2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 979-8-3503-8622-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/BIBM62325.2024.10822329

Federated Learning for COVID-19 Detection: Optimized Ensemble Weighting and Knowledge Distillation

Richard Annan
Department of Computer Science
North Carolina A&T State University
Greensboro, NC, USA
rkannan@aggies.ncat.edu

Letu Qingge*
Department of Computer Science
North Carolina A&T State University
Greensboro, NC, USA
lqingge@ncat.edu

***Abstract*—The COVID-19 pandemic has underscored the need for effective diagnostic tools, particularly in resource-limited settings. While RT-PCR and CT scans are standard, their limitations drive the need for advanced techniques. This study leverages Convolutional Neural Networks, Knowledge Distillation, Ensemble Learning, and Federated Learning to develop robust, privacy-preserving models for COVID-19 detection from CT scans. We propose two federated learning strategies to simplify deep learning models for use in clinical environments with limited computational resources. The first strategy uses knowledge distillation from a complex model to a simplified model shared across a federated network. The second allows each hospital to distill knowledge to its simplified model, later combined into a global model via ensemble learning. Our methods, AFKD and IKDEFL, outperform traditional federated learning approaches such as FedAvg and FedAdam. AFKD, paired with the COVID-CNN model, achieves 91%-95% accuracy on IID (Independent and Identically Distributed) datasets and 70%-89% on non-IID datasets. IKDEFL further improves performance, with 92%-95% accuracy on IID datasets and 76%-88% on non-IID datasets. These approaches provide promising solutions for enhancing COVID-19 detection in federated learning.**

***Index Terms*—COVID-19 detection, Federated Learning, Knowledge Distillation, Ensemble Learning, Deep Learning**

I. Introduction

The COVID-19 pandemic has significantly impacted people's lives, with its primary affect on the respiratory system. The virus is generally spread through airborne particles and physical contact involving saliva or mucus from an infected person. Common symptoms include fever, cough, and shortness of breath, posing severe risks to individuals with weakened immune systems or pre-existing health conditions [1], [2]. Rapid diagnosis is crucial for effective treatment and often relies on the Reverse Transcription-Polymerase Chain Reaction (RT-PCR) assay, which typically requires a patient sample, such as a nasal swab. However, despite the widespread use of RT-PCR assays, their relatively high false-negative rate remains a concern [3]. To address this limitation, medical imaging techniques, such as X-rays, lung ultrasound and Computer Tomography(CT) scans reveal signs of infection, like lung inflammation, that may not be detected by RT-PCR [2], [4]. These imaging methods support a more accurate diagnosis and aid in timely treatment. Chest X-rays provide a fast and cost-effective method for evaluating lung involvement in COVID-19 patients. However, due to their limitations in sensitivity and interpretation variability, CT scans are often preferred to provide more detailed visualizations of lung abnormalities [4]. Additionally, lung ultrasound provides a safe and portable diagnostic alternative without radiation exposure. However, it is less comprehensive compared to CT scans, which is why CT scans remain a popular choice for COVID-19 detection [2]. Recent advancements in Artificial Intelligence (AI), particularly deep learning, have shown promise in diagnosing COVID-19 from CT scan images. These technologies enhance classification accuracy and provide valuable insights that assist clinicians in making treatment decisions.

Convolutional Neural Networks (CNNs) have proven effective in detecting COVID-19 from CT scans, but they struggle to generalize beyond the datasets on which they are trained [5]. This limitation is exacerbated by privacy concerns and data-sharing restrictions, which hinder the collection of large, diverse datasets necessary for better generalization. Federated Learning (FL) enables collaborative global model training while preserving data privacy, making it valuable for COVID-19 detection [6]. However, FL frameworks often incorporate CNNs into complex architectures like Generative Adversarial Networks (GANs) and 'VGGish' networks [7]–[9], which are unsuitable for resource-limited settings. Knowledge distillation techniques, introduced by Hinton *et al.* [10], addressed this by transferring knowledge from complex models to simpler ones, reducing complexity and computational demands. This makes the models more efficient for deployment in FL frameworks in clinical environments. For instance, Qin *et al.* [11] improved the deployability of models like RA-UNet and PSPNet in clinical settings by using knowledge distillation to reduce their size and computational requirements

In this study, we explore the integration of several advanced techniques, including CNNs, Ensemble Learning (EL),

*This work is supported by the U.S. National Science Foundation under awards 2434487 and 2200138. Corresponding Author: Dr. Letu Qingge.

Knowledge Distillation (KD), and Federated Learning (FL) for COVID-19 detection. Our goal is to develop a robust, privacy-preserving model for detecting COVID-19 from CT scans. This model is designed to be applicable in resource-constrained medical settings. To the best of our knowledge, our research is pioneering in integrating these advanced techniques specifically for COVID-19 detection from CT scans. We focus on training complex models on individual hospital datasets and examine two FL strategies. One involves KD from a selected complex model to a simplified model across the network, and the other combines independently trained simplified models using EL.

The rest of the paper is organized as follows: Section II offers an overview of CNNs, Ensemble Learning, Knowledge Distillation, and Federated Learning. Section III details the methodology and introduces our proposed federated optimization frameworks, which incorporate ensemble learning and knowledge distillation techniques using CNNs as backbone networks. Section IV focuses on the experiments, presenting our datasets, evaluation metrics and a discussion of the results. Lastly, Section V concludes the paper.

II. Background

This section provides the foundational concepts and techniques that underpin the study of federated learning for COVID-19 CT detection using optimized ensemble weighting and knowledge distillation. We will explore the essential principles of CNNs, Ensemble Learning, Knowledge Distillation, and Federated Learning.

A. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are specialized deep learning architectures effective in processing grid-like data, such as images, by learning complex features. CNNs consist of key components: convolutional layers, pooling layers, and fully connected layers [12]. Convolutional layers apply learned filters to detect patterns in the input image, producing feature maps that help classify the image. This operation is expressed as [12], [13]:

$$Z_{i,j}^{k} = \sigma\left(\sum_{m}\sum_{n} X_{i+m,j+n} \cdot W_{m,n}^{k} + b^{k}\right) \tag{1}$$

where $Z_{i,j}^{k}$ is the output feature map, X is the input image, W^{k} are the filter weights, b^{k} is the bias, and σ is a non-linear activation function, typically ReLU. Pooling layers, such as max pooling, reduce feature map dimensionality by retaining only the most important information. Fully connected layers at the network's end map the extracted features to the output space, enabling effective image classification across various datasets.

In our study, we utilized a simple CNN as the student model (S) within our proposed framework, alongside several teacher models which were also CNNs utilizing the convolution operation expressed in Eq.1. The student model features an input layer for 200×200×1 images, a convolutional layer with 32 filters of size 3×3 and ReLU activation, followed by a 2×2 max-pooling layer, and a dense layer with 2 units for classification. For teacher models, we employed three architectures: *COVID-CNN* [5], tailored for COVID-19 detection with test accuracy of 93% on our datasets; *Deep-COVID* [14], a deep network for lung scan analysis, achieving 92.23% accuracy; and *COVID-VGG16*, which combines COVID-CNN's fully connected layer with VGG16 for feature extraction, with an approximate accuracy of 87%. These models collaboratively enhance the learning process, improving the robustness and accuracy of S.

B. Ensemble Learning with Adaptive Weighting Voting

As highlighted by Mohammed and Kora [15], EL combines multiple models to improve predictive performance, using techniques such as bagging, boosting and stacking. This approach, particularly in deep learning models, addresses high variance, enhancing their effectiveness for complex tasks like image classification. In EL, predictions from individual models are combined using methods such as *Max Voting*, *Averaging Voting*, and *Weighted Average Voting* [15]. *Max Voting* selects the class label with the most votes from integrating models, using hard voting: $y = \text{mode}[C_1(x), C_2(x), \ldots, C_n(x)]$ or soft voting: $y = \arg\max_i \sum_{j=1}^{n} w_j P_{ij}$ where w_j is the weight assigned to the j-th model (P) [16]. *Averaging Voting* predicts by averaging predictions from all models: $y = \arg\max_i \frac{1}{n}\sum_{j=1}^{m} w_{ij}$; while *Weighted Average Voting* assigns different importance to each model's prediction [17]:

$$y = \frac{\sum_{j=1}^{m} w_j x_j}{\sum_{j=1}^{m} w_j} \tag{2}$$

where w_j is the weight, m are the number of models, and x_j are predictions.

The Weighting method (*Weighted Average Voting*) assigns different weights to the models that are combined. By giving more importance to better-performing models, often determined by their performance on a validation dataset, this method improves overall prediction accuracy. According to Mao *et al.* in [18], the weighted votes can be made adaptive (trainable) by minimizing the error between weighted outputs and true values. This approach ensures that the weights sum to one ($\sum w_i = 1$) and lie within a specified range ($-1 < w_i < 1$), optimizing and balancing the ensemble model.

In our work, we utilized *Weighted Average Voting* as an adaptive, trainable layer in our ensemble model. This design allows the ensemble to be fine-tuned while optimizing the voting weights. We implemented three adaptive voting mechanisms within the ensemble: (1) making the soft-voting weights trainable, (2) employing multi-head attention with eight heads and a head size of 128, and (3) incorporating four transformer blocks, each with four heads and a head size of 128, to optimize and balance the ensemble weights. This approach enhances the ensemble's flexibility and performance by dynamically adjusting the contribution of each model based on their predictive strengths.

C. Knowledge Distillation

Deep Learning models are powerful, but often too large for deployment on resource-constrained devices. Knowledge Distillation (KD), introduced by Hinton *et al.* [10], addresses this by transferring knowledge from complex models to smaller ones, enabling efficient deployment while maintaining performance. In KD, a large model generates soft target distributions at a high temperature, which are used to train a smaller model. This process is mathematically described by the Kullback-Leibler divergence loss ((KD_{loss})) in Eq.4 to obtain the distillation loss $L_{distill}$ in Eq.3. The distillation loss between the teacher (t) and student model (s) is give by:

$$L_{\text{distill}}(\rho_s, \rho_t) = \text{KD}_{\text{loss}}\left(\sigma\left(\frac{\rho_t}{\tau}\right), \sigma\left(\frac{\rho_s}{\tau}\right)\right) \times \tau^2 \quad (3)$$

where ρ_s and ρ_{t_i} are logits from the student and i-th teacher models, respectively, and τ is a temperature scaling hyperparameter set to 10 in this study. The KD_{loss} between softened predictions $p_s(\rho_t/\tau)$ and $p_t(\rho_t/\tau)$ is computed as [19]:

$$\text{KD}_{\text{loss}}(p_t \parallel p_s) = \sum_j p_{t,j} \log\left(\frac{p_{t,j}}{p_{s,j}}\right) \quad (4)$$

where j indexes the classes. After training, the smaller model returns to using the standard temperature setting of [10].

In this study, the distillation loss ($L_{distill}$) is used to update student model (S) weights with the Adam optimizer in [20], calculating gradients $\nabla_{w_t} L_{\text{distill}}$. The optimizer updates weights w_t using first and second moment estimates m_t and v_t, with β_1 and β_2 as exponential decay rates. Subsequently, the student model's total loss L_{total}, combining cross-entropy (CE) loss $L_{\text{CE}} = -\sum_j y_j \log(p_{s,j})$ and $L_{distill}$, is optimized as $L_{\text{total}} = \alpha \cdot L_{\text{CE}} + (1-\alpha) \cdot L_{\text{distill}}$ where α balances these losses. Through repeated training, S can pick up intricate patterns from the teacher model (T) and adjust to its own tasks. It strikes a good balance between the teacher's insights and the S's task specific performance needs, helping to reduce inconsistencies and biases.

D. Federated Learning

Federated Learning (FL) enables collaboration among entities like hospitals without sharing sensitive data [21], [22]. Each client trains a model locally and sends updates to a central server, which aggregates them into a global model. This iterative process continues until convergence, allowing collaboration in contexts like COVID-19 detection from CT scans while maintaining patient privacy. Notable FL algorithms include FedAvg, which averages model weights from clients, and FedProx, which addresses data distribution differences [21]. Other algorithms, like FedSGD and FedMA, tackle specific FL challenges [22]. In this study, we used TensorFlow Federated to implement our FL frameworks incorporating Ensemble Learning (EL) and Knowledge Distillation (KD) techniques.

III. Methodology and Proposed Framework

In this section, we explain a method that combines EL with weighted voting, KD and FL to get the best results in identifying COVID-19 from CT scan images. The key challenge is to develop a model that leverages diverse datasets from multiple hospitals (clients) while ensuring that sensitive patient data remains local.

A. Problem Definition

In the healthcare sector, maintaining patient privacy is paramount while ensuring accurate and effective identification of diseases such as COVID-19 from CT scan images. Traditional centralized machine learning approaches require pooling data into a single repository, posing significant privacy risks. This necessitates a solution that allows collaborative model training without compromising data privacy.

B. Proposed Frameworks

Our approach involves training multiple complex models on different local datasets and then exploring two federated learning strategies. Let $\{M_1, M_2, \ldots, M_n\}$ be a set of n individual complex models at n clients with respective datasets (D). Let S be the simplified student model.

1) Appointive Federated Knowledge Distillation: The Appointive Federated Knowledge Distillation (AFKD) approach, summarized in Algorithm 1, enhances the student model (S) by leveraging the expertise of a selected teacher model (M_i) from participating hospitals. Initially, S is shared with all hospitals, and one teacher model (M_i) is chosen to guide the distillation process. Each hospital then locally trains S using the predictions from M_i, with distillation loss $L_{distill}$ calculated via Kullback-Leibler (KL) divergence [19]. This process refines S by minimizing $L_{distill}$ and the student's own classification loss (CE) to balance its learning. After training, updated parameters w_t^{k+1} from each hospital are aggregated on a central server into a global model (S_g), which is then redistributed. This iterative process continually improves S using collective insights while preserving data privacy.

2) Independent local Knowledge Distillation with post-Ensemble Federated Learning: The Independent Local Knowledge Distillation with post-Ensemble Federated Learning (IKDEFL) algorithm initializes local student models $S_1, S_2, \ldots, S_n$ at each hospital, where knowledge is distilled from complex models M_k using respective datasets D_k (see Algorithm 2). These local models are then combined into an ensemble model S_{ensemble} through weighted voting, with the weights adjusted to form a unified output. S_{ensemble} is further trained on local datasets, with updates sent to a central server to form the global ensemble model (S^*_{ensemble}). This iterative process continues over several rounds, with S^*_{ensemble} being refined and redistributed after each round, ultimately producing a well-optimized global model that benefits from the diversity of the participating clients.

IV. Experimental Study

To verify our method's effectiveness, we carried out an experimental study, which is described in this section. We begin by explaining the selected datasets and performance metrics. Then, we delve into the experimental results. Finally,

Algorithm 1 Appointive Federated Knowledge Distillation (AFKD)

Require: A set of complex models $\{M_1, M_2, \ldots, M_n\}$ at n clients, with datasets $D = \{D_1, D_2, \ldots, D_n\}$, simplified student model S, E: Number of local epochs, η_s: Student Learning rate,η_t: Server Learning rate, T: Temperature parameter, α: Weight for student loss ($0 \leq \alpha \leq 1$)
Ensure: Fine-tuned student model S

Initialize student model S
Elect one complex model M_i from the n clients
Distribute the student model S parameters to all n clients
function LocalTraining(S_w, D_k, M_i)
    Initialize student model S with S_w
    $S_w \leftarrow$ Distill(M_i, S, D_k)
    Update S with S_w
    return S_w (updated parameters of S)
end function
function Distill(M_i, S, D_k)
    for each local epoch i from 1 to E **do**
        for each batch b in D_k **do**
            Compute predictions of M_i and S on batch b as ρ_t and ρ_s respectively
            Compute the distillation loss $L_{\text{distill}} = KD_{loss}(\varsigma(\rho_t/\tau), \varsigma(\rho_s/\tau)) \cdot T^2$
            Compute gradients: $\nabla_{w_t^k} L_{\text{distill}}$
            $m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla_{w_t} L_{\text{distill}}$
            $v_t = \beta_2 v_{t-1} + (1-\beta_2)(\nabla_{w_t} L_{\text{distill}})^2$
            $w_t^{k+1} = w_t - \eta \frac{m_t}{\sqrt{v_t}+\epsilon}$
            Compute the student loss $L_{\text{CE}} = S_{loss}(y, p_s) = -\sum_j y_j \log(p_s)$
            Compute total loss $L_{\text{total}} = \alpha \cdot L_{\text{CE}} + (1-\alpha) \cdot L_{\text{distill}}$
            Compute gradients: $\nabla_{w_t^k} L_{\text{total}}$
            $m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla_{w_t} L_{\text{total}}$
            $v_t = \beta_2 v_{t-1} + (1-\beta_2)(\nabla_{w_t} L_{\text{total}})^2$
            $w_t^{k+1} = w_t - \eta \frac{m_t}{\sqrt{v_t}+\epsilon}$
        end for
    end for
    return Updated (w_t^{k+1}) parameter of S
end function
function Aggregation(local_updates)
    Send local updates to central server
    Aggregate updates to form a global model
    return Global model parameters
end function
function FederatedLearning(S, M_i)
    $S_i \leftarrow$ Initialize parameters of S
    for each round $t = 1, 2, \ldots, R$ **do**
        for each client k = 1 to n in **parallel do**
            $S_w \leftarrow$ LocalTraining(S_i, D_k, M_i)
            Send (S_w) to server for aggregation
        end for
        $S_g \leftarrow$ Aggregation(local_updates parameters)
        Distribute updated global model's (S_g) parameters to clients
    end for
    return Fine-tuned student model S parameters
end function
$S^* \leftarrow$ Run FederatedLearning(S, M_i)

Algorithm 2 Independent local Knowledge Distillation with post-Ensemble Federated Learning (IKDEFL)

Require: A set of complex models $M = \{M_1, M_2, \ldots, M_n\}$ at n clients with datasets $D = \{D_1, D_2, \ldots, D_n\}$, Local student models $\{S_1, S_2, \ldots, S_n\}$, Adaptive weights $\{w_1, w_2, \ldots, w_n\}$
Ensure: Fine-tuned ensemble student model S_{ensemble}

Initialize local student models $S_1, S_2, \ldots, S_n$
function LKD(M, D)
    for each client $k = 1$ to n **on its dataset do**
        $S_k \leftarrow$ DISTILL(M_k, D_k)
    end for
    return Local student models $S_1, S_2, \ldots, S_n$
end function
function AVSEM($S_1, S_2, \ldots, S_n$)
    Initialize input layer x based on the shape of S
    Collect outputs (predictions) from each student model $S_i(x)$ in $S_1, S_2, \ldots, S_n$
    Initialize adaptive weights w_i for $i = 1, 2, \ldots, n$ such that $\sum_{i=1}^{n} w_i = 1$ where n = len(student models)
    Combine outputs using the adaptive weights:

$$\text{combined\_output} = \frac{\sum_{i=1}^{n} w_i S_i}{\sum_{i=1}^{n} w_i}$$

    S_{ensemble} = Model(inputs=x, outputs=combined_output)
    return S_{ensemble}
end function
function LocalTraining(S_{ensemble}, D_k)
    Train S_{ensemble} on local dataset D_k
    Update local model parameters and adaptive weights
    return Updated S_{ensemble}
end function
function Aggregation(local_updates)
    Send local updates to central server
    Aggregate updates to form a global model
    return Global model parameters
end function
function FederatedLearning(S_{ensemble})
    $S_i \leftarrow$ Initialize parameters of S_{ensemble}
    for each round $t = 1, 2, \ldots, R$ **do**
        for each client k = 1 to n in **parallel do**
            $S^w_{\text{ensemble}} \leftarrow$ LocalTraining(S_i, D_k)
            Send (S^w_{ensemble}) to server for aggregation
        end for
        $S^g_{\text{ensemble}} \leftarrow$ Aggregation(local_updates parameters)
        Distribute updated global model's (S^g_{ensemble}) parameters to clients
    end for
    return Fine-tuned student model S parameters
end function
$S_1, S_2, \ldots, S_n \leftarrow$ Run LKD(M_k, D_k)
$S_{\text{ensemble}} \leftarrow$ Run AVSEM($S_1, S_2, \ldots, S_n$)
$S^*_{\text{ensemble}} \leftarrow$ Run FederatedLearning(S_{ensemble})

we compare our proposed methods (AFKD and IKDEFL) with the traditional FedAvg and FedAdam algorithms, where the latter use only the student model as the backbone network.

A. Dataset Description and Preprocessing

Three hospitals, H_1, H_2, and H_3, were simulated as federated clients, each with Independent and Identically Distributed (IID) and non-IID datasets. IID datasets consist of data points that are independent and from the same distribution, simplifying the learning process. In contrast, non-IID datasets include data points that may be dependent and originate from different distributions, adding complexity and better reflecting real-world scenarios.

The non-IID datasets for H_1, H_2, and H_3 were sourced from [23], [24]. Teacher models T_1, T_2, and T_3 were trained using COVID-19 CT scan images. Specifically, T_1 was trained on 2500 images with 1400 used for federated training at H_1 [25], T_2 on 6500 images with 3899 for federated training [26], [27], and T_3 on 1200 images with 1339 for federated training. These datasets were also utilized for local knowledge distillation training of the student model as outlined in Algorithm 2. The IID datasets for H_1, H_2, and H_3 were also obtained from [25],

with a comparable number of images to the non-IID datasets. To balance the datasets, 1200 images were sampled from the unbalanced datasets for training. Figure 1 illustrates examples of COVID-19 CT scan images.

Data augmentation techniques such as rotation, flipping, and zoom adjustment were applied to enhance robustness. The images were resized to 200x200x1 pixels, and pixel values were normalized to the range of 0 to 1. For both IID and non-IID datasets, data for H_1, H_2, and H_3 were split 80% for training and 20% for testing. Additionally, 20% of the training data was reserved for validation to fine-tune the models and their hyper-parameters when training teacher models T_1, T_2, and T_3.

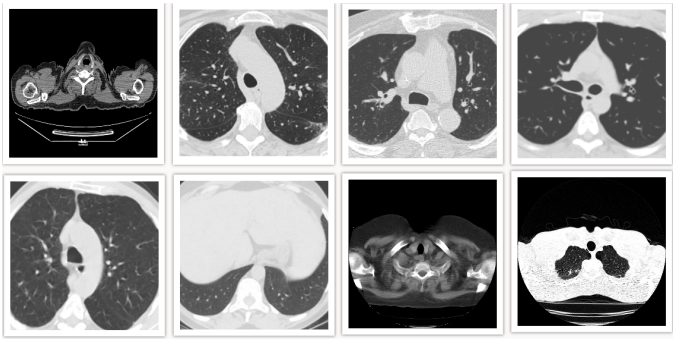

Fig. 1. Example images of patients who are COVID-19 positive are shown in the top row, while images of COVID-19 negative cases are displayed in the bottom row.

B. Evaluation Metrics

To measure the effectiveness of the proposed frameworks, accuracy and the F1-score are utilized. Accuracy calculates the ratio of correct predictions to the total number of predictions. The F1-score, which is the harmonic mean of precision and recall, balances precision (the correctness of positive predictions) and recall (the ability to capture all positive instances). These metrics provide a well-rounded evaluation of the model's performance, as shown in Equation (5) [28].

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}} \\ \text{Recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}} \\ \text{F1-score} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}} \\ \text{Accuracy} &= \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \end{aligned} \tag{5}$$

Thus, the classifiers from Algorithms 1 and 2 are assessed using a confusion matrix, where True Positives (TP) indicate images correctly identified as belonging to a specific class, and False Negatives (FN) indicate images incorrectly classified as not belonging to that class.

C. Results and Analysis

To evaluate our proposed approach, we simulate three hospitals, each with its own dataset as described in section IV-A. Three teacher models, COVID-CNN (T_1), DeepCovid (T_2), and CovidVGG16 (T_3), are trained centrally at each hospital. These teacher models detailed in section II-A, then transfer knowledge to a student model (S) at their respective hospitals via knowledge distillation. The performance of the teacher and student models through centralized training is shown in Table I.

TABLE I
PERFORMANCE OF TEACHER AND STUDENT MODELS ON TEST DATA

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
COVID-CNN (T_1)	93.00	93.16	93.00	93.00
DeepCovid (T_2)	92.23	92.25	92.23	92.23
CovidVGG16 (T_3)	87.50	87.51	87.50	87.50
COVID-CNN (S_1)	88.60	88.63	88.60	88.60
DeepCovid (S_2)	94.46	94.47	94.46	94.46
CovidVGG16 (S_3)	79.16	79.27	79.17	79.13

1) Experiment 1: Federated Knowledge Distillation though Designation: In this experiment, we evaluate the proposed Algorithm 1 (AFKD) as outlined in section III-B1 on test datasets. The FL server optimizer used is Stochastic Gradient Descent (SGD) with a learning rate of 0.5 and a momentum of 0.9. For the client-side optimization, we considered two methods: (1) Stochastic Gradient Descent (SGD) with a learning rate of 0.01 and (2) Adaptive Moment Estimation (ADAM) with a learning rate of 0.0001, β_1 of 0.9, β_2 of 0.99, and ϵ of 1 $\times 10^{-7}$. The KD parameters include the distillation smoothing (α) of 0.5 and distillation temperature (T) of 10. In the federated training for AFKD, each local training session utilizes 1 epoch, followed by the aggregation of updates at a central server. This process is iteratively repeated over 50 communication rounds on our IID and non-IID datasets. The results of this experiment are summarized in Table II and Fig. 2.

2) Experiment 2: Autonomous Local Knowledge Distillation with Subsequent Ensemble Federated Learning: In this experiment, we assess the subsequent proposed Algorithm 2 (IKDEFL) as described in section III-B2 on test datasets. This algorithm combines the strengths of knowledge distillation, federated and ensemble learning. In Algorithm 2, the federated learning training of $S_{ensemble}$ employs different adaptive voting layers (soft-voting, multihead-attention, and transformer blocks), each considered independently. The training is conducted using the FedAvg and FedADAM algorithms, referred to in [24] as $IKDEFL_{SDG}$ and $IKDEFL_{ADAM}$. For both $IKDEFL_{SDG}$ and $IKDEFL_{ADAM}$, the server optimizer employed is SGD with learning rate of 0.5 and a momentum of 0.9. On the client-side, $IKDEFL_{SDG}$ utilizes SGD with a learning rate of 0.01, while $IKDEFL_{ADAM}$ uses the ADAM optimizer with a learning rate of 0.0001, β_1 of 0.9, β_2 of 0.99, and ϵ of 1×10^{-7}. Each local training session at the clients (hospitals) involves one epoch, followed by the aggregation of updates at a central server. This iterative process is repeated over 50 communication rounds on both IID and non-IID datasets. The results of this experiment are summarized in Table III and Fig. 2.

TABLE II
PERFORMANCE OF STUDENT MODELS FROM ALGORITHM 1 ON TEST DATA

FL Algorithm	Hospital (H)	Non-IID Dataset				IID Dataset			
		Unbalanced		balanced		Unbalanced		balanced	
		Accuracy(%)	F1 Score(%)	Accuracy(%)	F1 Score	Accuracy(%)	F1 Score(%)	Accuracy(%)	F1 Score(%)
AFKD$_{sgd}$ (COVID-CNN)	H1	78.92	77.94	81.07	80.43	93.57	93.58	91.42	91.42
	H2	89.48	89.47	86.02	86.01	92.56	92.57	91.53	91.54
	H3	72.01	70.98	73.88	73.31	94.02	94.02	95.14	95.14
AFKD$_{adam}$ (COVID-CNN)	H1	75.71	75.35	80.35	80.38	91.42	91.43	86.07	86.06
	H2	90.51	90.5	80.51	80.21	92.56	92.56	89.1	89.11
	H3	77.61	77.61	73.88	73.84	92.91	92.89	89.17	89.17
AFKD$_{sgd}$ (DeepCovid)	H1	78.57	77.36	77.5	75.89	93.57	93.58	89.64	89.65
	H2	87.3	87.2	86.79	86.77	92.56	92.55	92.05	92.04
	H3	71.26	70.03	63.43	60.1	92.16	92.11	92.16	92.12
AFKD$_{adam}$ (DeepCovid)	H1	58.21	44.61	71.14	71.24	81.78	80.98	75.71	73.71
	H2	86.53	86.41	82.43	82.2	80.12	79.28	77.94	76.99
	H3	47.01	31.29	61.56	57.77	80.22	79.58	73.5	72.03
AFKD$_{sgd}$ (CovidVGG16)	H1	78.82	77.86	82.85	82.57	93.21	93.22	90.35	90.35
	H2	88.97	88.9	85.64	85.63	93.46	93.46	91.28	91.28
	H3	73.5	72.76	76.88	76.67	92.91	92.88	91.04	91.03
AFKD$_{adam}$ (CovidVGG16)	H1	71.78	71.45	71.04	67.61	82.14	81.44	87.14	87.11
	H2	90.25	90.23	85.76	85.77	83.84	83.46	89.35	89.36
	H3	71.64	71.67	64.55	62.65	85.07	84.88	92.16	92.16

TABLE III
PERFORMANCE OF STUDENT MODELS FROM ALGORITHM 2 ON TEST DATA

FL Algorithm	Hospital (H)	Non-IID Dataset				IID Dataset			
		Unbalanced		balanced		Unbalanced		balanced	
		Accuracy(%)	F1 Score(%)	Accuracy(%)	F1 Score	Accuracy(%)	F1 Score(%)	Accuracy(%)	F1 Score(%)
IKDEFLsgd Soft-Voting	H1	81.42	81.22	81.78	81.57	93.21	93.22	90	90.01
	H2	84.48	84.23	86.92	86.86	92.98	92.95	91.66	91.66
	H3	76.49	76.24	77.98	77.85	95.52	95.52	93.65	93.65
IKDEFLsgd MultiheadAttention-Voting	H1	79.28	79.3	82.14	82.04	94.28	94.29	92.14	92.15
	H2	88.84	88.8	85.51	85.5	93.58	93.59	92.17	92.18
	H3	79.47	79.44	78.35	78.26	95.89	95.89	95.52	95.52
IKDEFLsgd TransformerBlock-Voting	H1	85	84.84	79.64	79.73	93.21	93.21	92.14	92.13
	H2	85.38	85.17	83.84	83.72	93.33	93.34	91.66	91.67
	H3	77.99	77.94	79.85	79.86	95.52	95.52	93.65	93.66
IKDEFL$adam$ Soft-Voting	H1	77.5	76.36	82.14	81.76	95.71	95.7	91.78	91.79
	H2	86.28	86.1	80.12	79.5	94.74	94.74	92.17	92.18
	H3	69.4	68.5	74.62	74.01	96.26	96.27	93.65	93.64
IKDEFL$adam$ MultiheadAttention-Voting	H1	81.42	80.98	81.78	80.86	94.28	94.29	86.78	86.6
	H2	83.71	83.39	84.35	84.18	93.58	93.59	90.25	90.23
	H3	70.89	70.39	63.42	60.46	94.77	94.77	92.53	92.54
IKDEFL$adam$ TransformerBlock-Voting	H1	75	72.92	79.64	78.61	94.28	94.28	92.14	92.15
	H2	81.92	81.45	84.61	84.48	93.84	93.85	92.56	92.56
	H3	72.01	71.66	64.17	64.13	95.52	95.52	92.91	92.89

3) Experiment 3: Training with Traditional FL Methods (FedAvg and FedADAM): To evaluate the effectiveness of $AFKD$ and $IKDEFL$, we compared them with traditional federated learning algorithms, FedAvg and FedAdam. In this comparison, only the student model *S* was trained using FedAvg and FedAdam. No additional techniques, such as knowledge distillation or ensemble learning, were incorporated. The federated learning hyperparameters were consistent with those in Experiment 2, and training was conducted over 50 communication rounds on both IID and non-IID datasets. The results are summarized in Table IV.

D. Discussions

1) Performance of AFKD with Teacher Models Across in Diverse Data Settings: Our discussion begins with an examination of Algorithm 1 ($AFDK$) and its performance using COVID-CNN (T_1), DeepCovid (T_2), and CovidVGG16 (T_3) as teacher models. As detailed in II, both $AFKD_{sgd}$ and $AFKD_{adam}$ algorithms show strong performance across all three teachers, particularly excelling in IID datasets. $AFKD_{sgd}$ performed best with COVID-CNN (T_3), achieving high accuracy and F1 scores, ranging from 70% to 89% on non-IID and 91% to 95% on IID datasets. This highlights COVID-CNN's effectiveness in distilling knowledge and enabling the student model to adapt to different data distributions. Similarly, $AFKD_{adam}$ achieved comparable results with COVID-CNN, with accuracy and F1 scores between 73% and 90% on non-IID data and 89% to 93% on IID datasets. DeepCovid also performed well but showed some variability in non-IID datasets, likely due to differences in data distribution. CovidVGG16, while generally strong, exhibited the most variability, especially in non-IID datasets for Hospital 3. Overall, both algorithms performed most reliably with COVID-CNN, making it the preferred choice for maintaining high accuracy and F1 scores across different data distributions. As shown in Fig. 2, balancing the datasets did not lead to significant performance gains.

2) Adaptability of IKDEFL's Voting Methods in Diverse Data Settings: We evaluated the $IKDEFL$ algorithm,

TABLE IV
PERFORMANCE OF STUDENT MODELS FROM FEDAVG AND FEDADAM ON TEST DATA

FL Algorithm	Hospital (H)	Non-IID Dataset				IID Dataset			
		Unbalanced		balanced		Unbalanced		balanced	
		Accuracy(%)	F1 Score(%)	Accuracy(%)	F1 Score(%)	Accuracy(%)	F1 Score(%)	Accuracy(%)	F1 Score(%)
FedAvg	H1	76.42	74.69	79.64	70.07	88.92	88.95	89.28	89.22
	H2	83.71	83.64	80.51	80.36	87.69	87.69	85.76	85.76
	H3	70.14	69.27	73.5	73.5	86.14	86.2	86.56	86.54
FedAdam	H1	76.27	75.22	79.53	79.03	95.35	95.35	93.57	93.57
	H2	82.0	81.68	81.0	80.78	94.87	94.87	93.33	93.33
	H3	63.05	59.56	72.38	71.4	93.28	93.29	90.67	90.68

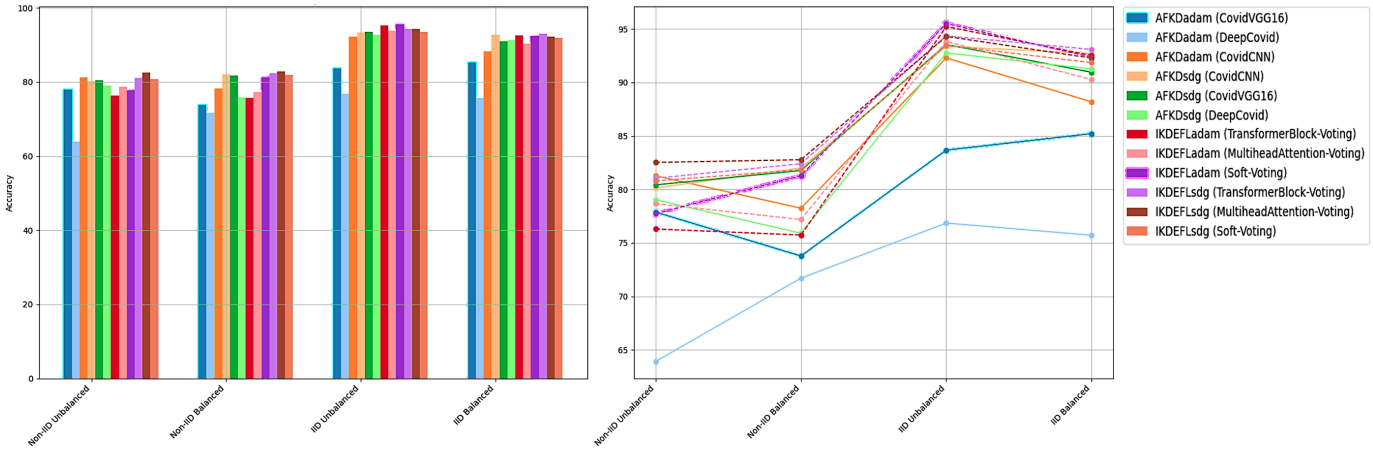

Fig. 2. Performance of Student Models From Algorithms AFKD and IKDEFL on Test Data.

applying different weighted voting methods (Soft-Voting, MultiheadAttention-Voting, and TransformerBlock-Voting) to combine student model outputs. As shown in Table III, both $IKDEFL_{sgd}$ and $IKDEFL_{adam}$ performed strongly across all voting methods on IID and non-IID datasets. $IKDEFL_{sgd}$ excelled with Soft-Voting and MultiheadAttention-Voting, achieving accuracy and F1 scores between 76% and 95%. Although $IKDEFL_{adam}$ showed some variability on non-IID datasets, it remained consistent on IID datasets, particularly with Soft-Voting, where scores ranged from 91% to 96%. Overall, both algorithms were most effective in IID scenarios, with $IKDEFL_{sgd}$ excelling in MultiheadAttention-Voting and $IKDEFL_{adam}$ showing steadiness with Soft-Voting. This analysis highlights the adaptability of IKDEFL's voting methods in enhancing student model performance across varied data distributions. Notably, balancing the datasets did not result in significant performance gains, as shown in Fig. 2

3) Assessment of Proposed Methods (IKDEFL and AFKD) with Traditional FL algorithms FedAvg and FedAdam in Diverse Data Settings: Table IV compares the performance of FedAvg and FedAdam for the student model S without additional techniques like KD or EL, evaluated in both IID and non-IID settings against Algorithms 1 and 2. Our method, $IKDEFL$, consistently outperforms traditional FL strategies, demonstrating superior stability and accuracy, particularly in IID scenarios (Fig. 3). In IID balanced settings, $IKDEFL$ maintains a narrow performance range, with a median accuracy of 93% and a mid-spread of 92% to 93%. In non-IID balanced settings, it shows strong but slightly more variable performance, with a median accuracy of 81% and a mid-spread of 77% to 82%. Conversely, $AFKD$ performs strongly in non-IID settings, with a median accuracy of 82.02% and a mid-spread of 77% to 84%, but shows more variability in IID settings. $FedAvg$ delivers moderate performance across both settings, with a median accuracy of 86% in IID unbalanced datasets and 80% in non-IID balanced settings. $FedAdam$ excels in IID unbalanced settings, achieving a median accuracy of 95%, but struggles with variability in non-IID contexts, where accuracy drops to 76% with a wide mid-spread of 63% to 82%. $IKDEFL$ stands out for its robustness across datasets, consistently balancing stability and accuracy, while $FedAdam$ excels in IID unbalanced settings but requires careful consideration in non-IID environments due to potential variability.

V. CONCLUSION

This study developed the $AFKD$ and $IKDEFL$ Federated Learning frameworks, combining Knowledge Distillation and Ensemble Learning for COVID-19 detection using CT scans. These frameworks enable the creation of diagnostic models that work effectively on resource-limited devices while maintaining data privacy. Our analysis shows that $AFKD$ and $IKDEFL$ outperform traditional methods like FedAvg and FedAdam across various data settings. $AFKD$, especially with the $COVID-CNN$ teacher model, consistently achieves high accuracy and F1 scores, demonstrating its effectiveness in knowledge distillation. $IKDEFL$ also proves

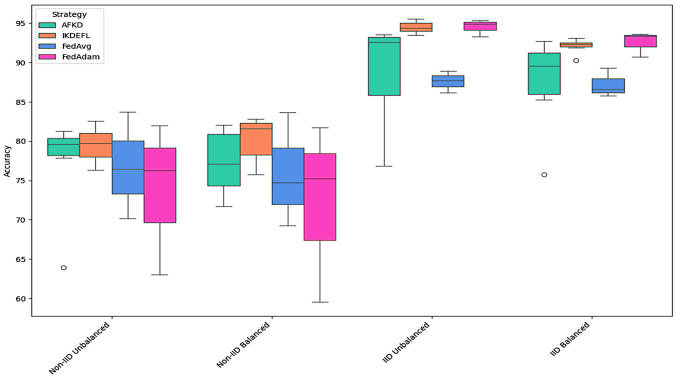

Fig. 3. Comparison of Accuracy Metrics Across Different Federated Learning Algorithms in Non-IID and IID Settings on Test Data

highly adaptable, with accuracy and F1 scores ranging from 92% to 95% on IID datasets and 76% to 88% on non-IID datasets. It particularly excels in IID scenarios, maintaining a median accuracy of around 93% with minimal variability. While $FedAdam$ performs well in IID unbalanced settings with a median accuracy of 95%, it shows significant variability in non-IID contexts. Overall, $IKDEFL$ emerges as the most reliable choice for achieving consistent outcomes across diverse environments. These findings underscore the effectiveness of our approaches in improving model performance in federated learning, offering valuable insights for future applications.

VI. Acknowledgment

This work is supported by the U.S. National Science Foundation under awards 2434487 and 2200138. We thank anonymous reviewers for their insightful comments and inputs.

References

[1] Han, X., Hu, Z., Wang, S. & Zhang, Y. A survey on deep learning in COVID-19 diagnosis. *Journal Of Imaging*. **9**, 1 (2022)

[2] Shoeibi, A., Khodatars, M., Jafari, M., Ghassemi, N., Sadeghi, D., Moridian, P., Khadem, A., Alizadehsani, R., Hussain, S., Zare, A. & Others Automated detection and forecasting of covid-19 using deep learning techniques: A review. *Neurocomputing*. pp. 127317 (2024)

[3] Bhatele, K., Jha, A., Tiwari, D., Bhatele, M., Sharma, S., Mithora, M. & Singhal, S. COVID-19 Detection: A Systematic Review of Machine and Deep Learning-Based Approaches Utilizing Chest X-Rays and CT Scans. *Cognitive Computation*. **16**, 1889-1926 (2024,7,1), https://doi.org/10.1007/s12559-022-10076-6

[4] Yang, W., Sirajuddin, A., Zhang, X., Liu, G., Teng, Z., Zhao, S. & Lu, M. The role of imaging in 2019 novel coronavirus pneumonia (COVID-19). *European Radiology*. **30** pp. 4874-4882 (2020)

[5] Annan, R., Qin, H. & Qingge, L. Generalized Deep Learning Models for COVID-19 Detection with Transfer and Continual Learning. *Proceedings Of The 16th International Conference On Bioinformatics And Computational Biology (BICOB-2024)*. **101** pp. 58-72 (2024), https://easychair.org/publications/paper/FzLP

[6] Mabrouk, A., Díaz Redondo, R., Abd Elaziz, M. & Kayed, M. Ensemble Federated Learning: An approach for collaborative pneumonia diagnosis. *Applied Soft Computing*. **144** pp. 110500 (2023), https://www.sciencedirect.com/science/article/pii/S1568494623005185

[7] Nguyen, D., Ding, M., Pathirana, P., Seneviratne, A. & Zomaya, A. Federated learning for COVID-19 detection with generative adversarial networks in edge cloud computing. *IEEE Internet Of Things Journal*. **9**, 10257-10271 (2021)

[8] Feki, I., Ammar, S., Kessentini, Y. & Muhammad, K. Federated learning for COVID-19 screening from Chest X-ray images. *Applied Soft Computing*. **106** pp. 107330 (2021)

[9] Xia, T., Han, J., Ghosh, A. & Mascolo, C. Cross-device federated learning for mobile health diagnostics: A first study on COVID-19 detection. *ICASSP 2023-2023 IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP)*. pp. 1-5 (2023)

[10] Hinton, G., Vinyals, O. & Dean, J. Distilling the knowledge in a neural network. *ArXiv Preprint ArXiv:1503.02531*. (2015)

[11] Qin, D., Bu, J., Liu, Z., Shen, X., Zhou, S., Gu, J., Wang, Z., Wu, L. & Dai, H. Efficient medical image segmentation based on knowledge distillation. *IEEE Transactions On Medical Imaging*. **40**, 3820-3831 (2021)

[12] Krichen, M. Convolutional Neural Networks: A Survey. *Computers*. **12** (2023), https://www.mdpi.com/2073-431X/12/8/151

[13] Taye, M. Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions. *Computation*. **11**, 52 (2023)

[14] Ghaderzadeh, M., Asadi, F., Jafari, R., Bashash, D., Abolghasemi, H. & Aria, M. Deep Convolutional Neural Network–Based Computer-Aided Detection System for COVID-19 Using Multiple Lung Scans: Design and Implementation Study. *J Med Internet Res*. **23** pp. e27468, http://www.ncbi.nlm.nih.gov/pubmed/33848973

[15] Mohammed, A. & Kora, R. A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal Of King Saud University-Computer And Information Sciences*. **35**, 757-774 (2023)

[16] Delgado, R. A semi-hard voting combiner scheme to ensemble multi-class probabilistic classifiers. *Applied Intelligence*. **52**, 3653-3677 (2022,3,1), https://doi.org/10.1007/s10489-021-02447-7

[17] Latif-Shabgahi, G. A novel algorithm for weighted average voting used in fault tolerant computing systems. *Microprocessors And Microsystems*. **28**, 357-361 (2004)

[18] Yang, Y., Lv, H. & Chen, N. A survey on ensemble learning under the era of deep learning. *Artificial Intelligence Review*. **56**, 5545-5589 (2023)

[19] Kim, T., Oh, J., Kim, N., Cho, S. & Yun, S. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. *ArXiv Preprint ArXiv:2105.08919*. (2021)

[20] Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S. & McMahan, H. Adaptive federated optimization. *ArXiv Preprint ArXiv:2003.00295*. (2020)

[21] Silva, P., Vinagre, J. & Gama, J. Towards federated learning: An overview of methods and applications. *Wiley Interdisciplinary Reviews: Data Mining And Knowledge Discovery*. **13**, e1486 (2023)

[22] Khan, M., Glavin, F. & Nickles, M. Federated learning as a privacy solution-an overview. *Procedia Computer Science*. **217** pp. 316-325 (2023)

[23] Soares, E., Angelov, P., Biaso, S., Froes, M. & Abe, D. SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification. (2020), https://www.medrxiv.org/content/early/2020/05/14/2020.04.24.20078584

[24] Ghaderzadeh, M., Asadi, F., Jafari, R., Bashash, D., Abolghasemi, H. & Aria, M. Deep Convolutional Neural Network–Based Computer-Aided Detection System for COVID-19 Using Multiple Lung Scans: Design and Implementation Study. *J Med Internet Res*. **23** pp. e27468, http://www.ncbi.nlm.nih.gov/pubmed/33848973

[25] Maftouni, M., Law, A., Shen, B., Zhou, Y., Ayoobi Yazdi, N. & Kong, Z. A Robust Ensemble-Deep Learning Model for COVID-19 Diagnosis based on an Integrated CT Scan Images Database.

[26] Cohen, J., Morrison, P., Dao, L., Roth, K., Duong, T. & Ghassemi, M. COVID-19 Image Data Collection: Prospective Predictions Are the Future. *ArXiv 2006.11988*. (2020), https://github.com/ieee8023/covid-chestxray-dataset

[27] Rahman, T., Khandakar, A., Kadir, M., Islam, K., Islam, K., Mazhar, R., Hamid, T., Islam, M., Kashem, S., Mahbub, Z. & Others Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. *Ieee Access*. **8** pp. 191586-191601 (2020)

[28] Miao, J. & Zhu, W. Precision–recall curve (PRC) classification trees. *Evolutionary Intelligence*. **15**, 1545-1569 (2022)