

RESEARCH ARTICLE

Decentralised Federated Learning for Hospital Networks With Application to COVID-19 Detection

ALESSANDRO GIUSEPPI¹, (Member, IEEE), SABATO MANFREDI²,
DANILO MENEGATTI¹, (Student Member, IEEE), CECILIA POLI³,
AND ANTONIO PIETRABISSA¹, (Senior Member, IEEE)

¹Department of Computer, Control, and Management Engineering “Antonio Ruberti” (DIAG), University of Rome Sapienza, 00185 Rome, Italy

²Department of Electrical Engineering and Information Technology, University of Naples Federico II, 80125 Naples, Italy

³National Centre for Innovative Technologies in Public Health Department, Istituto Superiore di Sanità, 00161 Rome, Italy

Corresponding author: Alessandro Giuseppe (giuseppi@diag.uniroma1.it)

This work was supported in part by the Lazio region, in the scope of the Project FedMedAI, POR FESR Lazio 2014–2020 (Azione 1.2.1), under Grant A0375-2020-36491-23/10/2020.

ABSTRACT Federated Learning (FL) is a distributed machine learning technique which enables local learning of global machine learning models without the need of exchanging data. The original FL algorithm, Federated Averaging (FedAvg), is extended in this work by means of consensus theory. Differently from standard FL algorithms, the resulting one, named FedLCon, does not need a coordinating server, which represents a single failure point and needs to be trusted by all the clients. Furthermore, the consensus paradigm is also applied to the Adaptive Federated Learning (AdaFed) algorithm, which extends FedAvg with an adaptive model averaging procedure. Performance comparison tests are performed over a real-world COVID-19 detection scenario.

INDEX TERMS Federated learning, heuristic algorithms, deep neural networks, distributed systems, consensus, COVID-19 detection.

I. INTRODUCTION

Federated Learning (FL) was originally introduced in [1] with the Federated Averaging (FedAvg) algorithm, as an alternative to conventional approaches for training learning models using data coming from mobile devices. Unlike distributed optimization, FL deals with non-IID, unbalanced, and massively distributed data by means of a federation of clients communicating with a central server in a privacy-preserving way that avoids the exchange of any data. Contrary to a standard distributed learning setting, depicted in figure 1 a), where data are shared and partitioned by a server among the clients for distributed processing, the standard FL setting envisages that all data are locally processed by the clients and the server task is only to perform a so-called model averaging procedure. In fact, the FL server iteratively updates its model during each *communication round* by averaging the trainable

weights gathered from the clients of the federation after having trained on their *locally available* data, as depicted in 1 b).

The original FedAvg [1] algorithm averages the clients models with an a-priori weighting strategy that typically depends on the numerosity of the various clients datasets. On the contrary, the Adaptive Federated Learning (AdaFed) algorithm [2], recently proposed by the authors, envisages a dynamic and adaptive heuristic weighting scheme that takes into account the performance attained by the various clients, so that the impact of low performing and/or malicious clients can be reduced.

Both FedAvg and AdaFed share the same overall architecture, with a centralized server orchestrating the learning procedure of the entire federation. On the one hand, this provides a communication-efficient solution with privacy guarantees, but on the other hand, it introduces a single point of failure vulnerability in the system. Moreover, a centralized architecture of this kind requires all the clients to completely

The associate editor coordinating the review of this manuscript and approving it for publication was Valentina E. Balas¹.

trust the server behaviour, since they do not have any means to verify it and do not communicate with each other.

To this end, a possible solution would be to allow *point-to-point* client agreements, as depicted in Figure 1 c), so that information is directly exchanged only with trusted entities. This work aims at developing a *fully decentralized* solution able to extend the FL paradigm to a group of federated entities of arbitrary topologies. The main challenge to be faced in such a decentralized setting is to assure the convergence of the clients models towards a common one [3]. In this direction, the present work proposes a scheme that relies on consensus theory, as introduced in the previous work [4]. The developed framework may find application over federations characterized by sparse communication graphs, as depicted in Figure 2, that arise in scenarios in which broader communications are prevented by regulations and/or physical connections.

The highlights and main contributions of this work are:

- The FL framework is extended to a fully decentralized setting, that does not require a coordinating server and is robust with respect to clients with poor quality data and malicious ones, by means of a consensus-based algorithm;
- The consensus-based FL algorithm FedLCon is designed so that it may provide a seamless decentralization solution for any FL algorithm that employs a centralized model averaging procedure;
- FedLCon, is applied to decentralize both FedAvg and AdaFed;
- Validating examples are discussed, demonstrating the applicability and performance properties on the X-Rays COVID-19 detection problem.

The remainder of the paper is organized as follows: Section II discusses the relevant works in the literature; Section III shows how the COVID-19 detection problem is formulated in a real-world decentralized scenario; Section IV presents the proposed algorithms; Section V reports the results of our tests; Section VI draws the conclusions and discusses future research directions.

II. RELATED WORKS

This section first presents a brief overview of the application of FL algorithms to the healthcare domain and then discusses some recent Deep Learning (DL) applications for COVID-19 detection.

A. FEDERATED LEARNING FOR HEALTHCARE

The distributed and privacy preserving nature of FL makes it an ideal choice for healthcare applications. In fact, FL has been broadly applied wherever the collaboration of stakeholders (e.g., hospitals, laboratories, public bodies) would have improved significantly the handling of a specific problem, such as for electronic health records (EHRs) management, remote health monitoring and medical imaging [6].

In particular, EHRs are used to store patients' health information, such as diagnoses, medications and analyses, in a digital format [7]. EHRs are hence a valuable source of data

that could be used and studied to facilitate the diagnosis and assessment of various diseases, but their sharing among different institutions poses several regulatory challenges. In [8] the authors address both the requirement of extensive computational resources and the vulnerability of transmission channels by proposing a privacy-aware and resource-saving collaborative learning protocol (PRCL). PRCL, based on a FL framework, makes use of a so-called model splitting method, that consists in outsourcing the most computing demanding part of the learning procedure to a cloud server and of a lightweight data perturbation method to prevent direct and indirect data leaks. The concept of splitting is also employed in [9], where a FL-based method called SplitNN is introduced; the basic idea is to split the model into multiple parts, each one trained by a single client, thus not requiring raw data sharing. Reference [10] uses a FL setting based on the soft-margin l_1 -regularized sparse Support Vector Machine (sSVM) classifier and on an iterative cluster Primal Dual Splitting (sPDS) algorithm to predict hospitalizations for cardiac events within a calendar year, based on information in the patients EHRs prior to that year. The authors of [11] focus on privacy aspects by introducing a FL framework based on a differential privacy mechanism whose effectiveness is proved using real-world EHRs of 1 million patients. The concept of differential privacy is employed also in [12] where artificial noises are added to the parameters at the clients side before aggregation. The work [13] aims at predicting patients mortality from their EHRs by means of a FL method called Federated-Autonomous Deep Learning (FADL), whose main novelty is that part of the model is trained by using all data sources in a distributed manner and other parts are trained using specific data sources. Prediction of patients mortality as a binary classification problem is explored in [14] where a stochastic FL algorithm called Stochastic Channel-Based Federated Learning (SCBF) is introduced; the privacy of each client is enforced by updating the weight of the server model by stochastically selecting the clients whose local gradients presented the largest magnitude. In [15], the problem of mortality and ICU stay-time forecasting are addressed via a FL-based algorithm called Community-Based Federated Machine Learning (CBFL) which clusters patients into clinically meaningful communities capturing similar diagnoses and geographical locations, learning then a different model for each cluster. Finally, the authors of [16] address the preterm-birth prediction by means of a novel FL algorithm called Federated Uncertainty-Aware Learning Algorithm (FUALA) which weights the contribution of each client based on its performance and introduces ensembling at prediction time.

Other than EHRs, a significant amount of healthcare-related data is produced by wearable devices, that allow for remote monitoring of patients activity and health status, gathering information like blood pressure and oxygenation levels, heart rate, sleep cycles and several other indicators. The work [17] proposes FedHealth, a FL algorithm tailored for wearable healthcare, which performs data aggregation

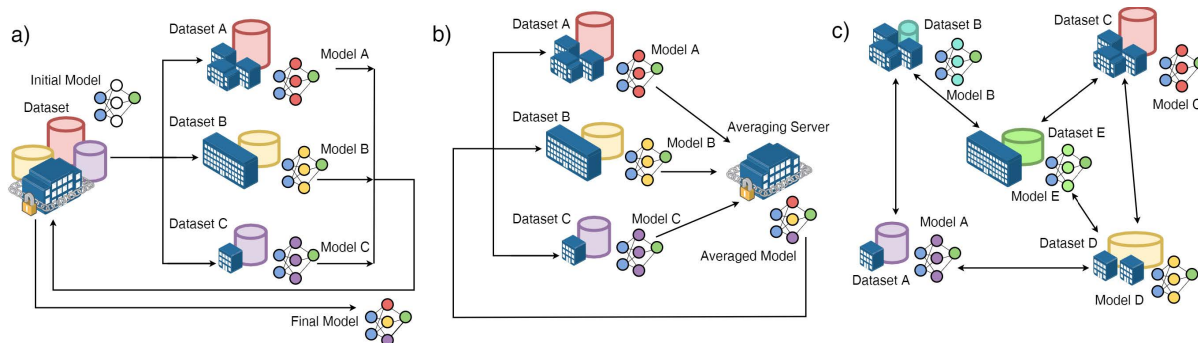


FIGURE 1. a) Typical distributed learning setting, where a central entity distributes data and governs the training process; b) Standard (centralized) Federated Learning, that does not envisage any data sharing as only the Neural Networks are shared, where a server is responsible for a model averaging procedure performed based on the locally trained models; c) Decentralized Federated Learning, where there is no central entity that coordinates the training process and the federation clients only exchange information over point-to-point agreements. The Figure has been derived from the figure in [5].

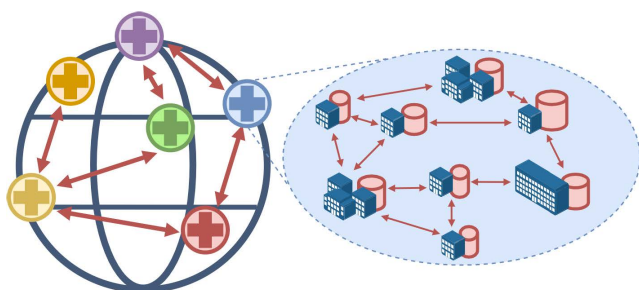


FIGURE 2. Example of a hierarchical federation with sparse communications: a high level point-to-point communication topology is used to link local federations characterized by a higher communication degree. FL may be used at both levels, with decentralised solutions becoming more appealing on sparser federations.

in a privacy-preserving way thanks to a standard FL framework, and builds personalized models using transfer learning. In [18], the authors propose to tackle the problem of heterogeneity in labels across users via model distillation techniques, and proves the validity of the approach on the Heterogeneity Human Activity Recognition (HHAR) dataset. Mood prediction and monitoring is carried out in [19], where metadata related to the use of a virtual keyboard, such as key letters, special characters and phone accelerometer values are used in a FL setting called FedMood, with fusion methods applied for data normalization.

Among data produced by healthcare facilities, medical images constitute one of the most common diagnostic tool for a vast number of diseases, as they reveal the interior of a body and offer a visual representation of the state of several organs and tissues. In [20], the authors developed a FL model to support brain tumor segmentation using Deep Neural Networks (DNNs) together with differential privacy techniques to enhance privacy among clients, evaluating the resulting framework over the BraTS dataset [21]. In [22], brain structural relationships are investigated using magnetic resonance images (MRI) scans distributed across institutions via a FL framework which accounts for data standardization,

confounding factors correction, and multivariate analysis related to the variability of high-dimensional features. Brain tissue-based MRI analytics via FL is also investigated in [5], and FL is also used to perform reconstruction of MRI from under-sampled data [23]. Li et al. [24] proposes a FL method to perform functional MRI (fMRI) classification via a randomization mechanism to coordinate the weight sharing process. The recent study [25] shows a real-world application of FL to breast density classification based on Breast Imaging, Reporting and Data Systems (BI-RADS) [26].

B. DEEP LEARNING FOR COVID DETECTION

Artificial Intelligence (AI) techniques have been largely applied to in the fight against Covid-19. In this context, a significant share of the Deep Learning (DL) applications are related to autonomous detection of COVID-19 cases from X-ray images and Computed Tomography (CT) scans. In [27], a 3D DNN is used to predict the probability of COVID-19 infectious over a dataset of 499 chest CT samples, each one with the lung region segmented using a pre-trained UNet [28]. In this direction, [29] proposes a DL algorithm capable of detecting, localizing, and quantifying the severity of COVID-19 manifestation from chest CT scans, as well as unsupervised clustering of abnormal slices. First the lung region of interest (ROI) is localized in a chest scan, then a 2D ROI classification network classifies the considered ROI as normal vs abnormal. The GradCam method is then used to obtain a fine-grained map of the extracted pathological tissue, enabling for the creation of a so-called *Corona score*, related to the volumetric measure of the disease extent. Finally, unsupervised clustering of normal and abnormal slices is carried out to learn the different patterns of the abnormal manifestation of the disease.

The use of chest CT scans for COVID-19 detection is also explored in [30], [31], and [32]. In [33], two “infection” metrics were introduced; the use of DL techniques on chest CT scans makes possible to quantify the volume of infection and percentage of infection.

Compared to CT scans, X-ray images are cheaper and faster to obtain, that is why a large number of works based on them can be found in the literature. In [34] a deep convolutional neural network (CNN) model, called COVID-Net, for the detection of COVID-19 cases based on chest X-ray (CXR) images, as well as the COVIDx benchmark dataset. In [35], transfer learning is used for COVID-19 classification from CXR and, in particular, a deep CNN called DeTraC is introduced. Similar approaches are followed by [36], [37], and [38], where the authors make use of transfer learning for COVID-19 detection.

The usage of FL in the context of COVID analysis has already been explored in works such as [39], [40], and [41], where the authors compare the detection performance of various DL models also employing transfer learning, [42] where FedAvg is employed to predict the future oxygen requirements of symptomatic patients in a federation of 20 institutions. The authors of [43] propose a dynamic logic for the clients participation in the averaging procedure, depending on both their computational time and model performance - following a logic similarly to the one behind AdaFed - while [44] focuses on mortality prediction in hospitalized patients.

We mention that DL has also been used for other critical tasks related to the pandemic, e.g., for modeling the disease transmission dynamics [45], for “drug-repurposing” [46], [47], for drug discovery [48] - where a DL model named CogMol first learn candidate molecules that can interfere with the COVID-19 virus, and then generate candidate drugs - and for protein structure prediction [49], [50].

Overall, FL is a promising solution for healthcare application that may even be an enabling technology in scenarios where privacy is critical and data sharing is prevented by strict regulations.

III. PROBLEM FORMULATION

The COVID-19 pandemic underlined the importance of scientific collaboration and knowledge sharing to overcome the great challenges that a treat to the entire world poses. Motivated by this, we explore the problem of COVID-19 detection using X-Rays images in a federated setting, namely considering a network of hospitals that collaborate with each other by sharing the X-Rays images of their patients in a privacy-preserving and communication-efficient way.

We model the hospital network by a graph, whose vertices represent the various hospitals and whose vertices capture the possibility of communication between pairs of hospitals. In fact, due to regulatory and trust reasons, some hospitals may be prevented from directly sharing information with certain institutions: for this reason, we will conduct the following analysis considering a sparse/non-complete graph.

The remainder of the section will present the needed background on Consensus Theory and FL algorithms.

A. BACKGROUND ON CONSENSUS THEORY

Given a graph of N vertices connected by a set of edges, the general goal of a consensus algorithm is to derive a fully

distributed information-sharing law that steers the state of all the various clients (*agents*) towards a common *consensus value*.

The following matrices are defined: the adjacency matrix $A = (a_{ij})_{i,j \in I} \in \mathbb{R}^{N \times N}$, where I is the set of N clients and with $a_{ij} = 1$ if an edge connects clients i and j , and $a_{ij} = 0$ otherwise; the out-degree diagonal matrix $O = (o_{ij})_{i,j \in I} \in \mathbb{R}^{N \times N}$, with $o_{ij} = \sum_j a_{ij}$ computed as the clients' out-degree; the Laplacian matrix $L := O - A$, and the diagonal matrix $P = \text{diag}(p_i)_{i \in I} \in \mathbb{R}^{N \times N}$, with p_i representing the weight given to client i .

Let $x_i(t)$ be the state of agent i at time-step t , and let I_i be its set of neighbors. Under the hypothesis of a connected and undirected consensus graph,¹ under the following discrete-time update rule:

$$x_i(t+1) = x_i(t) + \frac{\epsilon}{p_i} \sum_{j \in I_i} a_{ij}(x_j(t) - x_i(t)), \quad (1)$$

in which the sampling time ϵ in such that $\epsilon < \min_{i \in I} (p_i / o_{ii})$ [51], [52], the clients reach a consensus value on their states x_i that coincides with the weighted average of their initial conditions:

$$\bar{x} = \frac{\sum_{i \in I} p_i x_i(0)}{\sum_{i \in I} p_i}. \quad (2)$$

The convergence of the agents follows the dynamics of the discrete-time systems (1), that can be equivalently written in matrix form [52] as

$$x_i(t+1) = H_p x(t), \quad (3)$$

with $H_p = I - \epsilon P^{-1} L$.

From (3), starting from the well known definition of dominant time constant for a discrete time linear time invariant system and its settling time [53], it follows that the agents will reach convergence, with a 99% precision, after a number of steps n_ϵ :

$$n_\epsilon = 5 \cdot \max_i \left\lceil \frac{-1}{\ln(|\lambda_i(H_p)|)} \right\rceil, \quad (4)$$

where $\lambda_i(H_p)$ is the i -th eigenvalue different from 1 of the matrix H_p and $\lceil \cdot \rceil$ denotes the ceiling function of its argument, with a resulting 1%-settling time $t_a \approx n_\epsilon \cdot \epsilon$.

B. BACKGROUND ON FEDERATED LEARNING

Suppose to consider a set I of N federated clients, sharing the same DL model architecture, i.e. a deep neural network. Let $D_i = \{(\alpha_n, \beta_n), n \in \{1, \dots, |D_i|\}\}$ be the dataset of client $i \in I$, with cardinality $|D_i|$, and w_i be the vector of its trainable parameters. We denote the total available data as $\mathbb{D} = \bigcup_i D_i$.

In the federation, each client i is trained to minimize the loss function $l_i((\alpha_n, \beta_n) | w_i)$ over its entire dataset D_i : given the generic input α_n , the loss function is used to quantify the quality of the model, with parameters w_i , against the

¹In strongly connected undirected graph, at least one path from each vertex to every other vertex exists and all the edges are bidirectional.

corresponding ground truth value β_n , with $(\alpha_n, \beta_n) \in D_i$. The choice of the loss function depends on the particular machine learning problem to be addressed. In general, regression tasks employ the mean squared error, whereas classification ones the categorical cross-entropy. We set:

$$L_i(w_i) = \frac{1}{|D_i|} \sum_{(\alpha_n, \beta_n) \in D_i} l_i((\alpha_n, \beta_n)|w_i) \quad (5)$$

as *loss function* of client i over its entire dataset D_i . The goal of the federation is then to find the optimal vector w^* of parameters that, when shared by all clients, solves the minimization problem with joint cost function defined as [54]:

$$\min_w \mathcal{L}(w) := \sum_{i \in I} \bar{p}_i L_i(w) \quad (6)$$

with $\bar{p}_i = |D_i|/|\mathbb{D}|$. While in the standard machine learning setting optimization (6) is tackled by a centralized system, which computes the gradient $\nabla L(w)$ given the whole set \mathbb{D} , in a distributed one, the gradient $\nabla L(w)$ has to be estimated starting from the gradients of the clients $\nabla L_i(w_i)$.

Moreover, in standard (non-federated) distributed learning, data can be distributed arbitrarily by a centralized entity over the clients. The typical assumption for this distribution is that the datasets D_i are IID with respect to \mathbb{D} , implying $\mathbb{E}[L_i(w)] = \mathcal{L}(w)$. In practice, under this assumption $L_i(w)$ provides a good approximation of $\mathcal{L}(w)$ [1] and the locally computed gradients $\nabla L_i(w_i)$ can be averaged to reconstruct $\nabla L(w)$.

On the contrary, in the federated setting such IID hypothesis can not be assumed, as the training data is processed without any re-distribution and $L_i(w)$ could provide an arbitrarily bad approximation of $\mathcal{L}(w)$. For this reason, in FedAvg [1] the author proposed a round-based iterative procedure for model averaging.

FedAvg is divided into two main phases, which are repeated iteratively. In the first phase (*local training*), the server selects a subset of clients that update the weights of their models by training on their local datasets D_i with a gradient descent update rule:

$$\tilde{w}_i(t) = w_i(t-1) - \eta \nabla L_i(w_i(t-1)) \quad (7)$$

where $0 < \eta < 1$ is the learning rate and $\tilde{w}_i(t)$ is the locally updated weight of the model of agent i at time-step t . We mention that in the FL setting it is typically assumed that all clients share a common initial weight vector, i.e., $w_i(0) = w_0 \forall i \in I$ [1]. Actually, the local weight update is performed iteratively over a given number E of *training epochs* using a variation of gradient descent (*mini-batch gradient descent*) that splits D_i 's into a set of *mini-batches*. For the sake of simplicity, equation (7) exemplifies the update rule with $E = 1$ and over the complete dataset, whereas the pseudo-code presented below reports the mini-batch multi-epoch version of the algorithm.

In the second phase (*centralized averaging*), the server collects the \tilde{w}_i 's, computes the weight vector $w(t)$ as the

Algorithm 1 FedAvg

```

1: SERVER UPDATE:
2: for each communication round  $t = 1, \dots, T$  do
3:   select a subset of clients for the averaging procedure
4:   for all selected client  $i$  do
5:     CLIENT UPDATE
6:     receive  $\tilde{w}_i$  from client  $i$ 
7:   end for
8:   set  $w(t) = \sum_i \bar{p}_i \tilde{w}_i(t)$ 
9:   propagate  $w$  in the federation ( $w_i(t) = w(t), \forall i$ )
10: end for

11: CLIENT UPDATE:
12: for each local epoch  $e = 1, \dots, E$  do
13:   for each mini-batch  $b$  from  $D_i$  do
14:      $w_i(t-1) \leftarrow w_i(t-1) - \eta \nabla L_i(b|w(t-1))$ 
15:   end for
16: end for
17: set  $\tilde{w}_i(t) = w_i(t-1)$ 
18: return  $\tilde{w}_i(t)$  to the server

```

weighted average

$$w(t) = \sum_{i \in I} \bar{p}_i \tilde{w}_i(t) \quad (8)$$

and propagates the weight vector $w(t)$ to all the clients:

$$w_i(t) = w(t), \quad \forall i \in I. \quad (9)$$

We report the pseudo-code for FedAvg (see Algorithm 1), showing an implementation where the clients perform E local training epochs using mini-batch Gradient Descent with a batch size of B . In the code, $\nabla L_i(b|w_i)$ denotes the gradient performed over the mini-batch b and it is assumed for simplicity that all clients participate in the averaging procedure.

Like several other variants of FedAvg [54], [55], AdaFed shares the same centralized setting and the two-phase approach, however at its backbone there are an adaptive model averaging procedure paired with an adaptive loss function that heuristically provide a more resilient solution to imbalanced data distributions:

- **Weighted Model Average:** During the server update, the performance $\tilde{p}_i(t)$ of each client model is evaluated over a common test set, and each model is weighted accordingly in the averaging;
- **Adaptive Loss:** The server propagates to the federation both the updated model and a new loss function, adapted to the performance $p(t)$ of its own model over a dedicated test set according to a use-case dependant metric.

We report the pseudo-code for AdaFed (see Algorithm 2) and refer the reader to [2] for a more detailed discussion of the algorithm.

IV. PROPOSED FL ALGORITHM

Similarities between the FL framework and the one for discrete-time weighted average consensus may be found in

Algorithm 2 AdaFed

```

1: SERVER UPDATE:
2: for each communication round  $t = 1, \dots, T$  do
3:   ClientsUpdate
4:   for each client  $i = 1, 2, \dots, K$  do
5:     receive  $\tilde{w}_i$  from client  $i$ 
6:     evaluate  $\tilde{w}_i$  on the server test set
7:     use the evaluation to determine the weight  $\tilde{p}_i$ 
8:   end for
9:   set  $w(t) = \frac{\sum_i \tilde{w}_i(t) \tilde{p}_i(t)}{\sum_i \tilde{p}_i(t)}$ 
10:  evaluate its performance  $p(t)$  on the server test set
11:  adapt the loss function  $l(t)$  depending on  $p(t)$ 
12:  propagate  $w(t)$  and  $l(t)$  to the clients
13: end for

14: ClientsUpdate:
15: for each local epoch  $e = 1, \dots, E$  do
16:   for each mini-batch  $b$  from  $D_i$  do
17:      $w_i(t-1) \leftarrow w_i(t-1) - \eta \nabla L_i(b|w(t-1))$ 
18:   end for
19: end for
20: set  $\tilde{w}_i(t) = w_i(t-1)$ 
21: return  $\tilde{w}_i(t)$  to the server

```

the interpretation of the weights $w_i(t)$ of the FL clients as the states $x_i(t)$ of a set of agents seeking consensus (even if the former are not dynamical systems). On this interpretation, in [4] we proposed to combine (7)-(9) and (1) as described in the following.

At each communication round t , the weights \tilde{w}_i of each client are computed by (7), but the update of the weights vectors $w_i(t)$ is performed via a *consensus round*. Let k be the consensus round index and recall that n_ϵ is the number of iterations required to reach consensus within the round. To reach consensus the federated clients exchange information n_ϵ times, starting from the initial values $x_i(0) = \tilde{w}_i(t), \forall i \in I$. The following iteration rule is executed for $k = 0, \dots, n_\epsilon - 1$:

$$x_i(k+1) = x(k) + \frac{\epsilon}{|D_i|} \sum_{j \in I_i} a_{ij}(x_j(k) - x_i(k)), \quad (10)$$

with n_ϵ computed by equation (4).

Due to the structure of the update rule (10) and the convergence properties of (1), already discussed in Section III-A, one has that

$$x_i(n_\epsilon) \approx \frac{\sum_i |D_i| \tilde{w}_i(t)}{|\mathcal{D}|}, \quad \forall i \in I, \quad (11)$$

i.e., at the end of the communications (when consensus is reached among the federated clients), the proxy variables x_i approximate the weights $w(t)$ computed by the centralized FL case with equation (8). Setting

$$w_i(t) = x_i(n_\epsilon), \quad (12)$$

Algorithm 3 FedLCon Applied to FedAvg

```

1: DECENTRALIZED FEDERATED TRAINING:
2: for each communication round  $t = 1, \dots, T$  do
3:   for all clients  $i \in I$  do
4:     for each local epoch  $e = 1, \dots, E$  do
5:       for each mini-batch  $b$  from  $D_i$  do
6:          $w_i(t-1) \leftarrow w_i(t-1) - \eta \nabla L_i(b|w(t-1))$ 
7:       end for
8:     end for
9:     set  $\tilde{w}_i(t) = w_i(t-1)$ 
10:   end for
11:   update  $w_i(t)$  via a CONSENSUS ROUND
12: end for

13: CONSENSUS ROUND:
14: Compute  $n_\epsilon$  according to (4) depending on the topology
15: Set  $x_i(0) = \tilde{w}_i(t), \forall i \in I$ 
16: for  $k = 0, \dots, n_\epsilon - 1$  do
17:   for all clients  $i \in I$  do
18:     update  $x_i$  according to (10)
19:   end for
20: end for
21: set  $w_i(t) = x_i(n_\epsilon), \forall i \in I$ 

```

the procedure can be repeated starting from the training of equation (7) for all communication rounds t . Note that each communication round t now yields n_ϵ information exchanges since it involves a consensus round, but at the same time it does not envisage the presence of any centralized entity.

The resulting consensus-based distributed FL algorithm (FedLCon) [4], applied to decentralize FedAvg, is reported as a pseudo-code (see Algorithm 3) in the same form of the two previous cases.

Remark 1. As the consensus round is transparent to the FedAvg algorithm, different consensus algorithms can be used to exploit the communication and/or topology properties of the application scenarios.

Remark 2. Regarding the application of FedLCon to AdaFed, there are some design choices to be made depending on the use case characteristics. If all clients share a common validation dataset, on which they evaluate their model performance, they can directly obtain their weight $\tilde{p}_i(t)$ at the start of the consensus round and then update their loss functions at its end, when all the clients share a practically identical model. On the contrary, if each client has a different validation dataset, a possible solution is to let each client assign a performance weight to its neighbours; for example, each client i may compute the weight $\tilde{p}_i(t)$ by averaging the performance of the model of its neighbours tested on their own test set. Independently from this choice, at the end of the consensus round all the clients models will converge towards the weighted average model envisaged by AdaFed. The pseudo-code for the decentralized version of AdaFed is reported in Algorithm 3.

Remark 3. The introduction of the consensus round (and its n_ϵ information exchanges) yields a communication overhead, that is the main disadvantage of the proposed algorithm. From (4), it is clear that n_ϵ is influenced by the eigenvalues of the matrix H_p , that in turn depend on the communication network Laplacian matrix L . We mention that, in general, such eigenvalues do not depend directly on the number of clients in the federation and instead capture the topology connectivity level, meaning that the scalability of FedLCon is mostly affected by the number of links available in the communication topology and their positioning. Note that, in the cases in which the communication overhead becomes not negligible in terms of both bandwidth consumption and training time, one may still deploy adequate counter measures, such as resorting to a transfer learning approach to limit the amount of trainable parameters or multi-hop consensus protocols that virtually increase the federation connectivity.

Remark 4. The requirement of completing n_ϵ information exchanges every consensus round causes FedLCon-based solutions to require more time to complete the model averaging process than centralized ones. We mention that, in general, the information exchange process is expected to require a significant lower amount of time than the training, making the impact of this overhead negligible in most application.

The discussion of this section can be summarized by the following theorem:

Theorem 1: By exchanging information following the consensus-based protocol (10) n_ϵ times, a federation of distributed clients is able to conduct a decentralized model averaging procedure that is equivalent to the one obtainable in a centralized setting. In fact, discrete-time dynamical systems and consensus theory assure that, with 99% precision, the decentralized and centralized averaged models are identical. Furthermore, since the learning process and the model averaging procedures are decoupled, (10) allows for effectively decentralizing any model averaging-based FL algorithm.

V. SIMULATIONS

In this Section, we compare the FedLCon paradigm applied to FedAvg and AdaFed over the X-Rays COVID-19 detection problem. Note that, unlike in the original FedAvg formulation, all the clients are involved in the averaging process by means of the consensus rounds. We consider a federation of $|I| = 7$ clients and we set the parameters of the algorithms as local client epochs $E = 4$, communication rounds $T = 15$, and local batch size $b = 64$. In the AdaFed implementation, the performance weight \tilde{p}_i of the i -th client is computed as its accuracy over a common evaluation dataset of 4876 samples. The model of each client is a Transfer Learning one [56], composed of the VGG19 [57] network trained for ImageNet [58] and one dense layer with 1024 neurons and ReLU [59] activation function. Being the problem at hand a binary classification problem, a binary cross-entropy loss function is employed. The class-dependant weights of the loss function are set to be inversely proportional to the performance $p(t)$ of the model the federation evaluation set with

Algorithm 4 FedLCon Applied to AdaFed

```

1: DECENTRALIZED FEDERATED TRAINING:
2: for each communication round  $t = 1, \dots, T$  do
3:   for all clients  $i \in I$  do
4:     for each local epoch  $e = 1, \dots, E$  do
5:       for each mini-batch  $b$  from  $D_i$  do
6:          $w_i(t-1) \leftarrow w_i(t-1) - \eta \nabla L_i(b|w(t-1))$ 
7:       end for
8:     end for
9:     set  $\tilde{w}_i(t) = w_i(t-1)$ 
10:    evaluate  $\tilde{w}_i(t)$  on the federation test set
11:    use the evaluation to determine the weight  $\tilde{p}_i$ 
12:  end for
13:  update  $w_i(t)$  via a CONSENSUS ROUND
14:  evaluate the performance  $p(t)$  of  $w_i(t)$ 
15:  adapt the loss function  $l_i$  depending on  $p(t)$ 
16: end for

17: CONSENSUS ROUND:
18: Compute  $n_\epsilon$  according to (4) depending on the topology
19: Set  $x_i(0) = \tilde{w}_i(t), \forall i \in I$ 
20: for  $k = 0, \dots, n_\epsilon - 1$  do
21:   for all clients  $i \in I$  do
22:     update  $x_i$  according to (10), replacing  $D_i$  with  $\tilde{p}_i$ 
23:   end for
24: end for
25: set  $w_i(t) = x_i(n_\epsilon), \forall i \in I$ 

```

respect to its corresponding class, and are updated after each communication round by means of its F_1^c -score, i.e., $\kappa^c = 1/(F_1^c + \epsilon)$, $\epsilon = 0.1$, where c denotes the considered class.

A. DATASET DESCRIPTION

To tackle the COVID-19 detection problem, we make use of the COVID-19, Pneumonia and Normal Chest X-ray PA Dataset [60] and formulate the problem as a binary classification one, i.e., the detection of COVID-19 cases. Standard data augmentation is performed by rotating ($+45^\circ$, -45°) and flipping upside down each image to increase the dataset numerosity. Before the augmentation and before dividing data among $N = 7$ clients, the dataset is shuffled and its 20% is used as the evaluation test set with the remaining 80% evenly divided among the clients, leading to every client having access to about 580 images.

B. FEDERATION MODELLING

By the very nature of the proposed consensus-based algorithm FedLCon, the topology of the considered network plays an important role. As pointed out in Section III, the algorithm is developed under the hypothesis of a connected and undirected consensus graph. The network topology that we consider is depicted in Figure 3, which shows that the algorithm is tested on a sparse communication graph over which a limited

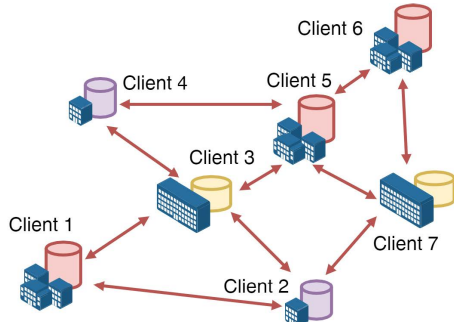


FIGURE 3. Topology considered for the simulations. Clients 3 and 7 (yellow) will be affected by gaussian noise in simulations 2 and 4, whereas clients 2 and 4 (purple) will be subject to an adversarial attack in simulations 3 and 4.

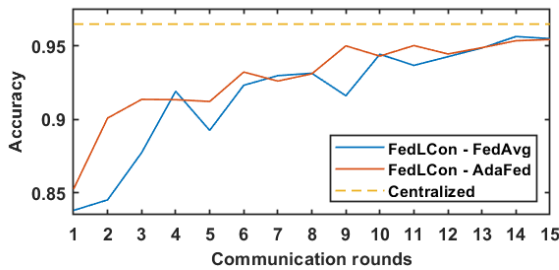


FIGURE 4. First simulation: ideal setting. Accuracy comparison of FedAvg and AdaFed decentralised by FedLCon. The dashed line reports the performance attained after 60 epochs (equivalent to the training epochs done by the federation over 15 communication rounds).

number of information exchanges are allowed. The resulting number of steps for each consensus round is $n_\epsilon = 10$.

C. SIMULATION 1 - ORIGINAL DATA

In this simulation, we want to evaluate whether the federation is able to solve the considered problem in the absence of any training disturbances. In fact, given the relatively low numerosity of the dataset the federated clients may in principle be subject to overfitting. Figures 4 and 5 show that the two proposed algorithms exhibit a similar performance across all the communication rounds, with AdaFed converging slightly faster to the final value. we remark that, in federation with balanced and IID data, AdaFed is expected to perform very similarly to FedAvg, as its dynamic weight update has a greater effect on uneven data distributions [2]. For benchmarking purposes, in Figure 4 we also include a dashed line that represents the performance attained after 60 epochs by a single, centralized, server that trains the same neural network on the entirety of the data. This comparison highlights how the decentralized FL setting shows only a slight performance decrease even when compared to a fully centralized, non federated, solution.

D. SIMULATION 2 - GAUSSIAN NOISE

In this simulation, the presence of two clients with poor quality data is simulated by adding gaussian noise to their

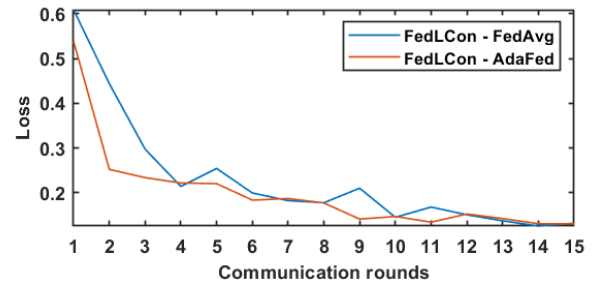


FIGURE 5. First simulation: ideal setting. Training loss comparison of FedAvg and AdaFed decentralised by FedLCon.

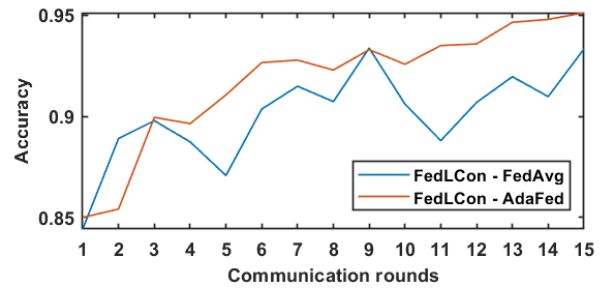


FIGURE 6. Second simulation: low quality data. Accuracy comparison of FedAvg and AdaFed decentralised by FedLCon.

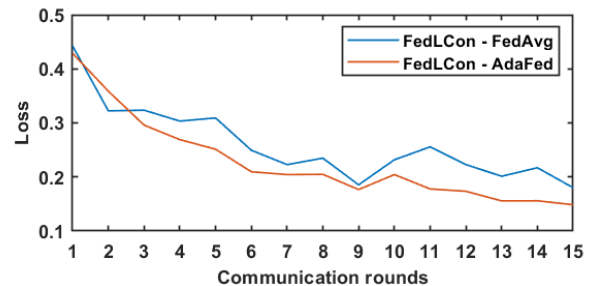


FIGURE 7. Second simulation: low quality data. Training loss comparison of FedAvg and AdaFed decentralised by FedLCon.

images. In particular, we choose to blur the images of clients 1 and 7 by adding a gaussian noise perturbation, with mean $\mu = 0$ and standard deviation $\sigma^2 = 1$, to each of the pixels of their images. Having to deal with perturbed data, AdaFed starts to show its robustness compared to FedAvg, demonstrating a better performance starting from the third communication round, as shown in Figures 7 and 8. This performance advantage is due to AdaFed automatically giving a lower weight, for the model averaging procedure, to the clients with corrupted data (lines 10 and 11 of Algorithm 4), effectively preventing their negative contribution on the federation training.

E. SIMULATION 3 - LABEL SWAP

In this simulation, a malicious attack affecting clients 1 and 7 is introduced by inverting the labels of all their data. From Figure 8 it can be noted how both FedAvg and AdaFed

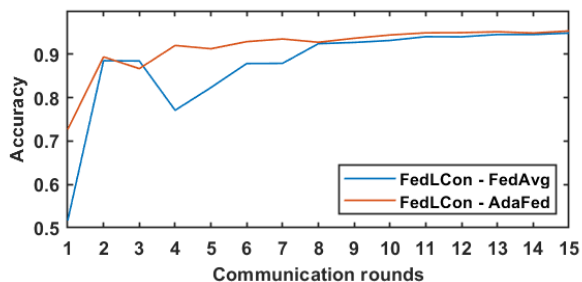


FIGURE 8. Third simulation: malicious clients. Accuracy comparison of FedAvg and AdaFed decentralised by FedLCon.

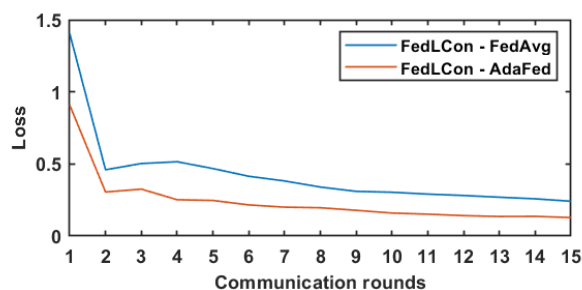


FIGURE 9. Third simulation: malicious clients. Training loss comparison of FedAvg and AdaFed decentralised by FedLCon.

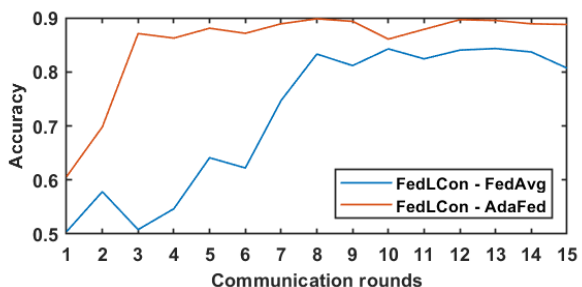


FIGURE 10. Fourth simulation: combined training complications. Accuracy comparison of FedAvg and AdaFed decentralised by FedLCon.

are similarly affected in terms of final accuracy. In fact, the extension of the attack (almost a third of the available data) removes a significant portion of information from the knowledge base available to the federation. Nevertheless, AdaFed demonstrates a better resiliency to this kind of scenario (9), as the corrupted clients are entirely removed from the model averaging procedure.

F. SIMULATION 4 - LABEL SWAP AND GAUSSIAN NOISE

In the final simulation we consider a more complex scenario that combines the previous situations by including both clients with poor quality data (clients 3 and 5) and malicious clients (clients 1 and 7). In this simulation the combination of label swapping and additive gaussian noise significantly lowers the performance of FedAvg, whereas AdaFed limits its performance degradation, as depicted in Figures 10 and 11.

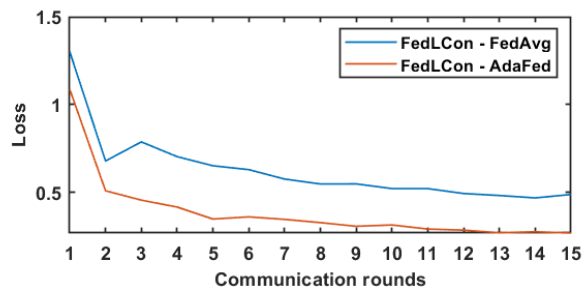


FIGURE 11. Fourth simulation: combined training complications. Training loss comparison of FedAvg and AdaFed decentralised by FedLCon.

VI. CONCLUSION AND FUTURE WORKS

This paper presents a FL consensus-based paradigm called FedLCon, originating on the ground of results from discrete-time average consensus theory, that is used to decentralize two FL algorithms - FedAvg and AdaFed. Decentralizing existing FL algorithms thought a solution such as FedLCon enables the application of FL over federations with sparse communications graphs, further enhancing its privacy-related features. The paper presented the results attained by the tested algorithms on several scenarios for a COVID-19 detection task.

ACKNOWLEDGMENT

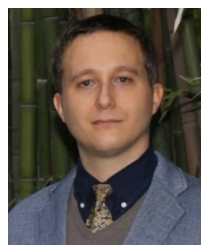
The authors would like to thank the colleagues from “La Sapienza” and the “Istituto Superiore di Sanità” involved in the FedMedAI Project.

REFERENCES

- [1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, “Communication-efficient learning of deep networks from decentralized data,” 2016, *arXiv:1602.05629*.
- [2] A. Giuseppi, L. D. Torre, D. Menegatti, and A. Pietrabissa, “AdaFed: Performance-based adaptive federated learning,” in *Proc. 5th Int. Conf. Adv. Artif. Intell. (ICAAI)*, Nov. 2021, pp. 38–43, doi: 10.1145/3505711.3505717.
- [3] A. Giuseppi, S. Manfredi, D. Menegatti, A. Pietrabissa, and C. Poli, “Decentralized federated learning for nonintrusive load monitoring in smart energy communities,” in *Proc. 30th Medit. Conf. Control Autom. (MED)*, Jun. 2022, pp. 312–317.
- [4] A. Giuseppi, S. Manfredi, and A. Pietrabissa, “A weighted average consensus approach for decentralized federated learning,” *Mach. Intell. Res.*, vol. 19, no. 4, pp. 319–330, Aug. 2022.
- [5] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen, and S. Bakas, “Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data,” *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, Jul. 2020, doi: 10.1038/s41598-020-69250-1.
- [6] D. C. Nguyen, Q.-V. Pham, P. N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. A. Dobre, and W.-J. Hwang, “Federated learning for smart healthcare: A survey,” 2021, *arXiv:2111.08834*.
- [7] T. D. Gunter and N. P. Terry, “The emergence of national electronic health record architectures in the United States and australia: Models, costs, and questions,” *J. Med. Internet Res.*, vol. 7, no. 1, p. e3, Mar. 2005, doi: 10.2196/jmir.7.1.e3.
- [8] M. Hao, H. Li, G. Xu, Z. Liu, and Z. Chen, “Privacy-aware and resource-saving collaborative learning for healthcare in cloud computing,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.
- [9] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, “Split learning for health: Distributed deep learning without sharing raw patient data,” 2018, *arXiv:1812.00564*.

- [10] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *Int. J. Med. Informat.*, vol. 112, pp. 59–67, Apr. 2018, doi: [10.1016/j.ijmedinf.2018.01.007](https://doi.org/10.1016/j.ijmedinf.2018.01.007).
- [11] O. Choudhury, A. Gkoulalas-Divanis, T. Salonidis, I. Sylla, Y. Park, G. Hsu, and A. Das, "Differential privacy-enabled federated learning for sensitive health data," 2019, *arXiv:1910.02578*.
- [12] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farhad, S. Jin, T. Q. S. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," 2019, *arXiv:1911.00222*.
- [13] D. Liu, T. Miller, R. Sayeed, and K. D. Mandl, "FADL: Federated-autonomous deep learning for distributed electronic health record," 2018, *arXiv:1811.11400*.
- [14] R. Shao, H. He, H. Liu, and D. Liu, "Stochastic channel-based federated learning for medical data privacy preserving," 2019, *arXiv:1910.11160*.
- [15] L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng, and D. Liu, "Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records," *J. Biomed. Informat.*, vol. 99, Nov. 2019, Art. no. 103291, doi: [10.1016/j.jbi.2019.103291](https://doi.org/10.1016/j.jbi.2019.103291).
- [16] S. Boughorbel, F. Jarray, N. Venugopal, S. Moosa, H. Elhadi, and M. Makhlouf, "Federated uncertainty-aware learning for distributed hospital EHR data," 2019, *arXiv:1910.12191*.
- [17] Y. Chen, J. Wang, C. Yu, W. Gao, and X. Qin, "FedHealth: A federated transfer learning framework for wearable healthcare," 2019, *arXiv:1907.09173*.
- [18] G. K. Gudur and S. K. Perepu, "Federated learning with heterogeneous labels and models for mobile activity monitoring," 2020, *arXiv:2012.02539*.
- [19] X. Xu, H. Peng, L. Sun, M. Z. A. Bhuiyan, L. Liu, and L. He, "FedMood: Federated learning on mobile health data for mood detection," 2021, *arXiv:2102.09342*.
- [20] W. Li, F. Milletari, D. Xu, N. Rieke, J. Hancox, W. Zhu, M. Baust, Y. Cheng, S. Ourselin, M. Jorge Cardoso, and A. Feng, "Privacy-preserving federated brain tumour segmentation," 2019, *arXiv:1910.00962*.
- [21] S. Bakas et al., "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge," 2018, *arXiv:1811.02629*.
- [22] S. Silva, B. A. Gutman, E. Romero, P. M. Thompson, A. Altmann, and M. Lorenzi, "Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data," in *Proc. IEEE 16th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2019, pp. 270–274.
- [23] P. Guo, P. Wang, J. Zhou, S. Jiang, and V. M. Patel, "Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 2423–2432.
- [24] X. Li, Y. Gu, N. Dvornek, L. Staib, P. Ventola, and J. S. Duncan, "Multi-site fMRI analysis using privacy-preserving federated learning and domain adaptation: ABIDE results," 2020, *arXiv:2001.05647*.
- [25] H. R. Roth, K. Chang, P. Singh, N. Neumark, W. Li, V. Gupta, S. Gupta, L. Qu, A. Ihsani, B. C. Bizzo, and Y. Wen, "Federated learning for breast density classification: A real-world implementation," in *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*. Cham, Switzerland: Springer, 2020, pp. 181–191.
- [26] D. A. Spak, J. S. Plaxco, L. Santiago, M. J. Dryden, and B. E. Dogan, "BI-RADS fifth edition: A summary of changes," *Diagnostic Interventional Imag.*, vol. 98, no. 3, pp. 179–190, Mar. 2017, doi: [10.1016/j.diii.2017.01.001](https://doi.org/10.1016/j.diii.2017.01.001).
- [27] C. Zheng, X. Deng, Q. Fu, Q. Zhou, J. Feng, H. Ma, W. Liu, and X. Wang, "Deep learning-based detection for COVID-19 from chest ct using weak label," *MedRxiv*, 2020.
- [28] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," 2015, *arXiv:1505.04597*.
- [29] O. Gozes, M. Frid-Adar, N. Sagie, H. Zhang, W. Ji, and H. Greenspan, "Coronavirus detection and analysis on chest CT with deep learning," 2020, *arXiv:2004.02640*.
- [30] M. Barstugan, U. Ozkaya, and S. Ozturk, "Coronavirus (COVID-19) classification using CT images by machine learning methods," 2020, *arXiv:2003.09424*.
- [31] O. Gozes, M. Frid-Adar, H. Greenspan, P. D. Browning, H. Zhang, W. Ji, A. Bernheim, and E. Siegel, "Rapid AI development cycle for the coronavirus (COVID-19) pandemic: Initial results for automated detection & patient monitoring using deep learning CT image analysis," 2020, *arXiv:2003.05037*.
- [32] W.-C. Dai, H.-W. Zhang, J. Yu, H.-J. Xu, H. Chen, S.-P. Luo, H. Zhang, L.-H. Liang, X.-L. Wu, Y. Lei, and F. Lin, "CT imaging and differential diagnosis of COVID-19," *Can. Assoc. Radiologists J.*, vol. 71, no. 2, pp. 195–200, Mar. 2020, doi: [10.1177/0846537120913033](https://doi.org/10.1177/0846537120913033).
- [33] F. Shan, Y. Gao, J. Wang, W. Shi, N. Shi, M. Han, Z. Xue, D. Shen, and Y. Shi, "Lung infection quantification of COVID-19 in CT images with deep learning," 2020, *arXiv:2003.04655*.
- [34] L. Wang, Z. Q. Lin, and A. Wong, "COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images," *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, Dec. 2020.
- [35] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network," *Int. J. Speech Technol.*, vol. 51, no. 2, pp. 854–864, Feb. 2021.
- [36] N. Eldeen M. Khalifa, M. Hamed N. Taha, A. E. Hassanien, and S. Elghamrawy, "Detection of coronavirus (COVID-19) associated pneumonia based on generative adversarial networks and a fine-tuned deep transfer learning model using chest X-ray dataset," 2020, *arXiv:2004.01184*.
- [37] I. D. Apostolopoulos and T. Bessiana, "COVID-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks," *Phys. Eng. Sci. Med.*, vol. 43, no. 2, pp. 635–640, 2020.
- [38] A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks," *Pattern Anal. Appl.*, vol. 24, no. 3, pp. 1207–1220, 2021.
- [39] I. Feki, S. Ammar, Y. Kessentini, and K. Muhammad, "Federated learning for COVID-19 screening from chest X-ray images," *Appl. Soft Comput.*, vol. 106, Jul. 2021, Art. no. 107330.
- [40] B. Liu, B. Yan, Y. Zhou, Y. Yang, and Y. Zhang, "Experiments of federated learning for COVID-19 chest X-ray images," *Tech. Rep.*, 2020.
- [41] R. Kumar, A. A. Khan, J. Kumar, Zakria, N. A. Golilarz, S. Zhang, Y. Ting, C. Zheng, and W. Wang, "Blockchain-federated-learning and deep learning models for COVID-19 detection using CT imaging," *IEEE Sensors J.*, vol. 21, no. 14, pp. 16301–16314, Jul. 2021.
- [42] I. Dayan et al., "Federated learning for predicting clinical outcomes in patients with COVID-19," *Nature Med.*, vol. 27, no. 10, pp. 1735–1743, 2021, doi: [10.1038/s41591-021-01506-3](https://doi.org/10.1038/s41591-021-01506-3).
- [43] W. Zhang, T. Zhou, Q. Lu, X. Wang, C. Zhu, H. Sun, Z. Wang, S. K. Lo, and F.-Y. Wang, "Dynamic-fusion-based federated learning for COVID-19 detection," *IEEE Internet Things J.*, vol. 8, no. 21, pp. 15884–15891, Nov. 2021, doi: [10.1109/IIOT.2021.3056185](https://doi.org/10.1109/IIOT.2021.3056185).
- [44] A. Vaid et al., "Federated learning of electronic health records to improve mortality prediction in hospitalized patients with COVID-19: Machine learning approach (preprint)," *Tech. Rep.*, Sep. 2020, doi: [10.2196/preprints.24207](https://doi.org/10.2196/preprints.24207).
- [45] Z. Hu, Q. Ge, S. Li, E. Boerwinkle, L. Jin, and M. Xiong, "Forecasting and evaluating intervention of covid-19 in the world," 2020, *arXiv:2003.09800*.
- [46] B. R. Beck, B. Shin, Y. Choi, S. Park, and K. Kang, "Predicting commercially available antiviral drugs that may act on the novel coronavirus (SARS-CoV-2) through a drug-target interaction deep learning model," *Comput. Struct. Biotechnol. J.*, vol. 18, pp. 784–790, 2020.
- [47] N. Savioli, "One-shot screening of potential peptide ligands on HR1 domain in COVID-19 glycosylated spike (S) protein with deep Siamese network," 2020, *arXiv:2004.02136*.
- [48] V. Chenthamarakshan, P. Das, S. C. Hoffman, H. Strobelt, I. Padhi, K. W. Lim, B. Hoover, M. Manica, J. Born, T. Laino, and A. Mojsilovic, "CogMol: Target-specific and selective drug design for COVID-19 using deep generative models," 2020, *arXiv:2004.01215*.
- [49] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, S. Qin, A. Židek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. T. Jones, D. Silver, K. Kavukcuoglu, and D. Hassabis, "Improved protein structure prediction using potentials from deep learning," *Nature*, vol. 577, no. 7792, pp. 706–710, Jan. 2020, doi: [10.1038/s41586-019-1923-7](https://doi.org/10.1038/s41586-019-1923-7).
- [50] A. Strokach, D. Becerra, C. Corbi-Verge, A. Perez-Riba, and P. M. Kim, "Fast and flexible protein design using deep graph neural networks," *Cell Syst.*, vol. 11, no. 4, pp. 402–411, Oct. 2020, doi: [10.1016/j.cels.2020.08.016](https://doi.org/10.1016/j.cels.2020.08.016).

- [51] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.
- [52] F. P. Sánchez, M. R. Pedruelo, C. C. Casamayor, and A. P. Chust, "Convergence of weighted-average consensus for undirected graphs," *J. Complex Syst. Sci.*, vol. 4, no. 1, pp. 13–16, 2014.
- [53] K. Ogata, *Discrete-Time Control Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 1995.
- [54] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," 2019, *arXiv:1907.09693*.
- [55] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2018, *arXiv:1812.06127*.
- [56] S. Bozinovski, "Reminder of the first paper on transfer learning in neural networks, 1976," *Informatica*, vol. 44, no. 3, Sep. 2020, doi: 10.31449/inf.v44i3.2828.
- [57] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [58] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2009, pp. 248–255.
- [59] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," 2018, *arXiv:1803.08375*.
- [60] A. Asraf. (2021). *COVID19, Pneumonia and Normal Chest X-Ray PA Dataset*. [Online]. Available: <https://data.mendeley.com/datasets/jctsfj2sfm/1>



ALESSANDRO GIUSEPPI (Member, IEEE) received the master's degree in control engineering and the Ph.D. degree in automatica from the Department of Computer, Control, and Management Engineering "Antonio Ruberti" (DIAG), University of Rome "La Sapienza", in 2016 and 2019, respectively. He is currently an Assistant Professor (RTDa) with the Department of Computer, Control, and Management Engineering "Antonio Ruberti" (DIAG), University of Rome

"La Sapienza". Since 2016, he has participated in six other EU and National Research Projects, mainly in the fields of control systems and artificial intelligence. He is also the Scientific Coordinator of the ESA-Funded Research Project ARIES, related to Wildfire Emergency Management and Work Package Leader at the EU-Korea H2020 Project 5G-ALLSTAR. Since 2020, he is serving as an Associate Editor for the *International Journal of Control, Automation and Systems* (Springer). His main research interests include network control and intelligent systems, where he published about 50 papers in international journals and conferences.



SABATO MANFREDI was a Visiting Professor at the School of Mathematical Sciences, Queen Mary, London, U.K., from 2017 to 2018. He is currently an Associate Professor of automatic control with the Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy. He has been a Visiting Academic with the Control and Power Group, Electrical and Electronic Engineering Department, Imperial College London,

London, since 2012. He has authored/coauthored more than 100 scientific publications including 18 single-author papers and the monograph: *Multilayer Control of Networked Cyber-Physical Systems: Application to Monitoring, Autonomous and Robot Systems* (Advances in Industrial Control Series, Springer, 2017). He has collaborates with many international universities and companies and holds European patent. He is also a Proponent member of an Academic Spin-Off. He is also involved in a range of academic, industrial, and consulting projects. His research interests include automatic control with a special emphasis on nonlinear and complex networks, distributed control and optimization, sensor/drone networks, and new technologies/algorithms for smart city and cyber-physical systems.

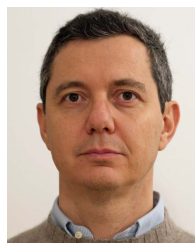


DANILO MENEGATTI (Student Member, IEEE) received the master's degree in control engineering from the Department of Computer, Control, and Management Engineering "Antonio Ruberti" (DIAG), University of Rome "La Sapienza," in 2020, where he is currently pursuing the Ph.D. degree in automatic control, bioengineering and operations research. His research interests include intelligent systems, distributed learning, and reinforcement learning applications.



CECILIA POLI received the M.Sc. degree in electronic engineering and the Ph.D. degree in thermal physics from the University of Rome Sapienza, in 2002 and 2006, respectively. She is currently a Researcher with the National Centre for Innovative Technologies in Public Health Department, Istituto Superiore di Sanità. She is a Regulatory Expert in European Medical Device Regulations (Reg. 745/2017, Reg. 746/2017), with particular attention to medical device software. She is also

involved in the Project FedMedAI on federated learning for biomedical applications.



ANTONIO PIETRABISSA (Senior Member, IEEE) received the degree in electronics engineering and the Ph.D. degree in systems engineering from the Department of Computer, Control, and Management Engineering "Antonio Ruberti" (DIAG), University of Rome Sapienza, in 2000 and 2004, respectively. He is currently an Associate Professor with the Department of Computer, Control, and Management Engineering "Antonio Ruberti" (DIAG), University of Rome

Sapienza, where he teaches automatic control and process automation. Since 2000, he has participated in about 25 EU and National Research Projects. He is currently the Coordinator of the Project ARIES on fire emergency prevention, funded by ESA; the Scientific responsible of the Research Projects 5GALLSTARS on 5G Communications, funded within the H2020 Europe-South Korea Cooperation; and the FedMedAI on medical applications of federated learning, funded by Regione Lazio (IT). He serves as an Associate Editor for *Control Engineering Practice* (Elsevier) and for IEEE Transactions on Automation Science and Engineering. He is the author of more than 50 journal articles and 80 conference papers. His research interests include the application of systems and control theory to the analysis and control of networks.

...