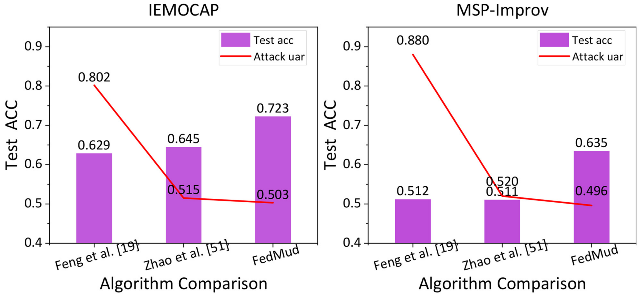

Fig. 9. Comparison of related approaches on system performance and attacks.

protection brought about by the additional modules. Among them are PE Module (including BiGRU and multihead self-attention), Client Selection Module (SE), and DP Module.

We observe that PE significantly improves the SER system performance by 6.1% (IEMOCAP) and 4.6% (MSP-Improv), while gender inference is weakening, with the attack acc trending toward IEMOCAP (0.553 → 0.503) and MSP-Improv (0.555 → 0.531), with lower values indicating stronger defense. This confirms the positive contribution of the PE module in focusing on discriminatively important features and filtering redundant information. DP modifies the gradient and generates adversarial samples through perturbation intending to prevent gradient reversal attacks and adversarial attacks. The participation of perturbed data in training results in a slight degradation of the performance. This degradation is a positive move that can be accepted within a certain range of magnitude. SE aims to help the system filter unreliable clients, avoid erroneous knowledge aggregation, prevent data poisoning attacks, and improve the overall performance through dynamic selection, and this improvement will be more prominent in real environments.

F. Performance and Attribute Inference Attack

According to previous work [51], the majority of information leakage occurs in the early layers of ML models. Fig. 9 compares gender inference attack (Attack UAR) and performance test (Test ACC). In contrast to Feng et al. [19], while Zhao et al. [51] showed insignificant improvement in performance, it substantially improves in defense against attribute inference attacks, down to 51.5% and 52.0%, respectively. In fact, we have a distinct advantage with a direct 8.0% and 12.0% breakthrough in performance on both data sets in the same setting, and the defense falls below the chance level (about 50.0%).

G. Sensitivity Analysis

Impact of Dropout: As a regularization technique, dropout may affect attribute inference attacks by randomly disabling activations between neurons [51]. Similar to [19], we evaluate this hypothesis by setting dropout to $\{0.2, 0.4, 0.6\}$ after the BiGRU layer and the first sense layer of the MLP classifier. As shown in Table II, the performance weakened as dropout increased, indicating that more SER-related features failed, as disabling more neuron activations normally makes the model more fragile and vulnerable, while the attack uar for gender inference remained almost unchanged, due to the dropout also drops and hides gender-related features. However, when using

TABLE III
IMPACT OF LEARNING EFFICIENCY FROM DIFFERENT LOCAL EPOCH

Dataset	Local-epoch	FedAvg	FedDistill+	FedProx	FedGen	FedMud
IEMOCAP	**5**	0.666	0.584	0.665	0.667	**0.731**
	10	0.671	0.580	0.670	0.674	0.723
	20	0.672	0.573	0.671	0.673	0.693
MSP-Improv	5	0.587	0.573	0.587	0.605	0.615
	10	0.585	0.570	0.585	0.589	**0.635**
	20	0.584	0.558	0.585	0.603	0.622

TABLE IV
IMPACT OF COMMUNICATION OVERHEAD FROM DIFFERENT GLOBAL ITERATION AND LOCAL ITERATION

Dataset	Global-local	FedAvg	FedDistill+	FedProx	FedGen	FedMud
IEMOCAP	50-10	0.669	0.578	0.670	0.678	0.716
	100-10	0.671	0.580	0.670	0.674	**0.723**
	150-10	0.672	0.578	0.672	0.673	0.708
MSP-Improv	50-10	0.587	0.568	0.586	0.604	0.628
	100-10	0.585	0.570	0.585	0.589	**0.635**
	150-10	0.587	0.567	0.588	0.604	0.629

TABLE V
COMPUTATIONAL COMPLEXITY (MACS), NUMBER OF PARAMETERS (PARAMS), AND MEMORY USAGE GENERATED BY DIFFERENT LOCAL MODELS

Local Models	MACs	Params	Memory Usage
CNN	78.22 KMac	31.90 k	124.60 Kb
CNN + BiGRU	426.72 KMac	94.74 k	370.00 Kb

IEMOCAP to mimic MSP-Improv, the attack acc improves, more speaker emotionally relevant representations are made in IEMOCAP, and gender is more inferred.

Impact of Learning Rate: We investigate the impact of different learning rates on system performance, which directly influences model convergence. Too large causes non-convergence, while too small leads to slow convergence or failure to learn. Table II shows that different learning rates produce different results, with increasing values contributing to a benefit effect. The optimal result is achieved by IEMOCAP and MSP-Improv at learning rates of 0.001 and 0.01, respectively.

Impact of FGSM Eps: As a DP technique, employing FGSM reduces the system performance, and the perturbation coefficient eps determines the perturbation magnitude. As shown in Table II, we set *eps* to $\{0.1, 0.25.0.35\}$. For IEMOCAP, the optimal performance is achieved when *eps* to 0.25. For MSP-Improv, the best performance is achieved when *eps* to 0.1, while the worst performance is achieved when *eps* to 0.35, and they differ by nearly 7.0%, thus, eps is parameter-sensitive. The effect on gender inference is weak, especially on the IEMOCAP. This suggests that client-side pseudo-data generated by FGSM obfuscates attributes such as gender, and it also relies on the privacy-preserving capabilities from FedMud to make client-side attributes less susceptible to inference.

Impact of Generator Gaussian Noise: The generator noise aims to make the generated samples diverse and enrich the distilled knowledge to better guide the client's learning. Too little noise may not cover the diversity, while too much can bring computational burden in addition to affecting the system judgment. As shown in Table II, we compare the effect of noise and conclude that for the IEMOCAP data set, less noise accomplishes higher performance, whereas for MSP-Improv, more noise better covers the sample diversity. One possible reason for this is that the data set contains a different number of speakers which affects the diversity requirements.

H. Learning Efficiency and Communication Overhead

In the FL settings, an effective system considers variables, such as privacy security, learning efficiency, and model communication overhead in addition to performance. Local models are trained to reach convergence after fewer epochs, which saves time and reduces the possibility of privacy leakage due to frequent interactions, especially in large federated training tasks involving multiple clients. Tables III and IV summarizes the learning performance and communication of several baselines across both data sets.

To begin, we compare the differences in learning efficiency, the default global iterations are 100. Most algorithms achieve the best results when the local epoch is 10, while FedMud achieves better performance than others at the start (epoch is 5). This indicates that when utilizing generative KD, the client learns consensus knowledge from others in fewer iterations. For MSP-Improv, the other baselines achieve good performance at first, whereas FedMud begins to harvest better performance when the local epoch is set to 10. Because the training set is larger, fewer local iterations make the generator learn less about the client information, keep continuous training, and more knowledge is acquired to guide the client learning. In summary, both benefit when the epoch is set to 10. We expect knowledge learning to be accomplished using fewer local epochs.

Next, we investigate the system's communication overhead, as reported in Table IV, which is constrained by the system bandwidth and the size of the transmission parameter. The less communication, the less data is transmitted to each other, which speeds up the system and decreases the risk of information leakage during data transmission. We consider setting the local epoch to 10, and the performance of each baseline remains pretty consistent as the global iterations increase, with only approximately 1.0% to 2.0% fluctuation. This suggests that any more communication frequency is not beneficial for performance improvement and keeping the communication frequency under control reduces the communication overhead. Moreover, the size of our model is only about 5.4M, which is at a low level and greatly reduces the communication overhead. For a more comprehensive explanation, we list the multiply-accumulate operations (MACs), number of parameters (Params), and Memory Usage from different local models. As shown in Table V, the simple local model produces fewer parameters and transmissions, corresponding to a slightly weaker performance. While maintaining superior performance and defense effects (referring to the findings of the ablation experiments), our model has no complex computational and memory requirements.

V. Conclusion and Future Works

In this article, we proposed a novel FL-based multiple defense approach, FedMud, which achieves privacy protection over multiple attack paths while maintaining superior system performance. To address the performance degradation caused by data non-iid in FL, we employed knowledge selection and knowledge integration strategies to distill the lightweight generated global distribution to the client to guide the local update, and theoretically analyzed the integration performance of cross-domain global distribution. For attribute inference attacks, we extracted discriminative emotion features and filtered other sensitive attribute information with the help of the feature filtering capability of deep networks. The gradient-based DP approach may successfully resist gradient reversal attacks and adversarial sample attacks by generating pseudo-data to participate in training by modifying the gradient. The validity of the client's information has a direct impact on knowledge integration and dictates the system's prediction direction, and a dynamic weighted selection technique was employed to remove untrustworthy knowledge and enable the system to converge in a positive direction. Other solutions, such as the parameter hierarchical sharing mechanism, help to reduce system communication overhead and preserve privacy.

Experiments on the IEMOCAP and MSP-Improv data sets (three non-iid data divisions) validate the approach's effectiveness, with performance improvements of at about 5.0% under the speaker-independent division, respectively, whereas for attribute inference attacks, gender prediction is reduced to approximately 50.0%, which corresponds to the chance level, when compared to other baselines. Expansion experiments on image data sets further demonstrate the excellent scalability and generalization of our proposed scheme. See section Appendix B for scalability and generalizability analysis.

In the future, we want to take deeper advantage of the convergence of cloud and edge computing, and federated aggregation employs a heterogeneous synchronization method to adapt to higher scale and real-time situations. Furthermore, we will provide a forward-thinking solution on FL big models based on the concept of large model migration learning.

Appendix

A. Proofs of Theorem 1

Generalization Bound: Consider an FL system with K users, $R : x \rightarrow z$ denotes the feature extraction function, lets denote the global distribution $\mathcal{D}$ for the global domain $\mathcal{T}$, the local distribution $\mathcal{D}_k$ for the kth local domain $\mathcal{T}_k$, and the empirical distribution $\widehat{\mathcal{D}}_k$, and h_k denotes the hypotheses learned on domain $\mathcal{T}_k$, then the global set of user hypotheses $h = (1/K)\sum_{k=1}^{K} h_k$. The empirical risk of a model trained on the global empirical distribution is $\widehat{\mathcal{D}} = (1/K)\sum_{k=1}^{K}\widehat{\mathcal{D}_k}$, while the upper bound on the risk of a collection of K local models over $\mathcal{D}$ is defined by the difference term of the distributions between $\mathcal{D}$ and $\mathcal{D}_k$, which has probability $1-\delta$

$$\mathcal{L}_{\mathcal{T}}\left(\frac{1}{K}\sum_{k=1}^{K} h_k\right) \le \frac{1}{K}\sum_{k=1}^{K}\widehat{\mathcal{L}}_{\mathcal{T}_k}(h_k) + \frac{1}{K}\sum_{k=1}^{K}\left(d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{\mathcal{D}}_k,\widehat{\mathcal{D}})+\lambda_k\right) + \sqrt{\frac{4}{m}\left(d\log\frac{2em}{d}+\log\frac{4K}{\delta}\right)} \qquad (25)$$

where $d_{\mathcal{H}\Delta\mathcal{H}}$ measures the difference between two distributions, m is the number of samples from each local distribution, and $\widehat{\mathcal{L}}_{\mathcal{T}_k}(h_k)$ signifies the empirical risk of $\mathcal{T}_k$. There exists $\mathbb{E}_{z\sim\widehat{\mathcal{D}}_k}[\mathcal{B}(z)] = \mathbb{E}_{x\sim\mathcal{D}_k}[\mathcal{B}(\mathcal{R}(x))]$ given a probability event $\mathcal{B}$. Following Theorem 1, we conclude that: 1) heterogeneity causes distributional differences and destroys the global model and 2) the performance of generalization is enhanced by using more empirical data. For a theoretical explanation of Theorem 1, see section Appendix of literature [14].

TABLE VI
Network Architecture for Different Local Models

Dataset	Local Model	Hyperparameter Settings
EMNIST	CNN MLP	[1, 6] [6, 16] [Flatten] [784, 32] [32, 26]
	LeNet MLP	[1, 16, MaxPool2d] [16, 32, MaxPool2d] [800, 120] [120, 84] [84, 26]

B. Scalability and Generalizability Analysis

To further explore the scalability of our proposed scheme, we conducted digital character classification experiments on the image data set EMNIST as detailed below.

Data Sets and Data Preprocessing: We employ the EMNIST [52] Letters category data, which consists of 124 800 training sets and 20 800 test sets. Following FedGen [31], the non-iid data distributions were modeled using Dirichlet(α) based on the heterogeneity coefficients $\{0.05, 0.1, 1.0, 10.0\}$.

Experimental Configurations: We allocate 20 Clients, of which 50.0% are activated, to simulate the situation where real-world clients go offline abnormally. We perform 200 global iterations and 20 local updates with a batch size of 32, a learning rate of 0.01, and a DP FGSM eps of 0.25. For local models, two network architectures are designed for comparison, CNN+MLP-based and LeNet-based, respectively. Their last MLP layer is considered a prediction layer, and all the previous layers are considered feature extraction layers, and the network parameters are shown in Table VI based on the above models, we then extend the experiments with the addition of a DP module, resulting in four comparison methods, namely, CNN, CNN+ (with DP), LeNet, and LeNet+ (with DP).

Performance Comparison: FedMud scales well to image classification tasks and achieves robust performance. The generic federated distillation-based framework allows for the selection of local models adapted to the task, as shown in Fig. 10, and the system performs well and significantly outperforms this work [31] when the local models are CNN and LeNet networks, and it is immune to user heterogeneity. DP helps against gradient reversal attacks and adversarial attacks, comparing the case without DP, the performance based on CNN+ and LeNet+ is weakened, while still maintaining a

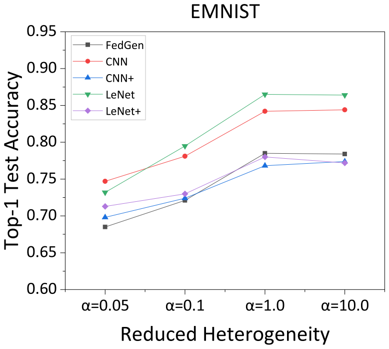

Fig. 10. Performance w.r.t local models with different network architectures under heterogeneous partitioning of MNIST data sets.

TABLE VII
COMPUTATIONAL COMPLEXITY (MACS), NUMBER OF PARAMETERS (PARAMS), AND MEMORY USAGE GENERATED BY DIFFERENT LOCAL MODELS

Local Model	MACs	Params	Memory Usage
CNN	88.70 KMac	26.96 k	105.30 Kb
LeNet	3.12 MMac	213.90 k	835.50 Kb

comparable performance with the work [31]. The above results match the SER task, which means that the system gets more defense while losing a little acceptable accuracy. Experiments demonstrate the cross-domain scalability and flexibility of our scheme.

Computation Complexity and Space Occupation: In an FL setting, the local model design not only affects the system performance, but also relates to the computational and storage costs. Table VII shows the MACs, Params, and Memory Usage of different local models. Obviously, the more complex the model, the higher the MACs and Params. In different scenarios, complex models may bring little performance gain but increase the system operation cost. While maintaining sufficient privacy and acceptable performance, our scheme enables the use of lightweight models and reduces the computation, storage, and transmission costs of the system through lightweight distillation and hierarchical sharing mechanisms.

REFERENCES

[1] R. Chatterjee, S. Mazumdar, R. S. Sherratt, R. Halder, T. Maitra, and D. Giri, "Real-time speech emotion analysis for smart home assistants," *IEEE Trans. Consum. Electron.*, vol. 67, no. 1, pp. 68–76, Feb. 2021.

[2] Z. Farhoudi and S. Setayeshi, "Fusion of deep learning features with mixture of brain emotional learning for audio-visual emotion recognition," *Speech Commun.*, vol. 127, pp. 92–103, Mar. 2021.

[3] L. Tan et al., "Speech emotion recognition enhanced traffic efficiency solution for autonomous vehicles in a 5G-enabled space–air–ground integrated intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2830–2842, Mar. 2022.

[4] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," 2016, *arXiv:1602.05629*.

[5] L. Lin and X. Zhang, "PPVerifier: A privacy-preserving and verifiable federated learning method in cloud-edge collaborative computing environment," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8878–8892, May 2023.

[6] 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.

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

[8] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst. (MLSys)*, 2020, pp. 429–450.

[9] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 5132–5143.

[10] D. Rothchild et al., "FetchSGD: Communication-efficient federated learning with sketching," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 8253–8265.

[11] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 10713–10722.

[12] W. Huang, M. Ye, and B. Du, "Learn from others and be yourself in heterogeneous federated learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 10143–10153.

[13] L. Sun and L. Lyu, "Federated model distillation with noise-free differential privacy," 2020, *arXiv:2009.05537*.

[14] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Proc. Adv. Neural Inf. Proces. Syst. (NeurIPS)*, 2020, pp. 2351–2363.

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

[16] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. Adv. Neural Inf. Proces. Syst. (NeurIPS)*, 2019, pp. 14747–14756.

[17] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Security Privacy. (SP)*, 2017, pp. 3–18.

[18] J. Geng et al., "Towards general deep leakage in federated learning," 2021, *arXiv:2110.09074*.

[19] T. Feng, H. Hashemi, R. Hebbar, M. Annavaram, and S. S. Narayanan, "Attribute inference attack of speech emotion recognition in federated learning settings," 2021, *arXiv:2112.13416*.

[20] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 1369–1378.

[21] Z. Ren, A. Baird, J. Han, Z. Zhang, and B. Schuller, "Generating and protecting against adversarial attacks for deep speech-based emotion recognition models," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2020, pp. 7184–7188.

[22] K. Wei et al., "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[23] N. Agrawal, A. Shahin Shamsabadi, M. J. Kusner, and A. Gascón, "QUOTIENT: Two-party secure neural network training and prediction," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, 2019, pp. 1231–1247.

[24] M. Dias, A. Abad, and I. Trancoso, "Exploring hashing and cryptonet based approaches for privacy-preserving speech emotion recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2018, pp. 2057–2061.

[25] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedAVG on non-IID data," 2019, *arXiv:1907.02189*.

[26] H. Seo, J. Park, S. Oh, M. Bennis, and S.-L. Kim, "federated knowledge distillation," 2020, *arXiv:2011.02367*.

[27] F. Sattler, T. Korjakow, R. Rischke, and W. Samek, "FedAUX: Leveraging unlabeled auxiliary data in federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 5531–5543, Nov. 2023.

[28] G. K. Nayak, K. R. Mopuri, V. Shaj, V. B. Radhakrishnan, and A. Chakraborty, "Zero-shot knowledge distillation in deep networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 4743–4751.

[29] R. G. Lopes, S. Fenu, and T. Starner, "Data-free knowledge distillation for deep neural networks," 2017, *arXiv:1710.07535*.

[30] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, "Mode seeking generative adversarial networks for diverse image synthesis," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 1429–1437.

[31] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 12878–12889.

[32] F. Mireshghallah, M. Taram, A. Jalali, A. T. T. Elthakeb, D. Tullsen, and H. Esmaeilzadeh, "Not all features are equal: Discovering essential features for preserving prediction privacy," in *Proc. ACM Web Conf. Compan. World Wide Web Conf. (WWW)*, 2021, pp. 669–680.

[33] T. Feng, R. Peri, and S. Narayanan, "User-level differential privacy against attribute inference attack of speech emotion recognition in federated learning," 2022, *arXiv:2204.02500*.

[34] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *Proc. Lect. Notes Comput. Sci. (ESORICS)*, 2020, pp. 480–501.

[35] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" in *Proc. Adv. Neural Inf. Proces. Syst. (NeurIPS)*, 2020, pp. 16937–16947.

[36] Y. Chang, S. Laridi, Z. Ren, G. Palmer, B. W. Schuller, and M. Fisichella, "Robust federated learning against adversarial attacks for speech emotion recognition," 2022, *arXiv:2203.04696*.

[37] C. Busso et al., "IEMOCAP: Interactive emotional dyadic motion capture database," *Lang. Resour. Eval.*, vol. 42, pp. 335–359, Nov. 2008.

[38] 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, pp. 67–80, Jan.–Mar. 2016.

[39] C. Xu, Z. Hong, M. Huang, and T. Jiang, "Acceleration of federated learning with alleviated forgetting in local training," 2022, *arXiv:2203.02645*.

[40] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[41] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat. (AISTATS)*, 2017, pp. 1273–1282.

[42] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[43] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Proces. Syst. (NeurIPS)*, 2017, pp. 5998–6008.

[44] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan, "Fine-tuning global model via data-free knowledge distillation for non-IID federated learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 10174–10183.

[45] Y. Wu, Y. Kang, J. Luo, Y. He, and Q. Yang, "FedCG: Leverage conditional GAN for protecting privacy and maintaining competitive performance in federated learning," 2021, *arXiv:2111.08211*.

[46] C. Fang, H. He, Q. Long, and W. J. Su, "Layer-peeled model: Toward understanding well-trained deep neural networks," 2021, *arXiv:2101.12699*.

[47] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. Adv. Neural Inf. Proces. Syst. (NeurIPS)*, 2006, pp. 137–144.

[48] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, pp. 151–175, May 2010.

[49] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: The munich versatile and fast open-source audio feature extractor," in *Proc. ACM Int. Conf. Multimed. (MM)*, 2010, pp. 1459–1462.

[50] D. Li, L. Sun, X. Xu, Z. Wang, J. Zhang, and W. Du, "BLSTM and CNN stacking architecture for speech emotion recognition," *Neural Process. Lett.*, vol. 53, pp. 4097–4115, Aug. 2021.

[51] H. Zhao, H. Chen, Y. Xiao, and Z. Zhang, "Privacy-enhanced federated learning against attribute inference attack for speech emotion recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2023, pp. 1–5.

[52] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "EMNIST: extending MNIST to handwritten letters," in *Proc. Int. Joint. Conf. Neural Netw. (IJCNN)*, 2017, pp. 2921–2926.

Haijiao Chen (Graduate Student Member, IEEE) received the M.S. degree in computer application technology from Xinjiang University, Urumqi, China, in 2017. He is currently pursuing the Ph.D. degree with Hunan University, Changsha, China.

He has published papers at the IEEE International Conference on Acoustics, Speech, and Signal Processing. His research interests include speech emotion computation, federated learning, deep learning, and privacy and security.

Mr. Chen is a Student Member of CCF.

Huan Zhao (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science and technology from Hunan University, Changsha, China, in 1989, 2004, and 2010, respectively.

She is currently a Professor with the College of Information Science and Engineering, Hunan University. She has published over 100 research papers in international journals and conferences, including *Information Processing and Management*, *Knowledge-Based Systems*, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, and IEEE International Conference on Acoustics, Speech, and Signal Processing. Her current research interests mainly include speech signal processing, cross-media retrieval, and natural language processing.

Zixing Zhang (Senior Member, IEEE) received the master's degree in physical electronics from Beijing University of Posts and Telecommunications, Beijing, China, in 2010, and the Ph.D. degree in computer engineering from the Technical University of Munich (TUM), Munich, Germany, in 2015.

He is currently a Full Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. From 2017 to 2019, he was a Research Associate with the Department of Computing, Imperial College London, London, U.K. Before that, he was a Postdoctoral Researcher with the University of Passau, Passau, Germany. To date, he has authored more than 110 publications in peer-reviewed books, journals, and conference proceedings, leading to more than 5000 citations (H-index 40). His research focuses on human-centred emotion and health computation.

Prof. Zhang serves as an Associate Editor of the IEEE TRANSACTIONS ON AFFECTIVE COMPUTING and the *Frontiers in Signal Processing*, an Editorial Board Member of the Nature Scientific Reports, and a Guest Editor of the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE.

Keqin Li (Fellow, IEEE) received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1985, and the Ph.D. degree in computer science from the University of Houston, Houston, TX, USA, in 1990.

He is currently a SUNY Distinguished Professor with the State University of New York, Albany, NY, USA, and a National Distinguished Professor with Hunan University, Changsha, China. He holds nearly 70 patents announced or authorized by the Chinese National Intellectual Property Administration. He has authored or coauthored more than 950 journal articles, book chapters, and refereed conference papers.

Dr. Li was a 2017 recipient of the Albert Nelson Marquis Lifetime Achievement Award for being listed in Marquis Who's Who in Science and Engineering, Who's Who in America, Who's Who in the World, and Who's Who in American Education for more than 20 consecutive years. He received the Distinguished Alumnus Award from the Computer Science Department, University of Houston in 2018. He received the IEEE TCCLD Research Impact Award from the IEEE CS Technical Committee on Cloud Computing in 2022 and the IEEE TCSVC Research Innovation Award from the IEEE CS Technical Community on Services Computing in 2023. He received several best paper awards from international conferences, including PDPTA-1996, NAECON-1997, IPDPS-2000, ISPA-2016, NPC-2019, ISPA-2019, and CPSCom-2022. He is among the world's top five most influential scientists in parallel and distributed computing in terms of single-year and career-long impacts based on a composite indicator of the Scopus citation database. He is a member of the SUNY Distinguished Academy. He is an AAAS Fellow, an IEEE Fellow, and an AAIA Fellow. He is a member of Academia Europaea (Academician of the Academy of Europe).