scientific reports

Check for updates

OPEN

Federated influencer learning for secure and efficient collaborative learning in realistic medical database environment

Haengbok Chung[1,2] & Jae Sung Lee[1,2,3]✉

Enhancing deep learning performance requires extensive datasets. Centralized training raises concerns about data ownership and security. Additionally, large models are often unsuitable for hospitals due to their limited resource capacities. Federated learning (FL) has been introduced to address these issues. However, FL faces challenges such as vulnerability to attacks, non-IID data, reliance on a central server, high communication overhead, and suboptimal model aggregation. Furthermore, FL is not optimized for realistic hospital database environments, where data are dynamically accumulated. To overcome these limitations, we propose federated influencer learning (FIL) as a secure and efficient collaborative learning paradigm. Unlike the server-client model of FL, FIL features an equal-status structure among participants, with an administrator overseeing the overall process. FIL comprises four stages: local training, qualification, screening, and influencing. Local training is similar to vanilla FL, except for the optional use of a shared dataset. In the qualification stage, participants are classified as influencers or followers. During the screening stage, the integrity of the logits from the influencer is examined. If the integrity is confirmed, the influencer shares their knowledge with the others. FIL is more secure than FL because it eliminates the need for model-parameter transactions, central servers, and generative models. Additionally, FIL supports model-agnostic training. These features make FIL particularly promising for fields such as healthcare, where maintaining confidentiality is crucial. Our experiments demonstrated the effectiveness of FIL, which outperformed several FL methods on large medical (X-ray, MRI, and PET) and natural (CIFAR-10) image dataset in a dynamically accumulating database environment, with consistently higher precision, recall, Dice score, and lower standard deviation between participants. In particular, in the PET dataset, FIL achieved about a 40% improvement in Dice score and recall.

Keywords On-device, Collaborative learning, Federated learning, Privacy, Security, Knowledge distillation

Deep-learning models[1] require large datasets for improved performance. Consequently, extensive datasets are stored in centralized data centers for training large models[2]. However, centralized training has faced notable criticisms[3], including issues related to data ownership, security, and resource capacity, particularly in hospitals. Data ownership concerns arise from the reluctance of data owners to share their data, even when it is stored in a centralized repository. Security issues become evident when malicious hackers breach hospitals' centralized databases, exposing sensitive information. According to recent reports[4], over 70% of hospitals' centralized databases have experienced data breaches in recent years, leading to significant increases in IT spending to address security vulnerabilities. Limited resource capacity is also a significant problem due to the rapidly growing size of models, which makes on-device personalization impractical. Additionally, sustainability, scalability, and cost are further concerns associated with centralized training. These challenges are particularly significant in the healthcare sector[5], where patient privacy and resource constraints are critical issues[6].

Federated learning (FL)[7,8] has emerged as a promising solution to these challenges and has garnered substantial interest in the medical domain[9]. In FL, clients perform independent and decentralized local training, and aggregate the model on the server until convergence. This approach eliminates the need for a centralized data center and enables personalization[10] with relatively small-scale models, thus mitigating the aforementioned

[1]Interdisciplinary Program in Artificial Intelligence, Seoul National University, Seoul, Korea. [2]Department of Nuclear Medicine, Seoul National University College of Medicine, Seoul, Korea. [3]Brightonix Imaging Inc., Seoul, Korea. ✉email: jaes@snu.ac.kr

concerns. Because FL augments security, scalability, and sustainability it encourages institutes to participate in training, culminating in superior deep learning performance at the end.

However, FL has several limitations[11,12]. First, the non-independently or non-identically distributed (non-IID[13]) nature of the data can lead to suboptimal performance compared to IID scenarios. Second, model aggregation often reduces personalization and introduces excessive communication overhead[14]. Third, since model parameters are exposed during aggregation, FL is vulnerable to various attacks[15–18] targeting these parameters. Specifically, malicious attackers can breach privacy through model inversion attacks or disrupt model performance through adversarial attacks. Additionally, the central server exposes FL to potential manipulation by malicious hackers or the server itself.

Several solutions have been proposed to address these issues. However, *they could not be a silver bullet*. For example, FedProx[19], MOON[20], FedAlign[21], and several other algorithms[13,22–24] have attempted to improve FL performance by addressing non-IID data issues. However, these efforts often overlook security concerns, leaving FL vulnerable to potential attacks. Techniques like differential privacy (DP) and secure multiparty computation (MPC) have been applied to FL to enhance security, but they come with trade-offs between security and accuracy, leading to performance degradation[11,25,26]. Furthermore, these solutions require additional computational or resource overhead. Some FL approaches[27–32] have leveraged knowledge distillation for secure and model-agnostic training, but their reliance on central servers makes them susceptible to server attacks. These approaches also face challenges when incorporating new low-performing clients due to the ensemble of logits from clients.

Furthermore, the scarcity of studies addressing dynamic databases-where the quantity of data held by clients increases randomly during training-is noteworthy because these conditions better simulate realistic hospital database environments. Most existing research has focused on optimizing FL performance within static database scenarios, where the amount of data remains constant. However, dynamically accumulating databases are more common in healthcare settings compared to static ones. Existing FL methods require adjustments to experimental setups in these these databases, such as hyperparameters, each time the data volume changes, which is impractical.

To address these limitations, we introduce federated influencer learning (FIL). FIL enhances FL by eliminating the need for central servers and model parameter exchanges in dynamically accumulating database scenarios. FIL comprises four stages: local training, qualification, screening, and influencing. It replaces the model aggregation steps found in conventional FL with a knowledge-based approach[33,34]. Therefore, the FIL eliminates the need for model parameter exchanges and enhances security and model agnoticism. In addition, because FIL is designed for dynamically accumulating databases, it does not require additional modifications to the experimental settings once the procedure begins. Unlike FL methods that employ knowledge distillation[31], FIL operates independently of central servers and generative models. Moreover, FIL differs from online student learning due to its multifaceted nature, involving multiple stages and local data heterogeneity. It employs a one-way distillation strategy for authorized participants, dynamically adapting based on their performance. This adaptability makes FIL resilient to the inclusion of new participants, as those with subpar performance cannot assume the role of influencer. Additionally, potential issues arising from noisy logit data produced by malicious influencers can be addressed by independently reassessing the qualification score associated with the influencer's logits through the administrator and followers during the screening stage.

The remainder of this paper is organized as follows: In the Related Works section, we review existing FL methods, focusing on security mechanisms and knowledge distillation techniques, and identify gaps in handling dynamically changing databases. In the Method section, we explain the terminologies and detail the FIL process, where participants are dynamically assigned roles as influencers or followers. In the Experiments and Results section, we demonstrate FIL's superior performance in classification and segmentation tasks across medical (chest X-rays, brain MRI, and head and neck PET) and natural image datasets (CIFAR-10), highlighting its robustness in a realistic database environment.

Our contributions can be summarized as follows:

- We propose a training paradigm, FIL, in which participants dynamically transition between roles as influencers or followers based on their performance. The FIL framework iteratively involves local learning, qualification, screening, and influencing steps until performance converges.
- We introduce a method that leverages a shared dataset to train and test participants' performances, enhancing participant consistency and boosting generalizability. The logits from the shared dataset are utilized when knowledge is distilled from the influencer to the followers.
- We demonstrated the benefits of FIL through various classification and segmentation experiments using medical and natural image datasets. Additionally, we compared FIL's performance with various settings.

Related works

Federated learning with security. Various security mechanisms, such as local differential privacy (LDP), distributed differential privacy (DDP), secure multi-party computation (MPC), homomorphic encryption (HE), trusted execution environment (TEE), and data compression are used to defend against various attacks[11,18]. DP introduces Gaussian or Laplace noise to model parameters or input data to hinder data reconstruction. MPC is a cryptographic technique that can encrypt model parameters. HE is a powerful tool for implementing the MPC. Furthermore, TEE creates secure and isolated areas for FL using separate hardware and trusted software.

However, these security mechanisms have several limitations. For instance, DP mechanisms involve noise-accuracy trade-offs. In addition, they are ineffective against attacks such as membership, model inversion, and adversarial attacks. Their scalability is also compromised due to significant computational overhead. MPC suffers from substantial computational overhead and requires additional resources, such as secure channels between parties. HE raises concerns regarding secret key ownership. TEE has limitations in interoperability and

requires specific hardware components and extensive memory storage. Moreover, data compression can lead to the loss of critical information. None of these methods offers complete protection against the aforementioned attacks. Therefore, we advocate replacing model parameter transitions with knowledge distillation to address these vulnerabilities. Knowledge distillation is also more efficient than model parameter transactions because it can be executed concurrently with local training.

Federated learning with knowledge distillation. A series of algorithms have been successively developed to enhance FL performance since the introduction of FedAvg[7]. FedProx[11] introduced regularization terms during model aggregation. Moreover, MOON[19] incorporated a contrastive learning concept to reduce the gap between local and global models while increasing the gap between the models of the previous and current rounds. FedAlign[20] improved performance by adapting GradAug. Several studies[35–38] have attempted to optimize model aggregation by transforming model parameters.

However, the transaction of model parameters is vulnerable to various attacks, including model inversion, model extraction, side-channel, and membership attacks. In particular, model inversion attacks involve malicious actors using model parameters to reconstruct training data. Additionally, attacks may disrupt model convergence. Adversarial, Byzantine, data poisoning, and model poisoning attacks can cause models to be incorrectly trained.

To address these limitations, distillation-based FL algorithms have been proposed. For example, FedMD[27] revolutionarily reformed the FL procedure using a publicly available dataset for clients. FedDF[28], similar to FedMD, employs unlabeled data for distillation without using it for local training. Federated mutual learning[29] applies deep mutual learning to FL. FedAD[30] trains clients, and then uses ensemble attention maps and logits to train a central server from scratch. FedFTG[31] employs a generative model to generate hard samples for aligning global and local models, eliminating the need for local datasets in training the global generative model. DaFKD[32] optimizes the ensemble procedure by evaluating the correlation between the distillation samples and the client dataset. However, many of these methods require substantial computational resources and memory overhead for central server training, making them susceptible to convergence attacks, where malicious clients may send incorrect server information or attention maps. Additionally, if the server intentionally distorts the training process or introduces a noisy model, overall performance may suffer. Some methods using generative models are also vulnerable to privacy leakage despite their data-free approach due to limitations in defense mechanisms.

Online knowledge distillation. The conventional approach involves using a pre-trained teacher model with a larger and more complex architecture to train smaller and simpler student models. However, this approach has limitations, including reliance on a teacher model and high time complexity. Online knowledge distillation was recently introduced to address these issues. This paradigm uses a one-stage training method involving multiple students, who are trained under diverse conditions using ensemble logits. Notable techniques within this framework include deep mutual learning (DML)[39], which aligns logits using KL-divergence, and KDCL[40], which aggregates logits from all students to minimize the gap between their outputs and each student's output. Peer collaborative learning[41] extends prior efforts by establishing a temporal mean teacher and peer ensemble teacher. Furthermore, FFSD[42] leverages ensemble logits and attention maps to train the leader students. Unlike these methods, our approach introduces a single influencer, or potentially more, with dynamically changing roles based on performance quality. The remaining participants benefit unilaterally from the influencer. Furthermore, FIL operates with heterogeneous private datasets and incorporates four stages, distinguishing it from conventional online knowledge distillation methods.

Methods

FL is a rapidly growing field addressing issues associated with centralized training[9]. However, problems[11] such as data ownership, security, limited resource capacity, and vulnerability to various attacks remain unresolved. These issues are particularly critical in the healthcare domain, which demands high confidentiality and performance. Many existing solutions to these problems, however, degrade either performance or efficiency. To address these challenges, we introduce a collaborative decentralized learning paradigm called federated influencer learning (FIL), which aims to overcome the limitations of traditional FL.

Terminologies

The FIL framework consists of participants (local nodes) who are managed by an administrator responsible for supervising and recording the training process. Participants start with equal status and then become influencers or followers based on their performance. The highest-performing influencer shares knowledge with the remaining followers. FIL iterates through these stages until convergence is achieved. Detailed explanations of the terminology used in this study are provided in Table 1, Table 2, and Table 3.

Problem definition

Based on aforementioned definitions, we can formulate an objective function for the FIL in Equation 1.

$$\min_{\omega_{1,\ldots,P}} \sum_{i=1}^{P} [F(\omega_i) + D(F(\omega_i), \mu)] \tag{1}$$

F is the loss function and D is the distance. μ represents the average loss value among all participants. This equation indicates that our objective is to find P local models which achieve both their own global minima and fairness simultaneously. Since we tested the local models on the shared test data, each participant's global minima and fairness were compatible.

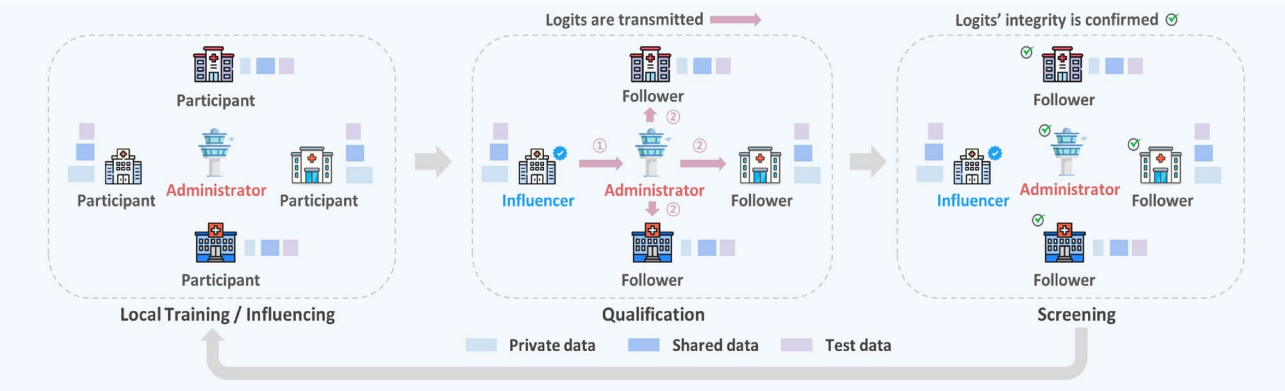

Fig. 1. Overall process of FIL. The FIL framework consists of four stages: local training, qualification, screening, and influencing. These stages are repeated continuously. In the local training stage, participants train their models using their private data and, optionally, shared data. In the qualification stage, the participant with the highest qualification score is selected as the influencer, while the remaining participants become followers. During the screening stage, the integrity of the logits from the influencer is verified by the administrator and followers. If no issues are found, the influencing stage proceeds in parallel with local training.

Term	Definition
Server	In FL, the process involves coordinating learning by aggregating model parameters from multiple client nodes. The global model is sent to the clients, who then perform private local training on their own data
Client	In FL, client nodes refer to the individual decentralized units that conduct private local training on their own data. These nodes send model parameters to the server and receive the aggregated global model in return
Logit	In classification, this term represents the final score for each class, whereas in segmentation, it refers to the output probability map for each class

Table 1. Basic terminologies used in our study. Server and clients are used in FL and logit is the key element for the FIL.

Term	Definition
Participant	Local nodes participating in FIL are referred to as participants rather than clients or a server, due to their equal status compared to the traditional server-client model in FL. Based on performance, participants assume temporary roles as "influencer" or "follower," with roles changing each round
Influencer	An influencer is a participant who has achieved the highest performance during the qualification stage. The influencer sends its logits to the remaining local nodes (followers) and imparts its knowledge
Follower	Followers are the participants who did not achieve the highest qualification score and are therefore not the influencer. They improve their performance by leveraging the knowledge shared by the influencer
Administrator	Instead of a central server, an administrator selects the influencer based on the submitted qualification scores from participants and distributes the logits from the selected influencer to all followers

Table 2. Terminologies regarding the topology of Federated Influencer Learning (FIL). Participants can become influencer or follower while administrator supervises entire training procedure apart from the participants.

Stage	Description
Local Training	Each participant independently trains their models using their private datasets
Qualification	The performances of the participants' local models are evaluated using a shared dataset. The participant with the highest-performing model is designated as the influencer, while the remaining participants become followers. The influencer's logits are sent to the administrator, who then forwards them to the followers
Screening	The administrator and followers verify the integrity of the received logits by recalculating the qualification score of the influencer's logits independently
Influencing	Followers improve their performances by distilling knowledge from the influencer's logits

Table 3. Terminologies regarding the stages in Federated Influencer Learning (FIL). FIL repeats four stages until the performance converges.

	# of Influencer	Q	NIH CXR14		CheXpert	
			AUC	SD	AUC	SD
SOLO	-	-	58.18	2.88	57.39	0.78
FedAvg	-	-	62.95	1.89	61.11	0.76
FedProx	-	-	63.58	3.69	63.61	0.16
MOON	-	-	62.45	1.40	59.30	0.34
FedAlign	-	-	61.23	0.98	60.81	0.31
FIL	1	O	65.43	1.76	65.83	0.12
FIL	1	X	**65.54**	0.70	60.66	0.10
FIL	2	O	64.20	**0.53**	**65.88**	**0.06**

Table 4. Multi-labe classification AUC results for the NIH Chest X-ray 14 (NIH CXR14) and CheXpert datasets in dynamic database environments. The configuration includes 10 rounds of local training, and 20 rounds of communication and influencing. The p-values of AUC scores between FIL and FL methods were 0.009 for NIH CXR14 and 0.013 for CheXpert.

CIFAR-10						
	# of Influencer	Q	IR = 20		IR = 40	
			ACC	SD	ACC	SD
SOLO	-	-	61.16	7.09	59.66	4.67
FedAvg	-	-	77.58	1.59	80.39	**0.14**
FedProx	-	-	76.90	**0.11**	80.05	0.47
MOON	-	-	76.12	1.07	79.73	1.74
FedAlign	-	-	76.91	0.67	80.52	0.73
FIL	1	O	**82.90**	0.60	**84.68**	0.50
FIL	1	X	75.70	1.04	80.33	0.60
FIL	2	O	80.49	0.63	84.01	0.68

Table 5. Multi-class classification accuracies (ACC) on CIFAR-10 in dynamic database environments with various influencing or communication rounds (IR). Specifically, the numbers of rounds were 20 and 40, and the number of local training epochs was 10 for all experiments. The number of participants was 10. The p-value of accuracies between FIL and FL methods was 0.006.

HECTOR2021(PET)						
	# of Influencer	Q	Precision	Recall	Dice Score	SD
SOLO	-	-	62.21	22.38	23.70	12.10
FedAvg	-	-	74.71	19.62	21.41	10.94
FedProx	-	-	68.81	21.28	21.15	10.34
FIL	1	O	**80.45**	**61.55**	**63.93**	4.73
FIL	1	X	68.76	25.84	28.57	7.12
FIL	2	O	79.03	59.26	61.29	**2.06**

Table 6. Segmentation results for the PET dataset. The number of local training epochs was 10, and the number of communication and influencing rounds was 30. The number of participants was 5. The p-value for the Dice scores between FIL and FL methods was 0.001.

In addition, we defined the model as θ_k^l and weight of the θ_k^l as ω_k^l, where k and l indicate the indices of the participants and local epoch, respectively. Furthermore, f denotes a deep-learning network, with x and y representing the input data and labels, respectively. Additionally, p and q denote the logits derived from the shared datasets, respectively.

Federated influencer learning

In FIL, each participant has a private dataset, a shared dataset, and a test dataset. The private datasets are stored independently in local databases and are not shared. These datasets are used for local training. As training progresses, the number of private datasets may randomly increase. The shared dataset is common to all

BraTS2021						
	# of Influencer	Q	Precision	Recall	Dice Score	SD
SOLO	-	-	64.91	51.40	57.17	0.78
FedAvg	-	-	67.69	54.04	59.87	0.19
FedProx	-	-	59.15	57.06	57.89	**0.01**
FIL	1	O	68.93	**57.22**	**62.37**	0.27
FIL	1	X	67.46	54.04	59.85	0.10
FIL	2	O	**70.53**	53.81	60.87	0.31

Table 7. Segmentation results for the MRI dataset. The number of local training epochs was 10, and the number of communication and influencing rounds was 20. The number of participants was 5. The p-value for the Dice scores between FIL and FL methods was 0.158.

participants and is used to evaluate their performance in selecting an influencer. Additionally, this dataset can be used for local training. Finally, all participants use the same test dataset to measure their performance.

FIL consists of four stages: local training, qualification, screening, and influencing. During the local training stage, all participants independently train their models using their respective private datasets over multiple epochs, employing either cross-entropy (CE) loss or binary cross-entropy (BCE) loss. For segmentation tasks, we utilize Dice loss combined with CE or BCE loss.

$$\mathscr{L}_{CE} = -\sum_{i=1}^{N} y_i * log(f(x_i)) \tag{2}$$

$$\mathscr{L}_{BCE} = -[\sum_{i=1}^{P_k} y_i * log(f(x_i)) + \sum_{i=1}^{N_k} (1 - y_i) * log(1 - f(x_i))] \tag{3}$$

$$\mathscr{L}_{Dice} = \sum_{i=1}^{N} [1 - \frac{2 * \sum_{h=1}^{H} \sum_{w=1}^{W} p_{iwh}^{true} * p_{iwh}^{pred}}{\sum_{h=1}^{H} \sum_{w=1}^{W} p_{iwh}^{true} + \sum_{h=1}^{H} \sum_{w=1}^{W} p_{iwh}^{pred} + \varepsilon}] \tag{4}$$

N is total number of data. x is the prediction and y is the label. p^{true} and p^{pred} indicates predicted and true segmentation maps. W and H is the width and height of an image. ε is added to the denominator to prevent it from being zero. During the qualification step, participants assess their models using the shared dataset and submit their qualification scores to the administrator. The administrator then selects the influencer with the highest qualification score, and this role can change in each round. The remaining participants become followers and learn from the influencer. This approach makes FIL particularly effective in dynamically changing databases, as it allows for the selection of a powerful influencer while participant performances evolve with the increasing data.

$$S = argmax(\mathscr{Q}(\theta_1^t), \mathscr{Q}(\theta_2^t), \ldots, \mathscr{Q}(\theta_M^t)) \tag{5}$$

Q represents the qualification function. This procedure is identical to the ordinary test stage. θ is model parameter and M is the total number of clients. t is round index. The index of the selected participant as influencer is S.

In the screening stage, the integrity of the transmitted logits is verified. To enhance security, both the administrator and followers check the accuracy of the influencer's logits. Specifically, after the influencer sends the logits, derived from the shared dataset, to the administrator and followers, the administrator calculates the qualification score and forwards the logits to the followers if no issues are detected. Followers then calculate their own qualification scores to verify the integrity of the logits received from the administrator. If no flaws are found, the followers proceed to the influencing stage.

During the influencing stage, followers compare their logits with those of the influencers. For classification tasks, we use Kullback-Leibler (KL) divergence to measure this difference.

$$\mathscr{L}_{KLD}(p^s, p^k) = \sum_{i=1}^{L} p_i^s log \frac{p_i^k}{p_i^s} \tag{6}$$

$\mathscr{L}_{KLD}$ denotes the loss value using the KL divergence in the discrete probability variable. p_i denotes a logit vector before passing the fianl activation function with the length of the number of classes and i corresponds to the batch index. L is the length of the shared dataset. s indicates the index of the participant which was selected as an influencer and k is the index of the participants.

To reduce computation overhead, the local training and influencing stages can be executed in parallel. It is because if participants cannot finish local learning synchronously, waiting for the slowest participant can be time-consuming. As a result, the overall loss function can be expressed as follows:

$$\mathscr{L}_k = \mathscr{L}_{CE} + \sum_{i=1}^{L} \mathscr{L}_{KLD}(softmax(p_i^s/\tau), softmax(p_i^k/\tau)) * \alpha * \tau * \tau \tag{7}$$

$$\mathscr{L}_k = \mathscr{L}_{BCE} + [\sum_{i=1}^{L} \mathscr{L}_{KLD}(log(\sigma(p_i^s)), \sigma(q_i^k)) + \sum_{i=1}^{L} \mathscr{L}_{KLD}(log(1-\sigma(p_i^s)), 1-\sigma(q_i^k))] * \alpha \tag{8}$$

α, τ is the hyperparameters and σ is the sigmoid function.

In segmentation, we leverage the $L1$ distance instead of KL divergence to reduce the computation cost. We calculate $L1$ loss value between the output maps of the influencer and the followers in the segmentation task. In segmentation task, p is the pixel value of the segmentation result. H and W are the height and width of the segmentation result.

$$\mathscr{L}_k = \mathscr{L}_{CE} + \mathscr{L}_{Dice} + \sum_{i=1}^{L} \sum_{h=1}^{H} \sum_{w=1}^{W} |p_{iwh}^s - p_{iwh}^k| \tag{9}$$

$$\mathscr{L}_k = \mathscr{L}_{BCE} + \mathscr{L}_{Dice} + \sum_{i=1}^{L} \sum_{h=1}^{H} \sum_{w=1}^{W} |p_{iwh}^s - p_{iwh}^k| \tag{10}$$

Therefore, we updated the local model θ_k^t as follows:

$$\theta_k^t = \theta_k^{t-1} - \eta * \frac{d}{dw} \mathscr{L}_k \tag{11}$$

η is learning rate. Figure 1 shows overall procedure of FIL.

The usage of shared dataset

In FL, distributed local nodes maintain private training datasets that are not shared among participants. In FIL, we propose leveraging a shared dataset in addition to the independent private data. This shared dataset serves as a benchmark for comparing participants' performances during the qualification stage, helping to select the influencer. It can also be used for local training to ensure minimal performance.

The shared dataset can be composed from public data or donated private data from the participants. Since data from multiple centers are aggregated, the distribution of the shared dataset varies from each participant's local dataset. This diversity allows local models to achieve better generalization by leveraging their own personalized data.

After the qualification stage, resulting logits from the shared dataset are saved. After that, the logits from the selected influencer are transmitted to the administrator then remaining followers. After the screening stage, followers distill knowledge from the influencer into their local models.

Utilizing a shared dataset eliminates the need for exchanging model parameters or gradients, which enhances the security of the FIL approach against malicious attacks. Since all participants use a common dataset, they can re-evaluate the influencer's qualification score, thus preventing manipulation of the logits by malicious influencers or administrators.

Furthermore, incorporating the shared dataset during the local training stage establishes a minimum performance guarantee for the participants. While there are concerns that reliance on the shared dataset might lead to overfitting and artificially high qualification scores, our experiments revealed that participants with limited private data typically received lower qualification scores.

Considerations of practical applications

Administrator

Instead of using a central server, we propose employing an administrator who selects an influencer based on the submitted scores from participants and distributes the influencer's logits to all followers. While the administrator could be compromised, followers can verify the integrity of the logits they receive by recalculating the score of the influencer's logits. By using the administrator with relatively higher capacity than the participants, we can solve the network bottleneck problem when distributing logits to all followers.

Dealing with communication overhead

Since FIL involves logit transactions, it may incur significant communication overhead, especially with a large shared dataset. To mitigate this, we propose a cascaded influence strategy. In this approach, the shared dataset is divided into subsets. The influencer sends these subsets of logits to the followers individually, who then perform the distillation step as they receive each subset. Although a cascaded influence might extend the process slightly, it does not significantly increase the overall training time because the influencing stage can run in parallel with local training.

The robustness of FIL

FIL exhibits a noteworthy resilience against a plethora of adversarial incursions, attributable largely to its distinctive architectural composition. The FIL framework demonstrates notable resilience against various adversarial attacks due to its unique architectural design. By eliminating the need for direct model parameter exchanges, FIL is protected against a range of threats, including adversarial attacks, model inversion, model extraction, Byzantine attacks, man-in-the-middle attacks, and both data and model poisoning. These attacks typically rely on access to model parameters. Moreover, the ability to change the influencer each round prevents a single point of failure, unlike in traditional FL systems. However, using a public dataset as a common dataset among participants does pose a risk of private information leakage through logit exchanges. To mitigate this risk, logits can be encrypted using various cryptographic techniques. Although exchanging logits does present some security concerns, especially since model parameters contain more information, leveraging logit exchanges still provides distinct advantages.

Furthermore, the FIL methodology includes a qualification process that acts as a safeguard against disruptions that could hinder participants' progress towards achieving personalized global minima. This issue is particularly prevalent in FL or online student learning environments. FIL mitigates the risk associated with integrating new participants by excluding those with low qualification scores from becoming influencers. Although using a shared dataset can present challenges, encrypting the dataset or having the administrator calculate qualification scores can address these issues.

Additionally, strategically using a shared dataset during the local training phase provides significant advantages, especially for smaller medical institutions facing data scarcity. Such constraints not only limit the scope of their model's applicability but also predispose these models to severe overfitting. Incorporating shared datasets into local training helps alleviate these issues by providing access to a broader and more diverse data pool. This approach not only addresses data scarcity and model overfitting but also enhances model generalizability by aligning models across different nodes and establishing minimum performance guarantee. Additionally, while the utilization of shared datasets for pre-training, as discussed in[27], has been shown to improve performance, it is crucial to consider the potential for catastrophic forgetting in real-world applications. Regularly incorporating shared datasets in each local training iteration can help prevent this phenomenon, further supporting their role in enhancing model robustness and generalizability.

Experiments and results

We conducted experiments on the classification and segmentation tasks. For classification, we used large public X-ray (NIH ChestX-ray14 (NIH CXR 14), CheXpert[43]) and natural image datasets (CIFAR-10[44]). For segmentation, we used a brain MRI (BraTS2021[45]), head and neck cancer PET dataset (HECTOR2021[46])). Here, IR denotes the number of rounds of influencing and Q represents whether the shared dataset is used in all participants' local training. We considered that the influencing round was equal to the communication round in FL to quantitatively compare the performances of FIL and FL. SOLO means the standalone in which only local training is executed without model aggregation or knowledge distillation. Regarding the metrics, SD means the standard deviation of all participants' accuracy or AUC values. All the metrics are averaged results of all participants. In the segmentation, this means the standard deviation of Dice score. It represents the consistency and fairness among the participants. For the medical image classification, we used Area Under the Receiver Operating Characteristic Curve (AUC). It measures how clearly models can classify images, with 1 indicating perfect discrimination and 0.5 indicating no discrimination. For segmentation, we assessed precision, recall, and Dice scores: precision measures the true positive rate among all cases classified as true, recall measures the true positive rate among all cases labeled as true, and the Dice score combines precision and recall to gauge the overlap between predicted and true values. To calculate p-values, we compared the two best results from the FIL experiments with the results of all FL methods.

Dynamic database

We implemented a dynamic database to test FIL in a setting closely emulating hospitals' dynamically accumulating database. Given that data are expensive and scarce in the medical domain, they accumulate over time. This dynamic database is different from the domain-incremental learning[47] in that accumulated data are not removed.

For the experiment, for each participant in each round, the amount of data added followed a Dirichlet distribution with the $\psi = 0.5$ which determines the skewness of the distribution. Low ψ value means the amounts of data added each round are highly heterogeneous.

Classification

We compared the performance of the FIL with several FL algorithms[7,19–21,24]. We set the hyperparameters following the original papers. If the hyperparameters were not provided, optimal values were assigned through experimental trials. The learning rate, batch size, the number of influencing epoch, and weight decay were set to 0.01, 32, 1, and 0.0001, respectively; we used RedNet56[48]. An SGD optimizer was used while setting the momentum value to 0.9. We set the α to 0.99 and τ to 1.5 for all FIL experiments.

Medical X-ray dataset

We used two different chest X-ray datasets (NIH ChestXray14, CheXpert). The NIH CXR 14 dataset included 112,120 images, which were divided into training (69, 219), common (17,305), and test (25,596) datasets. The size of the images was $1024 \times 1024 \times 3$; we resized them to $150 \times 150 \times 3$ pixels. All X-ray images were in the anterior view. The CheXpert dataset comprised 223,415, 235, and 669 training, validation, and test images, respectively. Among the 223,415 training data, we randomly sampled 69,219 and 17,305 for the training and

shared datasets, respectively. The images in this dataset are either anterior or lateral and were normalized during preprocessing. The image was resized to match the NIH CXR 14 size. The number of clients (participants) is set to five for all classification experiments which used the medical X-ray dataset. Q represents whether the shared dataset is used during the local training. Table 4 compares the results of the FIL and FL algorithms. The FIL recorded the highest performance and fairness, which were assessed using the standard deviation of the AUC. When employing two influencers, FIL exhibited superior fairness compared to the other methods.

Natural image dataset
We use CIFAR-10 to test the generalizability of the FIL. CIFAR-10 consisted of 10 classes of 32 × 32 RGB images. For the pre-processing, we normalized data. Each dataset had 50,000 training and 10,000 test images. We divided 40,000, 10,000, and 10,000 samples into training, common, and test datasets, respectively. The number of clients (participants) was set to ten for all classification experiments using natural image dataset. Table 5 shows that FIL outperformed other FL methods in average AUC values and descent performance of standard deviation.

Segmentation

We used BraTS2021-the brain tumor MRI, HECTOR2021-head and neck cancer dataset to demonstrate the applicability of FIL to medical segmentation tasks. For the segmentation experiments, we leveraged 2D vanilla U-Net architecture. Learning rate was set to 0.01, batch size was 32, the number of influencing epoch to 1. For the optimizer, we selected SGD with momentum value 0.9. The number of participants is 5 for all experiments.

PET dataset
HECTOR2021 is comprised with PET and CT data. Among them, we used PET for the experiments. This dataset is designed to segment tumor in the head and neck cancer 3D PET images. It consists of five centers and each sample has its corresponding center label. We also performed slice-wise training. Slice-wise training was conducted by slicing 3D data into 2D data in the axial direction. For the shared and test datasets, we random sampled ten percent of slices from each institution. We resized the image to 150 x 150 for the computation efficiency. Regarding the pre-processing, we applied simple linear registration and standardization. We did not used bounding-boxes. As a result, each institution has 4,073, 1,858, 4,648, 8,305, 14,093 slices for the training and 4,122 for the common dataset, and 4,120 for the test. We set the number of clients (participants) to five for all segmentation experiments. Table 6 shows that the FIL demonstrated superior results to the standalone and several FL methods in various metrics. In particular, in the HECTOR2021 dataset, This improvement is attributed to the dataset's unique experimental conditions, where each participant's training data came from a single institution. The multi-institutional shared dataset helped mitigate the domain shift between training and test data, leading to substantial performance enhancements.

MRI dataset
The BraTS2021 dataset consisted of 5,004 brain MRIs from 1,251 patients. We included 'Flair,' 'T1,' 'T2,' and 'T1ce' types of MRI images. Each image measured 240 × 240 × 155. For the pre-processing, we applied normalization. Two-dimensional slice-wise training was performed. We randomly sampled 62,000, 38,750, and 39,060 slices for training, common, and testing, respectively. Table 7 shows that the FIL demonstrated superior results to the standalone and several FL methods in various metrics.

Efficiency of FIL

The communication process must be carefully managed when implementing FL in real-world settings. In FL, training is paused while waiting for the merged model from the server to aggregate it. Additionally, clients may face limitations in participating in the global communication step due to time zone differences or geographic constraints, which can lead to a biased global model. Furthermore, the associated communication overhead can be substantial. In contrast, FIL does not require halting local training for knowledge distillation; local learning and its effects occur simultaneously. Followers also have the flexibility to perform the influencing stage at a later time, even after receiving logits. Consequently, FIL is more efficient than FL in practical scenarios.

Discussion

Regarding security, while FIL is capable of mitigating many types of attacks, it is vulnerable to the manipulation of logits transmitted from the influencer to the followers. This vulnerability presents a challenge, although follower nodes can scrutinize the received logits and opt to reject them if anomalies are detected, which can hinder performance improvement. Therefore, it is crucial to address potential attacks targeting logit manipulation within the FIL paradigm. Future studies should aim to strengthen the screening stage and enhance FIL's resilience against sophisticated adversarial manipulations, thereby protecting the collaborative learning process and its outcomes. Additionally, future research should focus on exploring and developing various fine-tuning mechanisms within the FIL paradigm to substantially enhance performance.

Limitation

In implementing FIL, it is crucial to establish a shared dataset among participants. However, this can pose challenges under certain conditions. For example, participants with limited computational resources, such as restricted GPU capabilities, may face significant increases in training time due to the shared dataset, potentially hindering their ability to participate in the qualification stage. Additionally, the lack of publicly available datasets suitable for shared use may require participants to contribute their own data, which can be problematic

as participants might be reluctant to share their data due to privacy concerns or other issues. Therefore, the applicability of FIL methodologies depends on overcoming these challenges.

Regarding security, while FIL can mitigate various types of attacks, it remains vulnerable to logit manipulation during transmission from the influencer to the followers. To address this, we assume that qualification scores are calculated on the administrator's side. In this scenario, since participants do not have access to the correct answers, their attempts at manipulation would be ineffective. Future research should focus on strengthening methodologies to prevent logit manipulation, thereby protecting the collaborative learning process and its outcomes.

In our experiment, we did not assess the performance of contemporary transformers. Incorporating transformers could enhance overall performance, especially with large datasets.

Conclusion

Privacy-preserving decentralized training is crucial for practical deep learning applications in healthcare. Despite advancements in FL, challenges remain, particularly concerning attacks and the dynamic accumulation of databases that mirror real-world medical environments. Our study introduces FIL, a method designed for secure and efficient training in hospitals with realistic database settings that offer clinical value. FIL enhances model performance through collaborative knowledge distillation by incorporating local training, qualification, screening, and influencing stages. A key advantage of FIL is its robustness against various attacks due to the lack of model aggregation and a central server. Logit transactions, rather than model parameters, substantially reduce the risk of privacy breaches-an essential consideration in the medical field where stringent security measures are required. Our experiments demonstrated FIL's superior performance across various tasks and domains, highlighting its potential for practical applications. Future research should address the limitations of FIL related to shared datasets and logit transactions and explore its applicability to diverse tasks and datasets. Additionally, when a suitable shared dataset is unavailable within the same domain as local datasets, leveraging a cross-domain shared dataset without compromising performance could be a promising approach.

Data availability statement

NIH ChestXray14: https://www.kaggle.com/datasets/nih-chest-xrays/data. CheXpert: https://www.kaggle.com/datasets/ashery/chexpert. CIFAR-10: https://cs.toronto.edu/~kriz/cifar.html. HECTOR : https://hecktor.grand-challenge.org/ (our data is 2021 version) BraTS2021: https://www.kaggle.com/datasets/dschettler8845/brats-2021-task1.

Received: 1 July 2024; Accepted: 22 September 2024
Published online: 30 September 2024

References

1. Krupa, A. J. D., Dhanalakshmi, S., Lai, K. W., Tan, Y. & Wu, X. An iomt enabled deep learning framework for automatic detection of fetal qrs: A solution to remote prenatal care. *In Journal of King Saud University-Computer and Information Sciences* **34**, 7200–7211 (2022) (**Elsevier**).
2. Achiam, J. et al. Gpt-4 technical report. In arXiv preprint arXiv:2303.08774 (2023).
3. Li, Q. et al. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. In *IEEE Transactions on Knowledge and Data Engineering* (IEEE, 2021).
4. Choi, S. J., Johnson, M. E. & Lee, J. An event study of data breaches and hospital it spending. *In Health Policy and Technology* **9**, 372–378 (2020) (**Elsevier**).
5. Ali, M., Naeem, F., Tariq, M. & Kaddoum, G. Federated learning for privacy preservation in smart healthcare systems: A comprehensive survey. *IEEE journal of biomedical and health informatics* **27**, 778–789 (2022).
6. El Ouazzani, Z., El Bakkali, H. & Sadki, S. Privacy preserving in digital health: main issues, technologies, and solutions. In *Research Anthology on Privatizing and Securing Data*, 1503–1526 (IGI Global, 2021).
7. McMahan, B., Moore, E., Ramage, D., Hampson, S. & y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 1273–1282 (PMLR, 2017).
8. Wu, X. et al. A novel centralized federated deep fuzzy neural network with multi-objectives neural architecture search for epistatic detection. In *IEEE Transactions on Fuzzy Systems* (IEEE, 2024).
9. Rieke, N. et al. The future of digital health with federated learning. *In NPJ digital medicine* **3**, 119 (2020) (**Nature Publishing Group UK London**).
10. Tan, A. Z., Yu, H., Cui, L. & Yang, Q. Towards personalized federated learning. In *IEEE Transactions on Neural Networks and Learning Systems* (IEEE, 2022).
11. Kairouz, P. et al. Advances and open problems in federated learning. *In Foundations and Trends® in Machine Learning* **14**, 1–210 (2021) (**Now Publishers, Inc.**).
12. Liu, B., Lv, N., Guo, Y. & Li, Y. Recent advances on federated learning: A systematic survey. *Neurocomputing* 128019 (2024).
13. Zhao, Y. et al. Federated learning with non-iid data. In arXiv preprint arXiv:1806.00582 (2018).
14. Qi, P. et al. Model aggregation techniques in federated learning: A comprehensive survey. In *Future Generation Computer Systems* (Elsevier, 2023).
15. Lyu, L., Yu, H. & Yang, Q. Threats to federated learning: A survey. In arXiv preprint arXiv:2003.02133 (2020).
16. Tolpegin, V., Truex, S., Gursoy, M. E. & Liu, L. Data poisoning attacks against federated learning systems. In *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25*, 480–501 (Springer, 2020).
17. Mammen, P. M. Federated learning: Opportunities and challenges. In arXiv preprint arXiv:2101.05428 (2021).
18. Wu, X., Wei, Y., Jiang, T., Wang, Y. & Jiang, S. A micro-aggregation algorithm based on density partition method for anonymizing biomedical data. *In Current Bioinformatics* **14**, 667–675 (2019) (**Bentham Science Publishers**).
19. Li, T. et al. Federated optimization in heterogeneous networks. *In Proceedings of Machine Learning and Systems* **2**, 429–450 (2020).
20. Li, Q., He, B. & Song, D. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10713–10722 (2021).
21. Mendieta, M. et al. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8397–8406 (2022).

22. Karimireddy, S. P. et al. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference On Machine Learning*, 5132–5143 (PMLR, 2020).
23. Li, T., Hu, S., Beirami, A. & Smith, V. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, 6357–6368 (PMLR, 2021).
24. Zhang, J. et al. Federated learning with label distribution skew via logits calibration. In *International Conference on Machine Learning*, 26311–26329 (PMLR, 2022).
25. Li, Z., Zhang, J., Liu, L. & Liu, J. Auditing privacy defenses in federated learning via generative gradient leakage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10132–10142 (2022).
26. Li, J. et al. Ressfl: A resistance transfer framework for defending model inversion attack in split federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10194–10202 (2022).
27. Li, D. & Wang, J. Fedmd: Heterogenous federated learning via model distillation. In arXiv preprint arXiv:1910.03581 (2019).
28. Lin, T., Kong, L., Stich, S. U. & Jaggi, M. Ensemble distillation for robust model fusion in federated learning. *In Advances in Neural Information Processing Systems* **33**, 2351–2363 (2020).
29. Shen, T. et al. Federated mutual learning. In arXiv preprint arXiv:2006.16765 (2020).
30. Gong, X. et al. Ensemble attention distillation for privacy-preserving federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15076–15086 (2021).
31. Zhang, L., Shen, L., Ding, L., Tao, D. & Duan, L.-Y. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10174–10183 (2022).
32. Wang, H. et al. Dafkd: Domain-aware federated knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20412–20421 (2023).
33. Wang, L. & Yoon, K.-J. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. In *IEEE Transactions On Pattern Analysis and Machine Intelligence*, vol. 44, 3048–3068 (IEEE, 2021).
34. Gou, J., Yu, B., Maybank, S. J. & Tao, D. Knowledge distillation: A survey. *In International Booktitle of Computer Vision* **129**, 1789–1819 (2021) (**Springer**).
35. Li, T., Sanjabi, M., Beirami, A. & Smith, V. Fair resource allocation in federated learning. In arXiv preprint arXiv:1905.10497 (2019).
36. Fang, X. & Ye, M. Robust federated learning with noisy and heterogeneous clients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10072–10081 (2022).
37. Ma, X., Zhang, J., Guo, S. & Xu, W. Layer-wised model aggregation for personalized federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10092–10101 (2022).
38. Duan, J.-h., Li, W., Zou, D., Li, R. & Lu, S. Federated learning with data-agnostic distribution fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8074–8083 (2023).
39. Zhang, Y., Xiang, T., Hospedales, T. M. & Lu, H. Deep mutual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4320–4328 (2018).
40. Guo, Q. et al. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11020–11029 (2020).
41. Wu, G. & Gong, S. Peer collaborative learning for online knowledge distillation. *In Proceedings of the AAAI Conference on Artificial Intelligence* **35**, 10302–10310 (2021).
42. Li, S. et al. Distilling a powerful student model via online knowledge distillation. In *IEEE Transactions on Neural Networks and Learning Systems* (IEEE, 2022).
43. Irvin, J. et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. *In Proceedings of the AAAI Conference on Artificial Intelligence* **33**, 590–597 (2019).
44. Krizhevsky, A. et al. *Learning multiple layers of features from tiny images* (ON, Canada, Toronto, 2009).
45. Menze, B. H. et al. The multimodal brain tumor image segmentation benchmark (brats). *In IEEE Transactions on Medical Imaging* **34**, 1993–2024 (2014) (**IEEE**).
46. Andrearczyk, V. et al. Overview of the hecktor challenge at miccai 2021: automatic head and neck tumor segmentation and outcome prediction in pet/ct images. In *3D head and neck tumor segmentation in PET/CT challenge*, 1–37 (Springer, 2021).
47. Van de Ven, G. M. & Tolias, A. S. Three scenarios for continual learning. In arXiv preprint arXiv:1904.07734 (2019).
48. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778 (2016).

Acknowledgements

This work was supported by K-Brain Project (No. RS-2023-00264160) and the Basic Research Project (NO. RS-2024-00354123) of the National Research Foundation (NRF) funded by Ministry of Science and ICT, the Korea Medical Device Development Fund grant funded by the Korea government (the Ministry of Science and ICT, the Ministry of Trade, Industry and Energy, the Ministry of Health & Welfare, the Ministry of Food and Drug Safety) (Project Number: 1711137868, RS-2020-KD000006), and the Institute of Information & communications Technology Planning & Evaluation (IITP) grant [NO.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)].

Author contributions

H Chung devised methods, conducted experiments, and wrote manuscript draft, and JS Lee supervised study and edited manuscript.

Declaration

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-024-73863-1.

Correspondence and requests for materials should be addressed to J.S.L.