DeceFL: A Principled Decentralized Federated Learning Framework

| Journal               | National Science Open |
|-----------------------|-----------------------|
| Manuscript ID         | NSO20220046.R1        |
| Manuscript Type       | Research Article      |
| Date Submitted by the Author | 05-Oct-2022           |
| Complete List of Authors | Yuan, Ye; Huazhong University of Science and Technology |
| Keywords              | Decentralized Federated Learning, Smart Manufacturing, Control Systems Privacy |

Note: The following files were submitted by the author for peer review, but cannot be converted to PDF. You must view these files (e.g. movies) online.

- NSO_MT_final.zip
- NSO_SI_final.zip
DeceFL: A Principled Fully Decentralized Federated Learning Framework

Ye Yuan\textsuperscript{1,2,*}, Jun Liu\textsuperscript{3,*}, Dou Jin\textsuperscript{1,*}, Zuogong Yue\textsuperscript{1,*}, Tao Yang\textsuperscript{4,*}, Ruijuan Chen\textsuperscript{5}, Maolin Wang\textsuperscript{1}, Chuan Sun\textsuperscript{1}, Lei Xu\textsuperscript{4}, Feng Hua\textsuperscript{2}, Yuqi Guo\textsuperscript{1}, Xiuchuan Tang\textsuperscript{2}, Xin He\textsuperscript{2}, Xinlei Yi\textsuperscript{6}, Dong Li\textsuperscript{7}, Guanghui Wen\textsuperscript{8}, Wenwu Yu\textsuperscript{8,9}, Hai-Tao Zhang\textsuperscript{1,2}, Tianyou Chai\textsuperscript{8}, Shaochun Sui\textsuperscript{10}, Han Ding\textsuperscript{2}

Abstract

Traditional machine learning relies on a centralized data pipeline for model training, in various applications, however, data are inherently fragmented. Such a decentralized nature of databases presents the serious challenge for collaboration: sending all decentralized datasets to a central server raises serious privacy concerns. Although there has been a joint effort in tackling such a critical issue by proposing privacy-preserving machine learning frameworks, such as federated learning, most state-of-the-art frameworks are built still in a centralized way, in which a central client is needed for collecting and distributing model information (instead of data itself) from every other client, leading to high communication burden and high vulnerability when there exists a failure at or an attack on the central client. Here we propose a principled decentralized federated learning algorithm (DeceFL), which does not require a central client and relies only on local information transmission between clients and their neighbors, representing a fully decentralized learning framework. It has been further proven that every client reaches the global minimum with zero performance gap and achieves the same convergence rate $O(1/T)$ (where $T$ is the number of iterations in gradient descent) as centralized federated learning when the loss function is smooth and strongly convex. Finally, the proposed algorithm has been applied to a number of applications to illustrate its effectiveness for both convex and nonconvex loss functions, time-invariant and time-varying topologies, as well as IID and Non-IID of datasets, demonstrating its applicability to a wide range of real-world medical and industrial applications.

\textsuperscript{1}School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. \textsuperscript{2}School of Mechanical Science and Engineering, Huazhong University of Science and Technology. \textsuperscript{3}Department of Applied Mathematics, University of Waterloo. \textsuperscript{4}State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University. \textsuperscript{5}School of Mathematical and Physical Sciences, Wuhan Textile University. \textsuperscript{6}School of Electrical Engineering and Computer Science, and Digital Futures, KTH Royal Institute of Technology. \textsuperscript{7}Shenyang Institute of Automation (SIA), Chinese Academy of Sciences. \textsuperscript{8}School of Mathematics, Frontiers Science Center for Mobile Information Communication and Security, Southeast University. \textsuperscript{9}Purple Mountain Laboratories. \textsuperscript{10}AVIC Chengdu Aircraft Industrial (Group) Co., Ltd.. *Equal contributions. For correspondence: yye@hust.edu.cn.
An urgent challenge for Artificial Intelligence (AI) applications today consists of the following dilemma concerning data privacy: on the one hand, a large number of sophisticated algorithms have been proposed to broaden the applicability of AI to various applications such as medicine, manufacturing and more [1]–[3]; on the other hand, regulations such as General Data Protection Regulation (GDPR) restrict data sharing, thus limiting the performance of AI algorithms [4]. As a result, models that are trained and evaluated on a limited amount of data since privacy could have biases [5]. This has become a well-known bottleneck in AI applications, especially medical AI [6].

Promising privacy-preserving methods such as federated learning can help maintain the performance of AI algorithms, while preserving the data stored locally [7]. Inspired by this, there has been a surge of interests in both the theory and applications of federated learning [8]. Federated averaging (FedAvg), the leading algorithm in the field of federated learning, was proposed in 2016 by researchers at Google [9], [10], and its variant, dynamic FedAvg, was proposed in [11]. The works [9]–[11] mentioned above were based on the analysis of first-order stationary points. The second-order optimality guarantee of federated learning was later established in [12]. Through crowded efforts and comprehensive surveys [13]–[18], widely used federated learning methods were established, the challenges and related applications were introduced, and a large number of valuable research directions were outlined. A notable example has been demonstrated in a report by Kaissis et al. [19], in which a convolutional neural network was trained over the public Internet with encryption from medical images using a secure federated learning framework.

Despite these breakthroughs, classical federated learning algorithms have a major drawback: the need for a central client, which could cause privacy, communication, computation, and resilience issues [20]. Much effort, therefore, has been invested to reduce the communication and computation complexity as well as increase the robustness and resilience of centralized federated learning algorithms. In order to deal with the challenge of system constraints, the authors of [21] applied sparse technology to reduce the communication and computation costs in the training process. In addition, the authors of [22] proposed an optimal trade-off control algorithm between the local update and the global parameter aggregation in the resource constrained system. Recently, federated schemes were extended to a time-varying centralized scheme, where a changing leader is selected based on certain rules, which is a firm step to full decentralization [23].

The current research works mainly discussed decentralized federated learning from three
different perspectives. Firstly, the initial work [24] realized fully asynchronous and decentralized machine learning by introducing gossip protocol into deep learning. Furthermore, Daily et al. [25] proposed a random gradient descent algorithm based on GossipGraD communication protocol for extended deep learning algorithms on large-scale systems. In addition, Hu et al. [26] proposed a decentralized federated learning system based on model segmented gossip aggregation. Secondly, being specialized for graph convolutional networks, Scardapane et al. [27] proposed an algorithmic framework for distributed training considering the case that data was collected by a group of agents communicating through the sparse network topology. Similarly, Pei et al. [28] proposed a new decentralized federated graph neural network model based on peer-to-peer network structure and realized the decentralized learning with privacy protection based on the Diffie-Hellman key exchange method. For decentralized federated learning in general, similar to [21] and [22], the works [29] and [30] proposed a decentralized distributed learning algorithm by introducing the concept of belief into the parameter space of the client model in which the client updates its belief by aggregating the information from one-hop neighbors to realize the optimal model of cooperative learning. A fully decentralized federated learning algorithm, IPLS, was proposed by Pappas et al. [31], which is based on model segmentation and partially based on the interplanetary file system. Based on the popular Time Slotted Channel Hopping radio interface of the IEEE 802.15.4e standard, Savazzi et al. [32] proposed a real-time framework for analyzing decentralized federated learning systems running on industrial wireless networks. Thirdly, in terms of decentralization via blockchain, similar to [23], Wu et al. [33] proposed a decentralized federated learning framework that realized the decentralization of all aspects of system. The work [34] proposed a decentralized privacy protection in-depth learning framework based on blockchain, which uses the local credibility mutual evaluation mechanism to realize the research on collaborative fairness in in-depth learning. Yapp et al. [35] proposed a federated edge learning (FEL) framework with efficient blockchain authorization and communication, which enhances the security and scalability of FEL's large-scale implementation.

However, these methods suffer from one of the following drawbacks: 1) lacks of theoretical guarantee, making it unreliable for practical use; 2) lacks of temporal flexibility of communication topology; and 3) suffers from potential leakage of private information. More specifically, the comparison between these existing works and the decentralized framework method proposed in this paper is shown in Supplementary Information Table 1. This paper proposes a principled fully decentralized federated learning algorithm (DeceFL), in which each client is guaranteed to achieve the same performance as the centralized algorithm in terms of training/test accuracy,
when the global objective function is smooth and strongly convex. Empirical experiments have demonstrated that the same claim also holds for nonconvex global objective functions, thus revealing its potential to be applied to a wider class of applications such as those using deep learning. In addition to all features that other state-of-the-art federated learning and swarm learning algorithms [23] possess, DeceFL has additional desirable features beyond classical centralized or decentralized federated learning and swarm learning, namely: 1) full decentralization: at any iteration, there is no central client that receives all other clients’ information, therefore removing monopoly and preventing attacks; 2) principled algorithms: it has been proved that zero performance gap between decentralized and centralized algorithms can be achieved under certain condition, therefore guaranteeing global performance; 3) flexible topology design: any connected time-varying communication graph suffices to achieve the same performance, therefore allowing clients to decide when and who to communicate with subject to their physical constraints; 4) equal rights for all: all clients can have the trained model, therefore incentivizing clients to participate; 5) privacy-preserving: every client does not communicate its gradient to other client, therefore preventing data leakage, even there is adversary in the communication channel [36].

THE PROPOSED FULLY DECENTRALIZED FEDERATED LEARNING FRAMEWORK

A. Problem formulation

We first formulate the decentralized federated learning problem. Assume that there are K clients with local data in the form for standard machine learning tasks: D_k for k ∈ {1, 2, . . ., K}. The conventional training of AI models (denoted by M) can be formulated as the following global learning problem (let D ≜ ∪_k D_k and ∩_k D_k = ∅):

\[ M_c ≜ \arg \min_M F(D; M). \]

In such a centralized framework, clients have to send their datasets D_k to a central client. In contrast, in a decentralized framework, clients would like to collaboratively train a model M_d using the same objective function F, in which the k-th client does not send its data D_k to others. We define the performance gap as a nonnegative metric, which quantifies the degenerative performance between a centralized model and a decentralized one:

\[ \Delta ≜ F(D; M_c) - F(D; M_d). \]

The goal is to make \( \Delta \) as small as possible and in the ideal case to make \( \Delta = 0 \).
B. The proposed DeceFL algorithm

To solve the optimization problem in a decentralized way, we model the communication network between clients as an undirected connected 1 graph $G = (\mathcal{N}, \mathcal{E}, W)$, where $\mathcal{N} := \{1, 2, \ldots, K\}$ represents the set of clients, and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ represents the set of communication channels, each connecting two distinct clients. For each pair $(i, j)$, the corresponding element in the adjacency matrix $W$, i.e., $W_{ij}$, indicates whether there is a communication channel between the $i$-th client and the $j$-th client. Specifically, when $W_{ij} > 0$, there is information communication between clients $i$ and $j$, while $W_{ij} = 0$ means none. For client $i$, when $W_{ij} > 0$, then client $j$ is called a neighbor of client $i$. The set of all such clients $j$ is represented as $\mathcal{N}_i$, i.e., $\mathcal{N}_i = \{j|W_{ij} > 0, \forall j \in \mathcal{N}\}$. Define the local loss function $F_k(w) \triangleq F(D_k; M)$ as the user-specified loss function on the dataset $D_k$ with model parameters $w$ in $M$, then $F(D; M)$ can be rewritten as $F(w) \triangleq F(D; M) = \frac{1}{K} \sum_{k=1}^{K} F_k(w)$. Let the client $k$ hold a local copy of the global variable $w$, which is denoted by $w_k \in \mathbb{R}^n$. Specifically, the update rule of DeceFL is, for each client $k = 1, \ldots, K$,

$$w_k(t+1) = \frac{\sum_{j=1}^{K} W_{kj} w_j(t)}{\text{average of neighbors' estimates}} - \eta_t \nabla F_k(w_k(t)),$$

where $\eta_t > 0$ is the learning rate, and the initial condition $w_k(0) \in \mathbb{R}^n$ can be arbitrarily chosen. Every client shares with its neighbors (which is a subset of all other clients) its model parameters rather than data. Specifically, every client runs its local training algorithm, e.g., gradient descent, and it only communicates its own estimate of the global parameter with its neighbors. Once a client receives other estimates from neighboring clients, it averages out other estimates, adds to its local gradient and generates its estimate in the next iteration. The above process will be repeated until convergence. As shown in Fig. 1, in DeceFL, each client completes the update by receiving and transmitting directly with neighboring clients and local gradient calculation, without needing the aggregation and transmission of a third-party central client at any iteration. Thus, it is fully decentralized.

We stack the $w_k(t)$ and $\nabla F_k(w_k(t))$ in (1) into vectors, i.e., define $w(t) = [w_1(t)^T, \ldots, w_K(t)^T]^T \in \mathbb{R}^{Kn}$ and $\nabla F(w(t)) = [\nabla F_1(w_1(t))^T, \ldots, \nabla F_K(w_K(t))^T]^T \in \mathbb{R}^{Kn}$. Then, we can compactly rewrite

1In this work, we consider that the information communication between clients is mutual for notational simplicity; therefore, the adjacency matrix $W = [W_{ij}] \in \mathbb{R}^{K \times K}$ is symmetric. Furthermore we assume that the underlying topology is connected, i.e., for any two clients $k$ and $j$, there is at least one path from $k$ to $j$.}

October 28, 2022 DRAFT

https://engine.scichina.com/doi/10.1360/nso/20220043
Fig. 1. Illustration of key concepts in different state-of-the-art federated learning frameworks. a) Classical Federated Learning: a central client is needed to receive and transmit all essential information to other clients. It is equivalent to an all-to-all network without such a central center, i.e., every client in the network can receive information from all other clients. b) Swarm Learning: there is no such a universal central client, but a potentially different central client is selected in every iteration. Mathematically, it is equivalent to FedAvg with varying central clients. c) The proposed Decentralized Federated Learning: there is no need for a central client in any iteration. Any connected time-invariant or time-varying topology (under certain mild condition given in the Supplementary Information) has guaranteed performance, therefore unifying the classical federated learning and swarm learning.

(1) as
\[ \mathbf{w}(t + 1) = (W \otimes \mathbf{I}_n)\mathbf{w}(t) - \eta_t \nabla F(\mathbf{w}(t)), \]
where \( W = [W_{ij}] \in \mathbb{R}^{K \times K} \) and \( \mathbf{I}_n \in \mathbb{R}^{n \times n} \) is the identity matrix. Next, we analyze the convergence of DeceFL under the following assumption about the global cost function, which is consistent with those made in the convergence analysis of FedAvg [37].

**Assumption 1:** For each \( k = 1, \ldots, K \), assume that \( F_k \) is \( L_k \)-smooth and \( \mu_k \)-strongly convex, where \( L_k, \mu_k > 0 \). That is, \( F_k \) is differentiable and the gradient is \( L_k \)-Lipschitz continuous, i.e., for any \( x, y \in \mathbb{R}^n \),
\[ \| \nabla F_k(x) - \nabla F_k(y) \| \leq L_k \| x - y \|, \]
and
\[ F_k(x) \geq F_k(y) + \langle \nabla F_k(y), x - y \rangle + \frac{\mu_k}{2} \| x - y \|^2. \] (4)

When Assumption 1 holds, the global objective function \( F(\cdot) \) is \( L \)-smooth and \( \mu \)-strongly convex, where \( L = \max\{L_1, \ldots, L_K\} \) and \( \mu = \min\{\mu_1, \ldots, \mu_K\} \). Clearly, \( \mu \leq L \).

Assumption 1 is standard and satisfied by typical loss functions in machine learning, including \( l_2 \)-regularized linear and logistic regressions. In order to analyze the convergence of the algorithm, we define the average sequence \( \bar{w}(t) = \frac{1}{K} (1_K \otimes I_n) w(t) = \frac{1}{K} \sum_{k=1}^{K} w_k(t) \), where \( 1_K \in \mathbb{R}^K \) is a vector in which all elements are 1. Denote \( \lambda \) as the spectral norm of \( W - \frac{1}{K} 11^T \), where \( \lambda \in (0, 1) \) in algebraic graph theory (see Supplementary Information).

**Theorem 1:** Consider algorithm (1), where the learning rate are chosen by \( \eta_t = \frac{\delta}{t+\Gamma} \), in which \( \delta > \frac{1}{\mu} \) and \( \Gamma > \frac{\lambda}{1-\lambda} \) satisfying \( \frac{\delta}{\Gamma} \leq \frac{1}{T} \). Denote the gap between the local function value at the initial point \( w(0) \) and local optimal value as \( \varepsilon_0 = \sum_{k=1}^{K} (F_k(w_k(0)) - F_k(w_k^*)) \geq 0 \), where \( w_k^* = \arg\min_{w_k} F_k(w_k) \). Then the following inequality can be obtained under Assumption 1:
\[ \| w(t) - 1 \otimes \bar{w}(t) \| \leq \frac{\zeta}{t+\Gamma}, \] (5)
and
\[ \| \bar{w}(t) - w^* \| \leq \frac{\tilde{\zeta}}{t+\Gamma}, \] (6)
where \( \zeta = \max \left\{ \Gamma \| w(0) - 1 \bar{w}(0) \|, \frac{\delta \sqrt{2L_0}}{1-\lambda}, \frac{\sqrt{2L_0}}{1-\lambda} \right\} \) and \( \tilde{\zeta} = \max \left\{ \Gamma \| \bar{w}(0) - w^* \|, \frac{1}{\mu \delta - 1} \frac{L \delta \zeta}{\sqrt{K}} \right\} \).

**Proof:** See Supplementary Information.

In [38], the authors studied the convergence of FedAvg and established the convergence rate \( O(1/T) \) (where \( T \) is the number of iterations in gradient descent) for strongly convex and smooth problems. Here Theorem 1 guarantees the proposed DeceFL algorithm has the same convergence rate as FedAvg under time-invariant connected communication topologies. In addition, in the Supplementary Information, it has been further demonstrated that the proposed DeceFL converges even for time-varying topologies, in which case the underlying topology does not need to be connected for all iterations (as illustrated in Figure 1). Finally, FedAvg and SL can be viewed as special cases of DeceFL, the proof of convergence for DeceFL also warrants that of centralized federated learning and SL.

**C. An illustrative example**

We use a simple yet important example to illustrate the problem and demonstrate the advantages of the proposed decentralized framework as compared with centralized learning,
federated learning and swarm learning. Consider the case where \( K \) clients would like to compute the average of every client’s private value \( D_k = w_k(0) \), i.e., \( w^* \triangleq \frac{\sum_{k=1}^{K} w_k(0)}{K} \). This task is simple if there exists a central client. However, it becomes challenging when a central client is not available. This is the well-known decentralized consensus problem [39]–[41], which has many engineering applications including synchronization, PageRank, state estimation, load balancing, and more.

We can convert the consensus problem to the following optimization problem

\[
\min_{w \in \mathbb{R}} F(D; w) \triangleq \frac{1}{2} \sum_{k=1}^{K} (w - w_k(0))^2,
\]

with its optimal value being coincided with \( w^* \). Rather than computing the mean, we convert it to solve the optimization problem (7) using algorithms FedAvg, SL and the proposed DeceFL respectively.

1) **FedAvg algorithm:** In the classical federated learning algorithm, i.e., FedAvg, there is a central server to collect the local parameter information of each client for average aggregation, and to assign it to each client, which is equivalent to a complete graph of \( K \) clients without a central server after simple derivation (Fig. 1a):

\[
w_k(t+1) = \frac{1}{K} \sum_{k=1}^{K} w_k(t) - \eta_t (w_k(t) - w_k(0)),
\]

where \( t \) represents the \( t \)-th iteration, \( w_k(t) \) represents the estimate of global optimum for the \( k \)-th client at iteration \( t \), and \( \eta_t \) is learning rate in the gradient descent algorithm. In essence, every client iteratively updates its estimate based on all others’ estimates together with its current gradient. Using derivation in the Supplementary Information based on dynamical system and algebraic graph theory, it can be shown that the system reaches the steady-state, i.e., \( \lim_{t \to \infty} ||w_k(t) - w^*|| = 0 \), if \( \eta_t = \frac{\gamma}{t+1} \) and \( \frac{\gamma}{\Gamma} < 1 \) is satisfied for any \( \gamma, \Gamma > 0 \).

2) **SL algorithm:** The SL algorithm considers the situation where there is no central server: in each iteration, a random leader is dynamically selected from the clients to aggregate the model parameters from all clients (including itself) and assign them to each client as shown in Fig. 1b. Mathematically, this is exactly the same as centralized federated learning as shown in the Supplementary Information.

3) **DeceFL algorithm:** Each client communicates parameters through the topology of the undirected connected graph shown in Fig. 1c. According to the weighted aggregation of
information obtained from neighboring clients, the local update is completed according to the following iteration,

\[ w_k(t + 1) = \sum_{j=1}^{K} W_{kj}(t) w_k(t) - \eta(t) (w_k(t) - w_k(0)), \]

where \( W(t) \in \mathbb{R}^{K \times K} \) is the weighted matrix of the undirected connected graph at iteration \( t \). Using derivation in the Supplementary Information based on dynamical system and algebraic graph theory, it can be shown that the system reaches the steady-state, i.e., \( \lim_{t \to \infty} \| w_k(t) - w^* \| = 0 \), if \( \frac{\eta}{\Gamma} \) and \( \frac{\eta}{\Gamma} < 1 - \sigma' \) are satisfied\(^2\) for any \( \gamma, \Gamma > 0 \).

4) Summary: It can be shown that all methods can reach consensus with zero performance gap, i.e., \( \Delta = 0 \). The convergence speeds of all methods are \( O(1/T) \). As shown in the Supplementary Information Figure 1, these methods converge to the consensus value exponentially. This is consistent with the theoretical results. The information used in three different algorithms is however distinct: FedAvg and SL need global information at every iteration, while DeceFL only needs local information from clients’ neighbors. In addition, the first two algorithms can be viewed as special cases of the proposed DeceFL algorithm by setting \( W \) to the corresponding adjacency matrix correspondingly. Specifically, when \( W(t) = \frac{1}{K} \mathbf{1}_K \mathbf{1}_K^T \) for all \( t \), the proposed DeceFL algorithm becomes to FedAvg and SL.

RESULTS

Experiments were carried out on real-world biomedical and industrial applications, which demonstrated the effectiveness and the wide applicability of DeceFL, as a fully decentralized framework. We benchmarked the performance of DeceFL, in comparison with FedAvg and SL that demand much more communication costs and strongly rely on a highly restricted communication topology. Furthermore, the superiority of DeceFL on robustness in the presence of communication topology interference (random node or edge malfunction) was shown in two experiments with time-varying communication topologies. The overall performance of DeceFL was corroborated by these practical applications.

A. Application to medicine: collaborative prediction of leukaemias

First, we used the dataset of peripheral blood mononuclear cell (PBMC) transcriptomes from [23], named “dataset A2”, as a benchmark example to compare three federated learning

\(^2\)Let us sort the eigenvalues of \( W \) in a non-increasing order as \( 1 = \lambda_1(W) > \lambda_2(W) \geq \cdots \geq \lambda_n(W) > -1 \), denoted \( \sigma' \) as the second largest eigenvalue of the weighting matrix \( W \), i.e., \( \sigma' = \max\{ |\lambda_2(W)|, |\lambda_n(W)| \} \in (0, 1) \).
frameworks: DeceFL, FedAvg and SL. The experiment setup is consistent with that of SL: the dataset is divided into a training set and a test set at the ratio of 8:2, and the dataset owned by each node was obtained from the training set. The logistic regression model with $l_2$ regularization and the 8-layer fully connected deep neural network (as in [23]) were selected for our experiments detailed in the Materials and Methods Section.

![Diagram](https://engine.scichina.com/doi/10.1360/nso/20220043)

Fig. 2. DeceFL to predict leukaemias from A2 benchmark dataset [23]. a, Data were divided into IID samples for all clients. b, Data were divided into Non-IID unbalanced samples. c, Different topologies for FedAvg, SL and DeceFL respectively. The topology for SL must hold in every iteration when any other node is selected as a central client. d, e, f, g, Performance of three algorithms on IID/Non-IID setups over logistic regression/neural networks. DeceFL presents a similar performance as FedAvg and SL in both IID and Non-IID scenarios, except during the transient period that DeceFL takes to reach consensus. In contrast to fixed topology in FedAvg and SL, DeceFL provides much more freedom on the choice of communication topology.

First, we benchmarked DeceFL against FedAvg and SL in the IID setup of dataset A2 (Fig. 2a), that is, the sample size of each node is the training set sample size divided by the number of nodes, which ensures that each node has the same number of samples and the ratio of the positive to the negative samples is approximately 1:1. DeceFL used multiple connected...
communication topologies with various connectivity probability values \((p = 0.3, 0.5, 0.7, 0.9)\). This benchmark shows that DeceFL can reach the same performance as FedAvg and SL which use a (temporary) central client to gather all information from every client. FedAvg and SL only perform better during the transient period that DeceFL takes to converge due to its decentralized nature. Whereas DeceFL relaxes significantly on the choice of communication topology, which, in contrast to FedAvg and SL, no longer requires a fixed topology and thus has higher flexibility to deal with various demands from clients (e.g., one client could refuse to directly communicate with certain clients). Second, the similar comparative study was repeated with the Non-IID setup of dataset A2 (Fig. 2b). The Non-IID setup explicitly designs, for the local data associated with each node, the sample size and the ratio between positive and negative samples (Supplementary Information). It allows us to benchmark performances on balanced/unbalanced, sufficient/deficient local training data. We obtained very similar results as in the IID setup, where DeceFL presents a similar performance as FedAvg and SL, after DeceFL reaches consensus in decentralized computation. It also shows the superiority of DeceFL over SL, which however demands huge amounts of communication costs for the selected central client at every round and relies heavily on the strong assumption of a stable fully-connected communication structure. Any bit of malfunction of clients or communication paths could melt down the whole SL process, since at each round a client is delegated to collect information from all other clients.

To show DeceFL functions well when an intervention to decentralized infrastructure happens, we conducted two experiments with time-varying graphs that take into account malfunction of clients and communication paths. First, communication topology was altered in runtime (Fig. 3a), that is, the adjacency matrix that describes how clients communicate with each other varied over time. This time-varying experiment further shows that the conditions of DeceFL can even be weaken and generalized to such an extent that the communication graph at each time is not necessary to be connected as long as within a fixed period the information can be transmitted between any pair of clients. To calibrate the performance of DeceFL, we assume the ideal communication topology (no structural intervention) in the run of FedAvg (that is, a central node with consistent connections to every client). Surprisingly, both experimental results Fig. 3c (IID) and Fig. 3d (Non-IID) show that DeceFL in such a scenario keeps similar performance as FedAvg (with an ideal setup). In other words, DeceFL can be so robust that random malfunction of a portion of edges may merely deteriorate DeceFL running processes.

As another source of structural intervention, the second experiment considers the removal and
supplement of nodes, which is a significant issue of federated learning in practice and in business.
In a long-running and data-intensive machine learning project including hundreds of clients, that clients may drop out or new clients may join in the middle must not deteriorate or break such a time-consuming computation process. As shown in Fig. 3b, during the first 50 rounds, we used an Erdos-Renyi graph of 6 nodes; then for the 51-100 rounds, the graph was of 8 nodes by adding 2 extra nodes; and in the rest rounds, the graph was randomly removed by 2 nodes. Under such node interventions, experimental results Fig. 3e and Fig. 3f show robust performance of DeceFL which is similar to FedAvg (that does not consider node interventions). Two experiments manifest the robustness of DeceFL on interventions of computation infrastructure in a decentralized framework.

B. Application to smart manufacturing: collaborative detection of bearing faults

Another application of DeceFL lies in the manufacturing sector: modern manufacturing is heavily influenced by AI technologies with extraordinary increase of computational power and data size. To raise productivity and reduce operational costs, a critical challenge is fault diagnosis in machining operations [42]. AI-based algorithms have the potentials to detect fault locations and even to predict faults in advance, which allow replacing regular maintenance with real-time data-driven predictive maintenance and further reduce unnecessary maintenance costs and guarantee reliability. A general fault detection framework has been proposed in [43], which, however, needs full-cycle measurements of large amounts of machines that are most likely unavailable for a single factory. Data generated by multiple factories could be sufficient to perform preventive maintenance, while sensitive data (security or business related) are less likely to be shared in practice. The fully decentralized framework DeceFL provides a way for multiple factories to develop a global model, which generates mutual benefit from private local data without having to resort to data sharing in public.

This experiment practices such a decentralized fault diagnosis application in manufacturing, using Case Western Reserve University’s (CWRU) bearing data, which comprises ball bearing test data for normal and faulty bearings, specified in the Methods Section. Specifically, we used three types of bearings data: 7 inches, 14 inches and 21 inches; and chose the drive end defects, which includes outer race defect, inner race defect, and ball defect. We chose the outer race defect appearing at the 6 o’clock (centered) position. Thus, there are in total ten distinct conditions: 9 faulty classes (3 bearing types times 3 defect types) and the normal condition. All data in use was collected at 12,000 samples/second for drive end bearing experiments. It was generated by
Communication topology for DeceFL with time-varying nodes

Step 1
Step 2
Step 3
Step 4
Step 5

Fig. 3. DeceFL to predict leukaemias from A2 benchmark dataset [23]. a, Time-varying communication topology that consists of a sequence of graphs each of which is not connected while the lump-sum graph over a fixed period is connected. b, Time-varying communication topology that adds or removes nodes over time. c,d, Performance of DeceFL with edge-varying graphs on the IID and Non-IID setups of dataset A2 using logistic regression, with reference performance of FedAvg that uses full information. e,f, Performance of DeceFL with node-varying graphs on the IID and Non-IID setups of dataset A2 using logistic regression, with reference performance of FedAvg that uses full information. These time-varying experiments manifest the robustness of DeceFL in the presence of interventions to communication topology. The dropout or supplement of clients in the middle of DeceFL computation will not break or deteriorate the learning process, which is particularly essential in practice for large-scale time-consuming data-intensive applications.

using 4 types of motor speed: 1797 rpm, 1772 rpm, 1750 rpm and 1730 rpm. The data from 1730 rpm is reserved for test.

Assume that there are 4 factories, as clients illustrated in Fig. 4c, which collect their private full-cycle bearing data. The training data associated with each client were prepared in the IID (Fig. 4a) and the Non-IID setup (Fig. 4b). A 10-way classification problem is considered, 9 fault
cases (B007, IR007, OR007, B014, IR014, OR014, B021, IR021, OR021) and 1 normal case. Learning used two methods, regularized logistic regression as a strongly convex method, and deep neural network (DNN) as a nonconvex method. In the usage of logistic regression, as guaranteed in theory, DeceFL in Fig. 4d,f confirms the similar performance as FedAvg after its transient periods. For the case of DNN, as a non-convex method, although there is no theoretical guarantee, DeceFL in Fig. 4e,g show competitive performance to FedAvg. The slight performance gap in test between DeceFL and FedAvg in Fig. 4f may be mostly caused by the chosen type of DNN, multilayer perceptrons as used in [23], which has many well-known defects; and more reasons are discussed and explored by more experiments in the Supplementary Information. Comprehensive experiments were conducted and can be found in the Supplementary Information, with more clients (that is, more factories), more learning methods, and another 4-way classification problem. Overall DeceFL manifests competitive performance on multi-class classification for industrial fault diagnosis applications, with implementations of (non-)convex methods in a fully decentralized framework that breaks through the barrier of data privacy.

DISCUSSIONS

In this paper, we propose a decentralized federated learning framework for privacy-preserving collaborative learning. The decentralized architecture eliminates a large number of drawbacks–due to having a central client–that state-of-the-art privacy-preserving algorithms have. The convergence of the DeceFL algorithm is analyzed in detail, showing that DeceFL guarantees convergence and has the similar convergence rate as the centralized federated learning algorithm. The convergence performance of the algorithm is verified by training neural networks over different datasets. Compared with other state-of-the-art privacy-preserving algorithms such as FedAvg and SL [23], the proposed DeceFL algorithm is guaranteed to reach the global optimum with a similar rate as the centralized federated learning algorithm under certain conditions. In addition, there has developed a sizable literature as surveyed in [44], which can be adapted to cope with quantization errors and noises that could happen over communication networks.

There is no doubt that such a decentralized federated learning framework will become increasingly popular in the nearest future for almost all AI applications given the privacy regulations. Yet our algorithm has a number of limitations that need to be taken into consideration for future development:

First of all, application of privacy algorithms (for example, blockchain or homomorphic encryption [45]) has not been considered in this study. However, similar to the centralized
Fig. 4. DeceFL to detect bearing faults from CWRU benchmark dataset. 

- **a.** Data were divided into IID samples for all 4 clients. 
- **b.** Data were divided into Non-IID unbalanced samples. Each client locally specified its data size and sample distribution. 
- **c.** Illustration of communication topology for FedAvg, SL and DeceFL. 
- **d,e.** Performance of DeceFL on IID data using logistic regression and DNN, respectively, with reference performance of FedAvg and SL. 
- **f,g.** Performance of DeceFL on Non-IID data using logistic regression and DNN, respectively, with reference performance of FedAvg and SL. 

The boxplots at bottom illustrate the performance comparison between DeceFL and each client that was trained independently (that is, each client trained its own model only using its associated local data without communicating with any other clients). The experiments on CWRU benchmark dataset confirms the similar performance of DeceFL, as a fully decentralized framework, in comparison to FedAvg and SL, for industrial datasets, while DeceFL offers more freedom on the choice of communication topology to handle practical issues in organizing clients for large-scale federated learning applications.

Federated learning and swarm learning framework, it should be straightforward to apply such techniques for data privacy protection in the proposed DeceFL framework to make communication secure.
Secondly, similar to the setup of federated learning and swarm learning, all clients in the network need to know the form of the global objective function. The proposed formulation is different from those used in multi-party computation [46], where model and data can be separated [47]. Future work lies in the integration of such techniques to the proposed DeceFL algorithm to make the global objective function unknown to clients.

Finally, all clients are assumed to be collaborative, it would be interesting to further investigate whether the proposed decentralized federated learning framework is vulnerable to semi-honest or malicious clients that are not collaborative [48], which could exist in the real-world applications.

**Materials and Methods**

**Data pre-processing**

Essential data preprocessing was performed for CWRU dataset, including class balancing and normalization, feature extraction by Fourier transform. The dataset has 10 classes in total, which vary in sample size. Hence samples in certain classes were deleted to balance sample sizes over all classes. The original data is time-series data, which was firstly divided by every 300 points and resulted in a family of time series. Each time series chose DE and FE features respectively, and produced 600 points. For every time series of each feature, we performed Fast Fourier Transform (FFT), which yielded 150 points. Thus each time series of both DE and FE has in total 300 points. The motivation to use FFT is to handle the mismatch of time stamps of sequential data. After FFT the training and test data were then normalized by removing the mean and scaling to unit variance (the test data is normalized by the normalizer of the training).

**Performance metrics**

The common performance metric “accuracy” is used for assessment of classification,

\[
\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}
\]  

where TP, TN, FP and FN denote the number of true positive, true negative, false positive and false negative samples, respectively.

**Implementation of learning methods**

To ensure in benchmark DeceFL can work well for either strongly convex learning methods or non-convex methods, we adopted two algorithms: logistic regression with $l_2$ regularization, and deep neural network (DNN), as used in [23]. For logistic regression, every node runs 1 epoch in each communication round, with batch-size 64. It uses the SGD optimizer, with weight decay coefficient $10^{-4}$ for the realization of $l_2$ regularization. The learning rate is fixed to 0.001. For DNN, every node runs 5 epochs (dataset A2) or 10 epochs (CWRU dataset) in each communication round, with batch-size 64. It uses the SGD optimizer, with weight decay coefficient $10^{-4}$. The initial learning rate is 0.1, which is decayed by multiplying 0.1 every 20 communication rounds. This DNN has 8 hidden layers, whose dimensions are 256, 512, 512, 256, 256, 128, 128, 64, respectively. The dropout rate is set to 0.3. Most experiments use softmax
as the activation function in the output layer for classification, except the logistic regression for dataset A2 uses \texttt{sigmoid}. The total number of running rounds is selected by visualization effects of convergence for all methods in comparison.

**Data availability**

The peripheral blood mononuclear cell (PBMC)-derived transcriptome dataset, named as “dataset A2” in [23], was used, which was originally generated with Affymetrix HG-U133 2.0 microarrays (8,348 individuals), by inspection of all publicly available datasets at National Center for Biotechnology Information Gene Expression Omnibus. To perform the IID experiments, the initial preparation of dataset randomly dropped negative samples, resulting in 5,176 samples, such that the whole dataset is balanced, i.e., the ratio of the positive to the negative samples is 1 : 1. CWRU Bearing dataset refers to the data of ball bearing test for normal and faulty bearings from the Case Western Reserve University, available on https://engineering.case.edu/bearingdatacenter.

**Code availability**

All source codes are openly available on GitHub (https://github.com/HAIRLAB/DeceFL).

**Acknowledgement**

We thank Mr. Anthony Haynes, Mr. Cai Huang for editing.

**Funding**

This work was supported by the National Natural Science Foundation of China under Grant 92167201 and Jiangsu Industrial Technology Research Institute (JITRI).

**Author contributions**

Idea was conceived by Y.Y.. Theory was developed by J.L., R.C., L.X., X.Y., T.Y., Y.Y.. Simulation codes were developed by D.J., C.S., M.W. and were reviewed by Z.Y.. Experiments were designed by Y.Y., Z.Y., and were performed by D.J., Z.Y., C.S., M.W., Y.Y., F.H., R.C.. Projects were supervised by Y.Y., S.S., H.D.. Funding was acquired by Y.Y., J.L., H.Z., H.D.. The original draft was written by Y.Y., Z.Y., J.L., R.C., X.Y. and all authors provided critical review of the manuscript and approved the final draft.

**Competing interests**

The authors declare no competing interests.

**REFERENCES**

[1] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
[2] Yuan, Y. et al. Data driven discovery of cyber physical systems. *Nature Communications* **10**, 1–9 (2019).
[3] Yan, L. et al. An interpretable mortality prediction model for covid-19 patients. *Nature Machine Intelligence* **2**, 283–288 (2020).
[4] Price, W. N. & Cohen, I. G. Privacy in the age of medical big data. *Nature Medicine* **25**, 37–43 (2019).
[5] DeGrave, A. J., Janizek, J. D. & Lee, S.-I. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence* **1–10** (2021).
[6] Roberts, M. et al. Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and ct scans. *Nature Machine Intelligence* **3**, 199–217 (2021).

[7] Konečný, J., McMahan, B. & Ramage, D. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575* (2015).

[8] Yang, Q., Liu, Y., Chen, T. & Tong, Y. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology* **10**, 1–19 (2019).

[9] Konečný, J. et al. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* (2016).

[10] McMahan, B., Moore, E., Ramage, D., Hampson, S. & y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, 1273–1282 (2017).

[11] Rizk, E., Vlaski, S. & Sayed, A. H. Dynamic federated learning. *arXiv preprint arXiv:2002.08782* (2020).

[12] Vlaski, S., Rizk, E. & Sayed, A. H. Second-order guarantees in federated learning. *2020 54th Asilomar Conference on Signals, Systems, and Computers* 915–922 (2020).

[13] Bonawitz, K. et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046* (2019).

[14] Gu, R., Yang, S. & Wu, F. Distributed machine learning on mobile devices: A survey. *arXiv preprint arXiv:1909.08329* (2019).

[15] Kairouz, P. et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977* (2019).

[16] Li, T., Sahu, A. K., Talwalkar, A. & Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* **37**, 50–60 (2020).

[17] Liu, J. et al. From distributed machine learning to federated learning: A survey. *Knowledge And Information Systems* **64**, 885–917 (2022).

[18] Bellet, A., Guerraoui, R., Taziki, M. & Tommasi, M. Personalized and private peer-to-peer machine learning. *arXiv preprint arXiv:1705.08435* (2017).

[19] Kaisiss, G. et al. End-to-end privacy preserving deep learning on multi-institutional medical imaging. *Nature Machine Intelligence* **3**, 473–484 (2021).

[20] Lian, X. et al. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In *31th Annual Conference on Neural Information Processing Systems (NIPS 2017)*, vol. 30, 5330–5340 (2017).

[21] Tang, Z., Shi, S. & Chu, X. Communication-efficient decentralized learning with sparsification and adaptive peer selection. *arXiv preprint arXiv:2002.09692* (2020).

[22] Wang, S. et al. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications* **37**, 1205–1221 (2019).

[23] Warnat-Herresthal, S. et al. Swarm learning for decentralized and confidential clinical machine learning. *Nature* **594**, 265–270 (2021). URL https://doi.org/10.1038/s41586-021-03583-3.

[24] Blot, M., Picard, D., Cord, M. & Thome, N. Gossip training for deep learning. *arXiv preprint arXiv:1611.09726* (2016).

[25] Daily, J. A., Vishnu, A., Siegel, C. M., Warfel, T. E. & Amatya, V. C. Gossipgrad: Scalable deep learning using gossip communication based asynchronous gradient descent. *arXiv preprint arXiv:1803.05880* (2018).

[26] Hu, C., Jiang, J. & Wang, Z. Decentralized federated learning: A segmented gossip approach. *arXiv preprint arXiv:1908.07782* (2019).
[27] Scardapane, S., Spinelli, I. & Lorenzo, P. D. Distributed training of graph convolutional networks. *IEEE Transactions on Signal and Information Processing over Networks* 7, 87–100 (2021).

[28] Pei, Y. *et al.* Decentralized federated graph neural networks. In *International Joint Conference on Artificial Intelligence (IJCAI)* (2021).

[29] Lalitha, A. Fully decentralized federated learning. In *Annual Conference on Neural Information Processing Systems (NeurIPS)* (2018).

[30] Lalitha, A., Kilinc, O. C., Javidi, T. & Koushanfar, F. Peer-to-peer federated learning on graphs. *arXiv preprint arXiv:1901.11173* (2019).

[31] Pappas, C., Chatzopoulos, D., Lalis, S. & Vavalis, M. Ipls: A framework for decentralized federated learning. In *2021 IFIP Networking Conference* (2021).

[32] Savazzi, S., Kianoush, S., Rampa, V. & Bennis, M. A joint decentralized federated learning and communications framework for industrial networks. In *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 1–7 (2020).

[33] Wu, X., Wang, Z., Zhao, J., Zhang, Y. & Wu, Y. FedBC: Blockchain-based decentralized federated learning. *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)* 217–221 (2020).

[34] Lyu, L. *et al.* Towards fair and privacy-preserving federated deep models. *IEEE Transactions on Parallel and Distributed Systems* 31, 2524–2541 (2020).

[35] Yapp, A. Z. H. *et al.* Communication-efficient and scalable decentralized federated edge learning. In *International Joint Conference on Artificial Intelligence (IJCAI)* (2021).

[36] Zhu, L., Liu, Z., & Han, S. Deep leakage from gradients. In *Annual Conference on Neural Information Processing Systems (NeurIPS)* (2019).

[37] Koloskova, A., Stich, S. U. & Jaggi, M. Decentralized stochastic optimization and gossip algorithms with compressed communication. *arXiv preprint arXiv:1902.00340* (2019).

[38] Li, X., Huang, K., Yang, W., Wang, S. & Zhang, Z. On the convergence of FedAvg on non-IID data. *arXiv preprint arXiv:1907.02189* (2019).

[39] Olfati-Saber, R. & Murray, R. M. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control* 49, 1520–1533 (2004).

[40] Ren, W. & Beard, R. W. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on automatic control* 50, 655–661 (2005).

[41] Jadabaie, A., Lin, J. & Morse, A. S. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on automatic control* 48, 988–1001 (2003).

[42] Isermann, R. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance* (Springer Science & Business Media, 2005).

[43] Yuan, Y. *et al.* A general end-to-end diagnosis framework for manufacturing systems. *National Science Review* 7, 418–429 (2020).

[44] Yang, T. *et al.* A survey of distributed optimization. *Annual Reviews in Control* 47, 278–305 (2019).

[45] DeMillo, R. A. Foundations of secure computation. Tech. Rep., Georgia Institute of Technology (1978).

[46] Yao, A. C. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, 160–164 (IEEE, 1982).

[47] Yu, Y. & Xie, X. Privacy-preserving computation in the post-quantum era. *National Science Review* (2021).

[48] Lyu, L., Yu, H. & Yang, Q. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133* (2020).