IEEE ComSoc | IEEE COMPUTER SOCIETY | IEEE Signal Processing Society | VTS | IEEE Transactions on Machine Learning in Communications and Networking

Received 25 May 2023; revised 5 November 2023, 18 March 2024, and 6 May 2024; accepted 8 May 2024.
Date of publication 16 May 2024; date of current version 24 May 2024.

The associate editor coordinating the review of this article and approving it for publication was N. H. Tran.

Digital Object Identifier 10.1109/TMLCN.2024.3400756

DIN: A Decentralized Inexact Newton Algorithm for Consensus Optimization

ABDULMOMEN GHALKHA[1], CHAOUKI BEN ISSAID[1], ANIS ELGABLI[2] (Senior Member, IEEE), AND MEHDI BENNIS[1] (Fellow, IEEE)

[1]Centre for Wireless Communications (CWC), University of Oulu, 90570 Oulu, Finland
[2]Department of Industrial and Systems Engineering, Interdisciplinary Research Center for Communication Systems and Sensing (IRC-CSS), Interdisciplinary Research Center for Intelligent Secure Systems (IRC-ISS), King Fahd University of Petroleum and Minerals (KFUPM), Dhahran 31261, Saudi Arabia

CORRESPONDING AUTHOR: A. GHALKHA (abdulmomen.ghalkha@oulu.fi)

This work was supported by the European Commission through Grant No. 101095363 (Horizon Europe SNS JU ADROIT6G project).

ABSTRACT This paper tackles a challenging decentralized consensus optimization problem defined over a network of interconnected devices. The devices work collaboratively to solve a problem using only their local data and exchanging information with their immediate neighbors. One approach to solving such a problem is to use Newton-type methods, which are known for their fast convergence. However, these methods have a significant drawback as they require transmitting Hessian information between devices. This not only makes them communication-inefficient but also raises privacy concerns. To address these issues, we present a novel approach that transforms the Newton direction learning problem into a formulation composed of a sum of separable functions subjected to a consensus constraint and learns an inexact Newton direction alongside the global model without enforcing devices to share their computed Hessians using the proximal primal-dual (Prox-PDA) algorithm. Our algorithm, coined DIN, avoids sharing Hessian information between devices since each device shares a model-sized vector, concealing the first- and second-order information, reducing the network's burden and improving both communication and energy efficiencies. Furthermore, we prove that DIN descent direction converges linearly to the optimal Newton direction. Numerical simulations corroborate that DIN exhibits higher communication efficiency in terms of communication rounds while consuming less communication and computation energy compared to existing second-order decentralized baselines.

INDEX TERMS Distributed optimization, decentralized learning, communication-efficient federated learning, second-order methods.

I. INTRODUCTION

MINIMIZING a sum of functions in a distributed manner is motivated by a wide range of applications in various networked systems, such as smart grids [1], federated learning (FL) [2], and wireless sensor networks [3]. A traditional approach involves using a central (on-cloud) server with high computational and storage capabilities, and each device sends its raw data to the server, which applies centralized optimization to minimize a global objective function. Although this traditional approach is simple, it suffers high communication costs and violates privacy [4]. To enable collaborative learning while protecting privacy, privacy-preserving collaborative learning techniques are necessary. Recently, thanks to the fast growth of the computation power of edge clients, the transmission of raw and private data to the cloud can be avoided using federated learning (FL). In the canonical FL approach, local models/gradients are updated locally, and an on-cloud parameter server (PS) aggregates the local models/gradients to update the global model/gradient, which is then shared with edge clients. Iterating this way, eventually, all clients converge to a global model.

Existing FL algorithms can be categorized into three groups based on which information from the objective function is used in the optimization process. Zeroth-order algorithms are the first category in which clients are limited to using samples of their own objective functions [5]. The

second category is first-order algorithms where edge clients use the gradients of their objective functions, to decide the direction of the update. Primal methods such as federated averaging [6], and primal-dual methods such as distributed alternating direction method of multipliers (ADMM) [7] are examples of first-order algorithms. Second-order algorithms, which are the last category, employ the objective function's second-order information, i.e., Hessian matrix, at each iteration. Despite the fast convergence of the Newton method, which is the standard second-order algorithm, it suffers from high communication cost. Moreover, it introduces privacy issues, since the Hessian matrix contains important information about the characteristics of the local objective function and data. For instance, the authors in [8] demonstrated how information from input images can be extracted using the eigenvalues of the Hessian matrix. The aforementioned frameworks require a PS to aggregate the first- and second-order data received from edge clients. Relying on a single PS may introduce a lot of communication overhead and may not even be possible in a large system. Moreover, as an aggregation hub, the network may experience a single point of failure [9]. Therefore fully decentralized approaches, where there is no central PS, have been gaining popularity. In fully decentralized FL, edge clients share their local information with their neighboring clients to establish model consensus, which avoids creating a single point of failure while reducing the communication bottleneck that occurs at the PS [10], [11], [12].

A. RELATED WORKS

Communication-efficient solutions for distributed optimization have been a study subject of several articles. The following discussion highlights various techniques.

1) FIRST-ORDER METHODS

The standard approach to solve the distributed optimization problem in the PS-based topology is to use first-order methods such as distributed gradient descent (DGD). At every iteration of DGD, each client computes its local gradient with respect to the current model parameters and sends that information to the PS. After receiving all gradients, the PS computes the global gradient and executes one GD step. In decentralized settings, the local gradients are shared among neighboring clients where each client averages the received gradients and then performs a local GD step to update its local model. Although first-order methods enjoy low computation complexity, they suffer from a slow convergence rate, which depends on the condition number [13]. For example, given a function that is L-smooth and μ-strongly convex, GD achieves a global linear convergence rate of $1 - \frac{2}{\kappa-1}$ for a step-size of $\alpha = \frac{2}{\mu+L}$, where $\kappa = \frac{L}{\mu}$ defines the condition number [14]. This calls for a large number of communication rounds; in addition to considerable energy and bandwidth resources per communication round. These issues can be tackled by reducing the number of communication rounds [15] and/or minimizing the communication overhead per communication round by leveraging some quantization and compression schemes. Several techniques were proposed to reduce the number of communication rounds; for example by accelerating the convergence using momentum [16], [17] and/or adaptive learning rate [18]. On the other hand, several quantization [19], [20] and censoring schemes [21] were proposed to minimize the payload size per communication round while maintaining the convergence guarantees. It is worth mentioning that using a fixed step size, DGD can only converge to a neighborhood of an optimal solution [22]. Gradient tracking decentralized gradient descent GTDGD tackles this and converges to the optimal solution with a fixed size by estimating the global gradient descent direction using the neighboring and past local gradients [23] by every client.

2) SECOND-ORDER METHODS

Recently, second-order algorithms have attracted a lot of attention, owing to their faster convergence compared to first-order techniques, by taking advantage of the second derivative's curvature information, which gives adaptive update directions. Although this reduces the number of communication rounds, second-order information necessitates significant computation and communication costs. In every communication round, the Hessian matrix is computed and transmitted, which induces a communication cost of $\mathcal{O}(n^2)$ per iteration compared to $\mathcal{O}(n)$ in first-order methods, where n is the dimension of the model. Furthermore, Newton's approach is sensitive to inversion attacks since it involves sharing both the gradient and the Hessian at each iteration, which creates a privacy concern [24].

The problem of sending the exact Hessian matrix has been addressed in various studies with communication-efficient solutions that avoid sending the exact Hessian. The authors in [25] suggested a Newton-based framework, in which edge clients communicate a compressed version of the local Hessian. However, gradients and compressed Hessians are still communicated; hence the privacy issue is not completely addressed. In a recent work [26], the privacy issue was solved by learning the inverse Hessian-gradient product. The idea is to formulate an inner problem with the objective of learning the inverse Hessian-gradient. One alternating direction method of multipliers (ADMM) step is performed at the client's side in every outer (global) iteration to approximate the solution of the inner problem, and then the output is shared with the PS. The algorithm still relies on the presence of a Parameter Server (PS) to aggregate received directions and construct the global Newton direction, introducing the risk of a single point of failure. In our work, we extend this concept to a fully decentralized setting, effectively mitigating the aforementioned PS-related limitations. This approach is especially crucial for applications dependent on battery-powered devices, where energy efficiency, privacy preservation, and the elimination of single points of failure are critical. These applications span various domains, including UAV networks [27], and ultra-reliable, low-latency communication in vehicular networks [28]. Few works have utilized second-order information in decentralized settings to accelerate convergence. In [29], the authors approximate

the exact Newton step by utilizing the initial $K+1$ terms of the Taylor series expansion associated with the Hessian matrix's inverse. To achieve this, they represent the Hessian matrix as the sum of two components, denoted as $\boldsymbol{D}$ and $\boldsymbol{A}$, where D corresponds to the diagonal elements, and A corresponds to the off-diagonal elements. Additionally, they leverage a mathematical expansion rule, expressed as $(\boldsymbol{I}-\boldsymbol{Z})^{-1}=\sum_{j=0}^{\infty}\boldsymbol{Z}^j$, with $\boldsymbol{Z}=\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$ to represent $\boldsymbol{H}^{-1}$. This approach allows them to approximate the Newton step while considering a limited number of terms from the expansion. However, it requires multiple exchanges of the local directions to resemble the exact Hessian, which calls for more communication rounds. Authors in [30] incorporate the local Hessian in the update direction while tracking the gradient. However, the local Newton direction may not be a good estimate for the global one, which calls for additional communication rounds to converge. Throughout this paper, we will refer to both algorithms as Network Newton (NN) and Newton Tracking (NT). Table 1 illustrates a comparison with related algorithms. The comparison is in terms of the communication overhead and storage requirements which reflects the energy consumption as we'll see in Section V.

An alternative to approximate Newton methods is the utilization of quasi-Newton techniques that rely on gradients to estimate curvature, avoiding the need for Hessian inverses. Authors in [31] and [32] introduce a distributed adaptation of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton optimization method. This approach focuses on preserving the global secant condition and offers advantages over Newton methods as it does not require Hessian computation and is applicable to scenarios where gradients are distributedly computable regardless of Hessian's structure. However, Quasi-Newton matrices are often dense, and for problems with large model sizes, the storage and computation requirements associated with those dense Quasi-Newton matrices can become prohibitive. Furthermore, despite these algorithms relying on first-order information to estimate the Hessian matrix, they necessitate the sharing of multiple control vectors in addition to the local gradients, such as the model and the neighborhood descent directions. This inclusion introduces an additional overhead during each communication round.

B. CONTRIBUTIONS AND OUTLINE

In this paper, we propose DIN, a novel second-order based, decentralized, and communication-efficient FL scheme that reduces the communication overhead per iteration and preserves privacy by concealing the gradient and the Hessian. DIN learns the inverse Hessian-gradient product alongside the model. The problem of learning the inverse Hessian-gradient product is formulated as a constrained optimization problem and a framework based on Prox-PDA is used to learn $\nabla^2 f(\boldsymbol{x})^{-1}\nabla f(\boldsymbol{x})$. In contrast to $\mathcal{O}(n^2)$ in standard Newton, each client in this step shares a model-sized vector, yielding $\mathcal{O}(n)$ communication complexity per iteration. Each client updates its model utilizing the inexact Newton step using the average estimates of the Newton direction received from the neighboring clients. We extend a prior work [33] and provide a detailed convergence analysis of DIN. Our contribution can be summarized as follows

- We propose DIN, a communication and energy-efficient decentralized FL framework that uses second-order information to solve the consensus optimization problem. More specifically, we use the Prox-PDA [12] algorithm to tackle the problem of learning the inverse-Hessian-gradient product by decomposing the global inverse-Hessian-gradient product learning function into a sum of separable local functions. DIN does not require clients to share their explicit gradient and Hessian matrix at any iteration, resulting in a communication cost of $\mathcal{O}(n)$ per iteration and privacy preservation.
- We prove convergence of DIN algorithm to an optimal direction of Newton method under some assumptions in Section. IV. The proof demonstrates that DIN converges linearly and the optimality gap goes to zero.
- We conducted several experiments to solve the decentralized logistic regression problem with real datasets while capturing energy consumption. Numerical results show that DIN outperforms NN and NT methods under different network topologies and graph densities. We also show that DIN consumes less energy to achieve the same optimality gap.

The paper is structured as follows. In Section II, we describe the system model and problem formulation. In Section III, we describe our proposed algorithm. Then, we conduct several numerical experiments to compare the performance of DIN with key baselines in Section V. Furthermore, we give a conclusion of our work in Section VI, and finally, prove the convergence of DIN in Appendices A-D.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a connected network consisting of N devices, each having a local loss function $f_i:\mathbb{R}^d\to\mathbb{R}$, assumed to be a convex and second-order differentiable, known only to device i. The devices are connected through a graph $\mathcal{G}=\{\mathcal{V},\mathcal{E}\}$, where $\mathcal{V}$ and $\mathcal{E}$ are the node and edge sets, respectively. Devices collaborate to minimize the empirical loss/risk, i.e., the average of their local objective functions, to learn a common model, $\boldsymbol{x}\in\mathbb{R}^d$. Every device i can only communicate with its immediate neighbors, defined as $\mathcal{N}_i=\{j|(i,j)\in\mathcal{E}\}$, with $|\mathcal{N}_i|=\delta_i$ denote the cardinality of its neighbor set. Specifically, the devices' goal is to find the model that solves the following learning problem in a decentralized manner

$$\text{(P1)}\quad \min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})=\frac{1}{N}\sum_{i=1}^{N} f_i(\boldsymbol{x}). \tag{1}$$

The starting point of our work is the Newton-like method introduced in [34], which solves (P1) in the presence of a PS. At iteration $(k+1)$, the *Newton* step update is given by

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k-\Big(\frac{1}{N}\sum_{i=1}^{N}\nabla^2 f_i(\boldsymbol{x}^k)\Big)^{-1}\Big(\frac{1}{N}\sum_{i=1}^{N}\nabla f_i(\boldsymbol{x}^k)\Big), \tag{2}$$

TABLE 1. Communication overhead and storage requirements for decentralized consensus optimization algorithms to solve (1)

Algorithm	Hessian utilization	Communication overhead	Storage requirements
DGD	No	$\mathcal{O}(n)$	$\mathcal{O}(n)$
GTDGD	No	$\mathcal{O}(2n)$	$\mathcal{O}(2n)$
NN	Yes	$\mathcal{O}(2n)$	$\mathcal{O}(2n^2)$
NT	Yes	$\mathcal{O}(n)$	$\mathcal{O}(2n^2)$
DIN	Yes	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$

where $\nabla^2 f_i(\cdot) \in \mathbb{R}^{d\times d}$ and $\nabla f_i(\cdot)$ are the Hessian and the gradient of $f_i(\cdot)$, respectively. For ease of notations, we define $\boldsymbol{H}_i^k = \nabla^2 f_i(\boldsymbol{x}^k)$ and $\boldsymbol{g}_i^k = \nabla f_n(\boldsymbol{x}^k)$ as the Hessian matrix and the gradient vector of device i evaluated at $\boldsymbol{x}^k$, respectively. We also define the network Hessian and gradient as

$$\bar{\boldsymbol{H}}^k = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{H}_i^k \quad \text{and} \quad \bar{\boldsymbol{g}}^k = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{g}_i^k. \tag{3}$$

Hence, we can write the step in (2) as follows

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - (\bar{\boldsymbol{H}}^k)^{-1}\bar{\boldsymbol{g}}^k. \tag{4}$$

Note that (4) can be implemented if every device has access to the average of all gradients and all Hessians evaluated at $\boldsymbol{x}^k$. However, this update cannot be implemented in a decentralized way since every device can only exchange information with a limited number of neighbors; thus it cannot obtain $\bar{\boldsymbol{g}}^k$ and $\bar{\boldsymbol{H}}^k$. Before we present our algorithm, we start by introducing matrices related to the network topology

- The degree matrix $\tilde{\boldsymbol{D}} = \text{diag}[\delta_1, \delta_2, \ldots, \delta_N]$, is a diagonal matrix containing the number of neighbors of each device, i.e., the degree of the device i.
- The incidence matrix $\tilde{\boldsymbol{A}}$ with entries $\tilde{\boldsymbol{A}}(k,i) = 1$ and $\tilde{\boldsymbol{A}}(k,j) = -1$ if $k = (i,j) \in \mathcal{E}$ with $j > i$.
- The signed and signless Laplacian matrices defined as $\tilde{\boldsymbol{L}}_- = \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{L}}_+ = 2\tilde{\boldsymbol{D}} - \tilde{\boldsymbol{L}}_-$, respectively.

We also define the extended versions of these matrices where the extended definition is given by taking the Kronecker product with the identity matrix, i.e., $\boldsymbol{A} = \tilde{\boldsymbol{A}} \otimes \boldsymbol{I}_d$.

III. PROPOSED ALGORITHM

Inspired by [26], we propose to replace the *inverse Hessian-gradient product*, i.e., the term $(\bar{\boldsymbol{H}}^k)^{-1}\bar{\boldsymbol{g}}^k$ in (4), with an approximate solution of the following optimization problem

$$\boldsymbol{d}^k = \arg\min_{\boldsymbol{d}\in\mathbb{R}^d} \frac{1}{2}\boldsymbol{d}^T\bar{\boldsymbol{H}}^k\boldsymbol{d} - \boldsymbol{d}^T\bar{\boldsymbol{g}}^k. \tag{5}$$

Specifically, when solving the problem in (5) at iteration k, we find the direction $\boldsymbol{d}^k = (\bar{\boldsymbol{H}}^k)^{-1}\bar{\boldsymbol{g}}^k$. Nevertheless, the solution to this problem in a decentralized manner is still not possible. To this end, we reformulate the problem in (5) and cast it as a decentralized optimization problem

$$\text{(P2)}\ (\boldsymbol{d}^\star)^k = \arg\min_{\{\boldsymbol{d}_i\}_{i=1}^N\in\mathbb{R}^d} \left\{\phi^k(\boldsymbol{d}) = \sum_{i=1}^{N}\phi_i^k(\boldsymbol{d}_i)\right\}$$
$$s.t.\ \ \boldsymbol{d}_i = \boldsymbol{d}_j,\ \ \forall\ (i,j)\in\mathcal{E}, \tag{6}$$

Algorithm 1 **D**ecentralized **I**nexact **N**ewton (DIN)

1: **Input**: N, $\{f_i(\cdot)\}_{i=1}^N$, ρ, K,
2: **Output**: $\boldsymbol{x}$, $\forall i$
3: **Initialization**: $\boldsymbol{x}_i^0$, $\boldsymbol{d}_i^{(-1)}$, $\boldsymbol{\lambda}_i^{(-1)}$, $\forall i$.
4: **for** $k = 0, \ldots, K$ **do**
5: **Every node in parallel**
6: Computes its Newton direction using

$$\boldsymbol{d}_i^k = (\boldsymbol{H}_{i,\alpha_i}^k)^{-1}\left(\boldsymbol{g}_i^k - \boldsymbol{\lambda}_i^{k-1} + \rho\left(\delta_i\boldsymbol{d}_i^{k-1} + \sum_{j\in\mathcal{N}_i}\boldsymbol{d}_j^{k-1}\right)\right).$$

7: Updates its dual variable via

$$\boldsymbol{\lambda}_i^k = \boldsymbol{\lambda}_i^{k-1} + \rho\left(\delta_i\boldsymbol{d}_i^k - \sum_{j\in\mathcal{N}_i}\boldsymbol{d}_j^k\right).$$

8: Updates its local model using $\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^k - \boldsymbol{d}_i^k$.
9: **end for**

where $\phi_i^k(\boldsymbol{d}_i) = \frac{1}{2}\boldsymbol{d}_i^T(\boldsymbol{H}_i^k + \alpha_i\boldsymbol{I}_d)\boldsymbol{d}_i - \boldsymbol{d}_i^T\boldsymbol{g}_i^k$, $\{\alpha_i\}_{i=1}^N$ are hyperparameters that we introduce to make sure that the matrices $(\boldsymbol{H}_i^k + \alpha_i\boldsymbol{I}_d)$ are invertible, and $\boldsymbol{d} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N]^T \in \mathbb{R}^{Nd}$ the concatenation of the local directions. Note that the inexact Newton direction, i.e., $-(\boldsymbol{H}_i^k + \alpha_i\boldsymbol{I}_d)^{-1}\boldsymbol{g}_i^k$, is also a valid descent direction [30], [35], [36]. For a given $\boldsymbol{x}_i^k$, solving (P2) exactly, i.e., until converging to $(\boldsymbol{d}^\star)^k$, comes at a very high communication cost since devices need to iterate and communicate their updates at each iteration until convergence. In this work, we propose to perform a single update at each outer iteration k to approximate the solution of (P2) and reduce the communication cost. In what follows, we elaborate on how the single pass update of the direction $\boldsymbol{d}$ is done. Using these introduced notations, (P2) can be re-written as

$$\min_{\boldsymbol{d}\in\mathbb{R}^{Nd}} \phi^k(\boldsymbol{d})$$
$$s.t.\ \ \boldsymbol{A}\boldsymbol{d} = \boldsymbol{0} \tag{7}$$

The augmented Lagrangian of (7) is given as

$$\mathcal{L}_\rho^k(\boldsymbol{d},\boldsymbol{\mu}) = \phi^k(\boldsymbol{d}) + \langle\boldsymbol{\mu}, \boldsymbol{A}\boldsymbol{d}\rangle + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{d}\|^2, \tag{8}$$

where $\rho > 0$ is a constant penalty parameter, and $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_N]^T \in \mathbb{R}^{Nd}$ is the concatenation of the dual variables. Minimizing the augmented Lagrangian directly leads to a solution that cannot be implemented in a decentralized way. Instead, we leverage the Prox-PDA algorithm [12],

which adds the proximal term $\frac{\rho}{2}\|\boldsymbol{d}-\boldsymbol{d}^{k-1}\|^2_{\boldsymbol{L}_+}$. In this case, the update of the primal variables, at iteration k, is given by solving the following optimization problem [12]

$$\min_{\boldsymbol{d}\in\mathbb{R}^{Nd}} \phi^k(\boldsymbol{d})+\langle\boldsymbol{\mu}^{k-1},\boldsymbol{A}\boldsymbol{d}\rangle+\frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{d}\|^2+\frac{\rho}{2}\|\boldsymbol{d}-\boldsymbol{d}^{k-1}\|^2_{\boldsymbol{L}_+}. \quad (9)$$

Using $\boldsymbol{L}_- = \boldsymbol{A}^T\boldsymbol{A}$ and $2\boldsymbol{D} = \boldsymbol{L}_- + \boldsymbol{L}_+$, we can write

$$\min_{\boldsymbol{d}\in\mathbb{R}^{Nd}} \phi^k(\boldsymbol{d}) + \langle\boldsymbol{\mu}^{k-1},\boldsymbol{A}\boldsymbol{d}\rangle + \rho\boldsymbol{d}^T\boldsymbol{D}\boldsymbol{d} - \rho\boldsymbol{d}^T\boldsymbol{L}_+\boldsymbol{d}^{k-1}. \quad (10)$$

Setting the derivative with respect to $\boldsymbol{d}$ to zero, we get

$$\nabla\phi^k(\boldsymbol{d}^k) + \boldsymbol{A}^T\boldsymbol{\mu}^{k-1} + 2\rho\boldsymbol{D}\boldsymbol{d}^k - \rho\boldsymbol{L}_+\boldsymbol{d}^{k-1} = \boldsymbol{0}. \quad (11)$$

On the other hand, the update of $\boldsymbol{\mu}$, at iteration k, is given by

$$\boldsymbol{\mu}^k = \boldsymbol{\mu}^{k-1} + \rho\boldsymbol{A}\boldsymbol{d}^k. \quad (12)$$

Next, we define $\boldsymbol{\lambda} = \boldsymbol{A}^T\boldsymbol{\mu}$ and multiply both sides in (12) by $\boldsymbol{A}^T$. Using the fact that $\boldsymbol{L}_- = \boldsymbol{A}^T\boldsymbol{A}$, we get

$$\boldsymbol{\lambda}^k = \boldsymbol{\lambda}^{k-1} + \rho\boldsymbol{L}_-\boldsymbol{d}^k. \quad (13)$$

Hence, the dual variable of the i^{th} device is updated as

$$\boldsymbol{\lambda}_i^k = \boldsymbol{\lambda}_i^{k-1} + \rho\left(\delta_i\boldsymbol{d}_i^k - \sum_{j\in\mathcal{N}_i}\boldsymbol{d}_j^k\right). \quad (14)$$

Writing the update of the primal variable of the i^{th} device from (7), we get

$$\nabla\phi_i^k(\boldsymbol{d}_i^k)+\boldsymbol{\lambda}_i^{k-1}+2\rho\delta_i\boldsymbol{d}_i^k-\rho\left(\delta_i\boldsymbol{d}_i^{k-1}+\sum_{j\in\mathcal{N}_i}\boldsymbol{d}_j^{k-1}\right)=\boldsymbol{0}. \quad (15)$$

Replacing the expression of $\nabla\phi^k(\boldsymbol{d}^k)$ and re-arranging the terms, we can write

$$\boldsymbol{d}_i^k=(\boldsymbol{H}_{i,\alpha_i}^k)^{-1}\left(\boldsymbol{g}_i^k-\boldsymbol{\lambda}_i^{k-1}+\rho\left(\delta_i\boldsymbol{d}_i^{k-1}+\sum_{j\in\mathcal{N}_i}\boldsymbol{d}_j^{k-1}\right)\right), \quad (16)$$

where $\boldsymbol{H}_{i,\alpha_i}^k = \boldsymbol{H}_i^k + (2\rho\delta_i+\alpha_i)\boldsymbol{I}_d$. Finally, the local model is updated using the local Newton direction as

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^k - \boldsymbol{d}_i^k. \quad (17)$$

The details of our algorithm are summarized in Algorithm 1.

IV. CONVERGENCE ANALYSIS

This section examines the convergence of the proposed DIN algorithm under the assumption that each function f_i in (P1) is both strongly convex and twice differentiable. Additionally, we introduce the extended function ϕ that is defined as

$$\phi(\boldsymbol{d}) = \frac{1}{2}\boldsymbol{d}^T(\boldsymbol{H}+\boldsymbol{\Gamma})\boldsymbol{d} - \boldsymbol{d}^T\boldsymbol{g}. \quad (18)$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{Nd\times Nd}$, and $\boldsymbol{H}$ are block diagonal matrices with ith blocks $\alpha_i\boldsymbol{I}_d$, and $\nabla^2 f_i(\boldsymbol{x}_i)$, respectively. Furthermore, ϕ is assumed to have an L-Lipschitz continuous gradient in $\boldsymbol{d}$. That is, for any $\boldsymbol{d}_1, \boldsymbol{d}_2 \in \mathbb{R}^{Nd}$, we have

$$\|\nabla\phi(\boldsymbol{d}_1) - \nabla\phi(\boldsymbol{d}_2)\| \le L\|\boldsymbol{d}_1 - \boldsymbol{d}_2\|. \quad (19)$$

Additionally, since α_i is chosen such that $(\boldsymbol{H}+\boldsymbol{\Gamma})$ is positive definite, ϕ is strongly convex with a parameter μ, and we have

$$\|\nabla\phi(\boldsymbol{d}_1) - \nabla\phi(\boldsymbol{d}_2)\| \ge \mu\|\boldsymbol{d}_1 - \boldsymbol{d}_2\|. \quad (20)$$

Lemma 1: From the definition of the function ϕ given in (18), and the assumptions in (19), and (20), the following inequality holds

$$\begin{aligned}&\frac{\mu L}{\mu+L}\|\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k\|^2\\&+\frac{\mu}{L(\mu+L)}\|\nabla\phi^k(\boldsymbol{d}^k)-\nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2\\&\le(\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T(\nabla\phi^k(\boldsymbol{d}^k)-\nabla\phi^k((\boldsymbol{d}^\star)^k)).\end{aligned} \quad (21)$$

Proof: The proof details are deferred to Appendix A. ■

Using the definition of $\boldsymbol{\lambda}$, we can write (11) as

$$\nabla\phi^k(\boldsymbol{d}^k) + \boldsymbol{\lambda}^{k-1} + 2\rho\boldsymbol{D}\boldsymbol{d}^k - \rho\boldsymbol{L}_+\boldsymbol{d}^{k-1} = \boldsymbol{0}. \quad (22)$$

The necessary and sufficient optimality conditions of the inner problem in (6) at the k-th iteration are given by

$$\boldsymbol{A}(\boldsymbol{d}^\star)^k = \boldsymbol{0}, \quad \text{(primal feasibility)} \quad (23)$$

$$\nabla\phi^k((\boldsymbol{d}^\star)^k) + (\boldsymbol{\lambda}^\star)^k = \boldsymbol{0}. \quad \text{(dual feasibility)} \quad (24)$$

Lemma 2: For each iteration k of the DIN algorithm, it holds that

$$\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1} - \rho\boldsymbol{L}_-(\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k) = \boldsymbol{0}, \quad (25)$$

and

$$\begin{aligned}&\nabla\phi^k(\boldsymbol{d}^k) - \nabla\phi^k((\boldsymbol{d}^\star)^k) + \boldsymbol{\lambda}^k - (\boldsymbol{\lambda}^\star)^k\\&+ 2\rho\boldsymbol{D}\boldsymbol{d}^k - \rho\boldsymbol{L}_-\boldsymbol{d}^k - \rho\boldsymbol{L}_+\boldsymbol{d}^{k-1} = \boldsymbol{0}.\end{aligned} \quad (26)$$

Proof: The details of the proof are deferred to Appendix B. ■

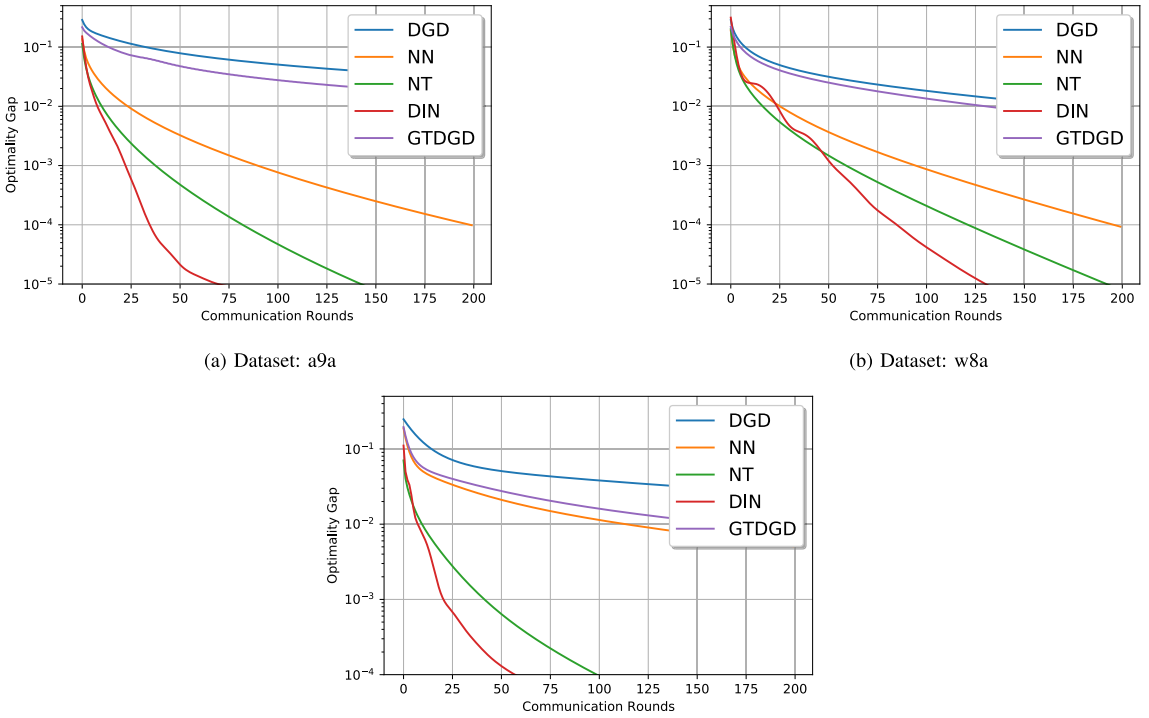
At this point, and considering the results in (40) and (41), we can present our third Lemma which gives the condition on ρ to ensure the convergence of the terms $\|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|^2$ and $\|\boldsymbol{\mu}^k - (\boldsymbol{\mu}^\star)^k\|^2$. To do so, we first define the vector $\boldsymbol{u} \in \mathbb{R}^{2Nd}$ and $\boldsymbol{G} \in \mathbb{R}^{Nd\times Nd}$ as

$$\boldsymbol{u} = \begin{bmatrix}\boldsymbol{\mu}\\ \boldsymbol{d}\end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix}\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{0} & 2\rho^2\delta_{\min}\boldsymbol{I}\end{bmatrix}, \quad (27)$$

where $\delta_{\min}$ is the minimum degree of the graph. Note that the sequence $\boldsymbol{u}^k$ combines the dual variable $\boldsymbol{\mu}^k$ and primal variable $\boldsymbol{d}^k$. Similarly, $\boldsymbol{u}^\star$ is defined as the concatenation of the optimal solutions $(\boldsymbol{\mu}^\star)^k$ and $(\boldsymbol{d}^\star)^k$. It is evident that $\|\boldsymbol{u}^k - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}}$ can be decomposed into $\|\boldsymbol{\mu}^k - (\boldsymbol{\mu}^\star)^k\|^2 + 2\rho^2\delta_{\min}\|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|^2$.

Lemma 3: Let $\frac{1}{2\delta_{min}} \le \rho \le \frac{1}{\sigma_{max}(\boldsymbol{A})}$. Then, the sequences $\|\boldsymbol{u}^{k-1} - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}} - \|\boldsymbol{u}^k - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}}$ of DIN satisfy

$$\begin{aligned}&\|\boldsymbol{u}^{k-1} - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}} - \|\boldsymbol{u}^k - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}}\\&\ge \frac{2\rho}{\mu+L}\|\nabla\phi^k(\boldsymbol{d}^k) - \nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2\end{aligned}$$



(a) Dataset: a9a

(b) Dataset: w8a

(c) Dataset: FHS heart disease

FIGURE 1. Optimality gap of DIN compared to baselines in terms of the number of communication rounds for a random topology using different datasets.

$$+ (2\rho^2\delta_{min} - \rho)\|\boldsymbol{d}^k - \boldsymbol{d}^{k-1}\|^2 + \|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|^2_{\frac{2\rho\mu L}{\mu+L}\boldsymbol{I}+\rho^2(1-\rho\sigma_{max}(\boldsymbol{A}))\boldsymbol{L}_-}. \quad (28)$$

where $\sigma_{max}(\boldsymbol{A})$ is the largest singular value of the extended incidence matrix $\boldsymbol{A}$.

Proof: The proof is provided in Appendix B where we use the result derived in Lemma 1. ■

According to Lemma 3, it is evident that the sequence $\|\boldsymbol{\mu}^k - (\boldsymbol{\mu}^\star)^k\|^2$ converges, indicating that $\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k$ converges at the same rate. In the subsequent theorem, we find that rate and demonstrate that the sequence $\|\boldsymbol{\mu}^k - (\boldsymbol{\mu}^\star)^k\|^2$ converges linearly.

Theorem 1: Assume $\gamma > 1$, and let $\sigma_{min}(\boldsymbol{A})$ denote the smallest non-zero singular value of the extended incidence matrix $\boldsymbol{A}$. Additionally, recall the definitions of the vector $\boldsymbol{u}$ and matrix $\boldsymbol{G}$ in (27). If the assumptions in (19) and (20) are satisfied, then the sequence $\|\boldsymbol{\mu}^k - (\boldsymbol{\mu}^\star)^k\|^2$ produced by DIN, stated in Algorithm 1, satisfies

$$\|\boldsymbol{u}^k - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}} \leq \frac{1}{1+\zeta}\|\boldsymbol{u}^{k-1} - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}}, \quad (29)$$

where the constant ζ is given by

$$\zeta = min\left\{\frac{2\rho\mu\sigma_{min}(\boldsymbol{A})}{\gamma L(\mu+L)}, \frac{(2\rho\delta_{min}-1)(\gamma-1)\sigma_{min}(\boldsymbol{A})}{2\gamma\rho\delta_{max}}, \frac{\mu L}{\rho\delta_{min}(\mu+L)}\right\} \quad (30)$$

Proof: The proof can be found in Appendix D. ■

Theorem 1 demonstrates that the DIN descent direction, $\boldsymbol{d}^k$, converges to the optimal Newton direction, $\boldsymbol{d}^\star$, with a linear rate of $\frac{1}{1+\zeta}$. While this approach utilizes an inexact Newton step, the guaranteed linear convergence ensures that $\boldsymbol{d}^k$ approaches the optimal descent direction $\boldsymbol{d}^\star$ in every iteration. To further establish the global convergence of DIN, i.e., $\boldsymbol{x}^k \to \boldsymbol{x}^\star$, one can utilize the global convergence analysis of inexact Newton methods in [35] and [37], which we leave as a subject of future work.

V. NUMERICAL EVALUATION

In this section, we conduct numerical experiments to evaluate the performance of our proposed algorithm DIN, against first- and second-order algorithms, DGD, GTDGD, Network Newton (NN) [29], and Newton Tracking (NT) [30], under different network topologies. We consider a binary classification problem using a regularized logistic regression.

A. EXPERIMENTAL SETUP

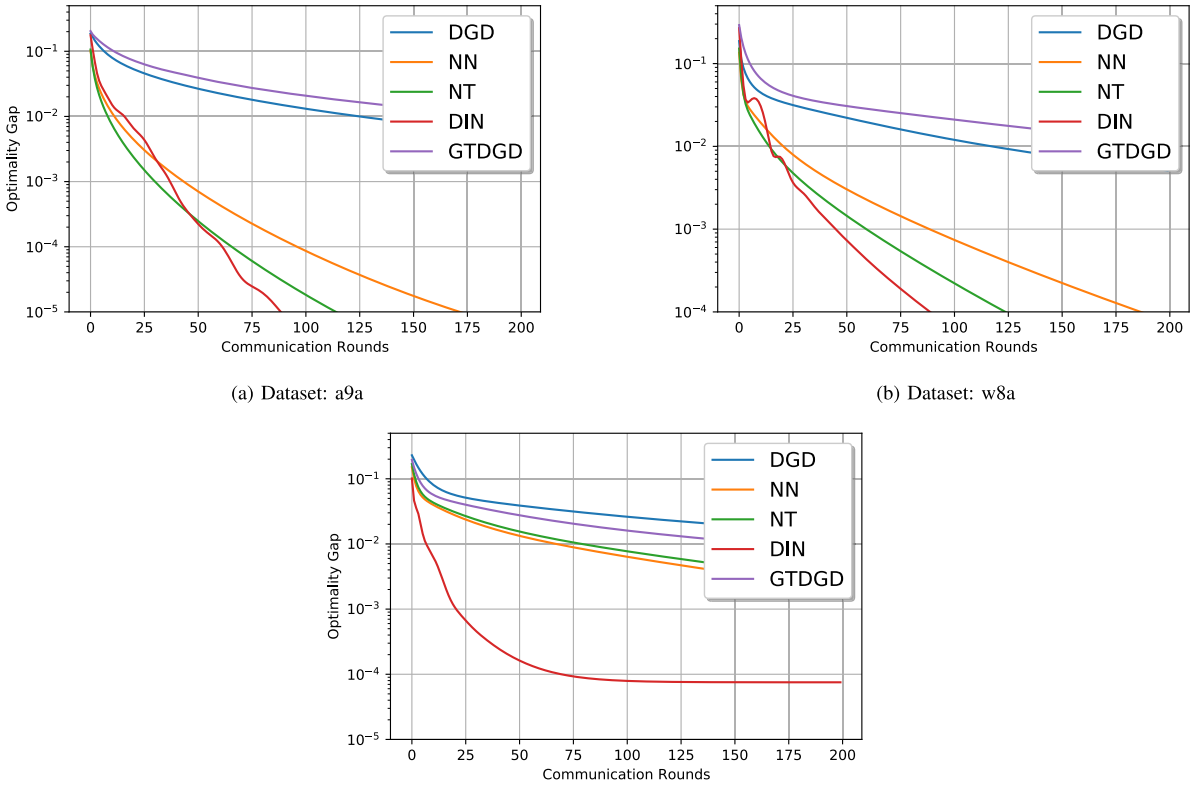
We consider the regularized logistic regression problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^d}\{f(\boldsymbol{x}) := \frac{1}{N}\sum_{i=1}^{N} f_i(\boldsymbol{x}) + \frac{\eta}{2}\|\boldsymbol{x}\|^2\}, \quad (31)$$

where the local loss function $f_i(\boldsymbol{x})$ is defined as

$$f_i(\boldsymbol{x}) = \frac{1}{m}\sum_{j=1}^{m}\log\left(1+\exp\left(-b_{ij}\boldsymbol{a}_{ij}^T\boldsymbol{x}\right)\right), \quad (32)$$

$\{\boldsymbol{a}_{ij}, b_{ij}\}_{j=1,\ldots,m}$ denote the data points at the i^{th} device ($i \in \{1,\ldots,N\}$), where m represents the number of data samples



(a) Dataset: a9a

(b) Dataset: w8a

(c) Dataset: FHS heart disease

FIGURE 2. Optimality gap versus the number of communication rounds for geometric network topologies using different datasets.

TABLE 2. Details of the datasets

Dataset	n	m	d	N
a9a	32560	407	109	80
w8a	49700	350	267	142
FHS	3565	118	15	30

of each device. A regularization parameter $\eta > 0$ is added to avoid overfitting and chosen to be equal to 10^{-3}.

We consider three real datasets: a9a and w8a which were taken from LibSVM [38] and FHS heart disease dataset [39]. The data is evenly split between N workers, which are connected with undirected edges of a given generated graph. The number of features of each dataset and the number of workers is depicted in Table 2. Two network topologies are implemented in the experiments: a binomial graph with edge creation probability $p = 0.4$, and a geometric graph with distance $d = 0.4$.

The energy footprint of the i^{th} device consumed during training consists of two parts, computation and communication components. The computation component E_c consists of the energy required to power up the hardware (e.g., CPUs, GPUs, Memories, etc.), while E_t represents the energy needed to transmit and receive bits between neighboring devices [40]. The total energy consumed by device i after t iterations can be written as

$$E_T(t) = E_c(t) + E_t(t), \tag{33}$$

with

$$E_c(t) = \sum_{k=1}^{t} e^k_{device,i} \quad \text{and} \quad E_t(t) = \sum_{k=1}^{t} \sum_{j\in\mathcal{N}_i} b(\boldsymbol{d}^k_i) e^k_{i,j}, \tag{34}$$

where $e^k_{device,i}$ is the computation energy consumed by device i to perform one iteration k, $b(\boldsymbol{d}^k_i)$ is the size of the inverse Hessian-gradient product vector in bits, and $e^k_{i,j}$ is the energy needed to transmit one bit from device i to neighbour j in the k^{th} iteration.

We conduct the experiment on an NVIDIA Jetson Dev Board [41], and we monitor the energy efficiency and the carbon emission using eco2AI python library [42]. The devices are randomly distributed over a 100×100 m^2 area, and we assume a digital communication link with a free-space path loss channel model. Hence, the maximum achievable rate $R = B\log_2(1 + \frac{P_t}{d^2_{i,j}BN_0})$, where B is the bandwidth, P_t is the transmission power, $d_{i,j}$ is the distance between transmitter i and receiver j, and N_0 is the noise spectral density. To find the maximum data rate between neighbouring devices and the energy consumed for transmission, we assume each device transmits at full power $P_t = 100$mW, $B = 2$MHz, $N_0 = 10^{-9}$W/Hz, and a 32-bit representation of transmitted elements.

To evaluate the performance of the aforementioned algorithms, we plot the optimality gap $f(\bar{\boldsymbol{x}}^k) - f(\boldsymbol{x}^\star)$ as a function of the number of communication rounds, where

TABLE 3. Computation/communication energy costs and corresponding carbon footprints for the a9a dataset for a target optimality gap 10^{-5}

Algorithm	Comp. Energy [J]	Comm. Energy [J]	Total Footprint [g-CO2-eq]
DIN	$1.32E-2$	17.01	$3.02E-4$
NT	$2.81E-2$	30.50	$5.43E-4$
NN	$10.41E-2$	49.40	$8.84E-4$
GTDGD	$7.61E-2$	120.54	$2.14E-3$
DGD	$8.10E-2$	128.23	$2.28E-3$

$x^\star$ and $f(x^\star)$ are pre-computed using standard Newton's method until convergence and $\bar{x}^k$ is the average model at iteration k. For hyperparameters tuning, we pick the parameters that lead to the best performance for each algorithm.

B. PERFORMANCE COMPARISON

Fig. 1 illustrates the optimality gap as a function of number of communication rounds in a decentralized network topology with a connection probability $p = 0.4$. We observe from Fig. 1 that DIN is the fastest, followed by NT, NN, GTDGD, and DGD for the three datasets. We clearly see in Fig. 1-(a-c) that DIN reaches the optimality gap of 10^{-5} within at least 50 communication rounds earlier than the fastest baseline (NT) for a9a and w8a datasets, and still reaches optimality gap of 10^{-4} earlier than (NT) for the FHS dataset. Since each algorithm has the same communication overhead per round, DIN is the most communication/energy efficient one; thanks to the fast convergence in terms of the number of communication rounds.

In Fig. 2, we investigate the performance of DIN with geometric network topology, in which every two devices are connected if they are located within a normalized distance $d = 0.4$. Each subfigure plots the optimality gap with respect to the number of communication rounds for the different datasets: a9a, w8a, and FHS. We observe from Fig. 2(a-c) that DIN converges faster than the considered baselines, although there is a degradation in the convergence speed compared to the random topology, as seen in Fig. 1, due to the increased sparsity of the network.

C. ENERGY-EFFICIENCY AND CARBON FOOTPRINT

In table 3, we report the energy consumption and the carbon footprint required by the four algorithms to achieve a 10^{-5} optimality gap using the a9a dataset. We observe that DGD's total energy consumption is the highest, and so is its carbon footprint. Although DGD is computationally less expensive, it requires a very large number of communication rounds to achieve the target optimality gap inducing high communication energy cost. On the other hand, NN consumes the highest computation energy since NN performs two matrix inversion operations in each communication round. Finally, DIN requires lower energy for both computation and communication due to its fast convergence while performing a single matrix inversion operation in each communication round.

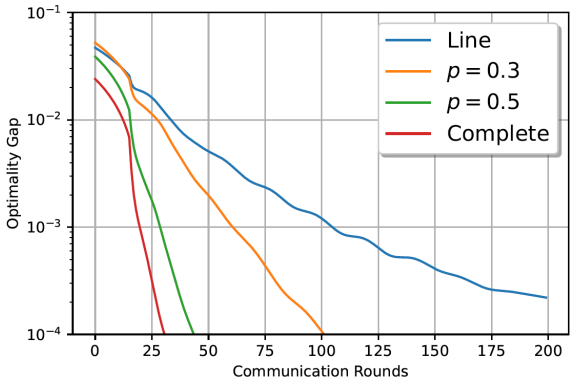


FIGURE 3. Effect of the network density on the DIN performance for the a9a dataset.

D. IMPACT OF THE GRAPH DENSITY

Finally, we investigate the effect of the graph density on the convergence speed of DIN. We use four topologies: the line graph, random graphs with $p \in \{0.3, 0.5\}$, and the complete graph using a9a dataset. The hyperparameters α and ρ are tuned to give the fastest convergence. Fig. 3 shows the optimality gap versus the number of communication rounds. We observe that the complete graph gives the fastest speed, whereas the line graph yields the slowest convergence among all topologies. Furthermore, when $p = 0.5$, DIN still achieves a comparable performance to the complete graph case indicating that DIN is still applicable in networks with limited connectivity.

VI. CONCLUSION

This paper presents a decentralized FL algorithm based on inexact Newton's method. Each client updates its model utilizing an approximate of the global inverse Hessian gradient product, which is calculated using its local function/data and shared approximate directions of its neighbors. By performing one Prox-PDA step, the proposed approach avoids sharing the Hessian of the device and thus ensures privacy. Furthermore, by only sharing a model-sized vector, DIN has the same per iteration communication efficiency as first-order methods, yet it is shown to be much faster and more energy-efficient. Numerical results show the supremacy of DIN over existing decentralized algorithms such as DGD, NN, and NT in solving the logistic regression problem. The convergence analysis shows that DIN the learned direction converges to the exact Newton direction. The utilization of quantization

for DIN and its applicability to non-convex settings are left as future work.

APPENDICES

A. PROOF OF LEMMA 1

From the assumption in (19), and (20) the objective function ϕ^k is strongly convex with a constant μ and has a Lipschitz continuous gradient with a constant L. As a result of the Lipschitz continuity assumption of the gradient of $\phi^k(\boldsymbol{d}^k)$, we can have

$$\frac{1}{L}\|\nabla\phi^k(\boldsymbol{d}^k)-\nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2 \leq (\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T(\nabla\phi^k(\boldsymbol{d}^k)-\nabla\phi^k((\boldsymbol{d}^\star)^k)) \quad (35)$$

Furthermore, from the strong convexity of ϕ in $\boldsymbol{d}$, we have

$$\mu\|\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k\|^2 \leq (\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T(\nabla\phi^k(\boldsymbol{d}^k)-\nabla\phi^k((\boldsymbol{d}^\star)^k)) \quad (36)$$

Multiplying (35) by μ, and (36) by L and summing both inequalities, we find a lower bound for the inner product $(\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T(\nabla\phi(\boldsymbol{d}^k)-\nabla\phi((\boldsymbol{d}^\star)^k))$

$$\frac{\mu L}{\mu+L}\|\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k\|^2+\frac{\mu}{L(\mu+L)}\|\nabla\phi^k(\boldsymbol{d}^k)-\nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2 \leq (\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T(\nabla\phi^k(\boldsymbol{d}^k)-\nabla\phi^k((\boldsymbol{d}^\star)^k)) \quad (37)$$

B. PROOF OF LEMMA 2

Consider the feasibility condition in (23), we notice that the optimal solution $(\boldsymbol{d}^\star)^k = [(\boldsymbol{d}^\star)^k,(\boldsymbol{d}^\star)^k,\ldots,(\boldsymbol{d}^\star)^k]^T$ lies in null$\{\boldsymbol{L}_-\}$. Thus we have

$$\rho\boldsymbol{L}_-(\boldsymbol{d}^\star)^k=0 \quad (38)$$

Subtracting the equations (24) and (38) from (11) we obtain

$$\nabla\phi^k(\boldsymbol{d}^k)-\nabla\phi^k((\boldsymbol{d}^\star)^k)+\boldsymbol{\lambda}^{k-1}-(\boldsymbol{\lambda}^\star)^k+2\rho\boldsymbol{D}\boldsymbol{d}^k-\rho\boldsymbol{L}_-(\boldsymbol{d}^\star)^k-\rho\boldsymbol{L}_+\boldsymbol{d}^{k-1}=\boldsymbol{0}. \quad (39)$$

Rearranging the terms in (14) and using (38), we have

$$\boldsymbol{\lambda}^k-\boldsymbol{\lambda}^{k-1}-\rho\boldsymbol{L}_-(\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)=\boldsymbol{0} \quad (40)$$

Substituting the term $\boldsymbol{\lambda}^{k-1}$ into (39), we obtain

$$\nabla\phi^k(\boldsymbol{d}^k)-\nabla\phi^k((\boldsymbol{d}^\star)^k)+\boldsymbol{\lambda}^k-(\boldsymbol{\lambda}^\star)^k+2\rho\boldsymbol{D}\boldsymbol{d}^k-\rho\boldsymbol{L}_-\boldsymbol{d}^k-\rho\boldsymbol{L}_+\boldsymbol{d}^{k-1}=\boldsymbol{0}. \quad (41)$$

Finally, using $2\boldsymbol{D}=\boldsymbol{L}_-+\boldsymbol{L}_+$, we can write (26) as

$$\nabla\phi^k(\boldsymbol{d}^k)-\nabla\phi^k((\boldsymbol{d}^\star)^k)+\boldsymbol{\lambda}^k-(\boldsymbol{\lambda}^\star)^k+\rho\boldsymbol{L}_+(\boldsymbol{d}^k-\boldsymbol{d}^{k-1})=\boldsymbol{0}. \quad (42)$$

C. PROOF OF LEMMA 3

According on the result in (42), we can replace the term $\nabla\phi^k(\boldsymbol{d}^k)-\nabla\phi^k((\boldsymbol{d}^\star)^k)$ in the left hand side of (37) by $-(\boldsymbol{\lambda}^k-(\boldsymbol{\lambda}^\star)^k)-\rho\boldsymbol{L}_+(\boldsymbol{d}^k-\boldsymbol{d}^{k-1})$ to get

$$\frac{\mu L}{\mu+L}\|\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k\|^2+\frac{\mu}{L(\mu+L)}\|\nabla\phi^k(\boldsymbol{d}^k)-\nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2 \leq -(\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T\left(\boldsymbol{\lambda}^k-(\boldsymbol{\lambda}^\star)^k\right)-\rho(\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T\boldsymbol{L}_+(\boldsymbol{d}^k-\boldsymbol{d}^{k-1}) \quad (43)$$

Using the fact that $\boldsymbol{\lambda}^k=\boldsymbol{A}^T\boldsymbol{\mu}^k$, we can write (40) as

$$\frac{1}{\rho}(\boldsymbol{\mu}^k-\boldsymbol{\mu}^{k-1})^T=(\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T\boldsymbol{A}^T \quad (44)$$

Substituting (44) in (43) and multiplying both sides by 2 we have

$$\frac{2\rho\mu L}{\mu+L}\|\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k\|^2+\frac{2\rho\mu}{L(\mu+L)}\|\nabla\phi^k(\boldsymbol{d}^k)-\nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2 \leq -2(\boldsymbol{\mu}^k-\boldsymbol{\mu}^{k-1})^T(\boldsymbol{\mu}^k-\boldsymbol{\mu}^\star)-2\rho^2(\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T\boldsymbol{L}_+(\boldsymbol{d}^k-\boldsymbol{d}^{k-1}) \quad (45)$$

Knowing that for any three vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ we can write

$$2(\boldsymbol{a}-\boldsymbol{b})^T(\boldsymbol{a}-\boldsymbol{c})=\|\boldsymbol{a}-\boldsymbol{b}\|^2+\|\boldsymbol{a}-\boldsymbol{c}\|^2-\|\boldsymbol{b}-\boldsymbol{c}\|^2 \quad (46)$$

Setting $\boldsymbol{a}=\boldsymbol{\mu}^k$, $\boldsymbol{b}=\boldsymbol{\mu}^{k-1}$, and $\boldsymbol{c}=(\boldsymbol{\mu}^\star)^k$, we can write the inner product $2(\boldsymbol{\mu}^k-\boldsymbol{\mu}^{k-1})^T(\boldsymbol{\mu}^k-\boldsymbol{\mu}^\star)$ as $\|\boldsymbol{\mu}^k-\boldsymbol{\mu}^{k-1}\|^2+\|\boldsymbol{\mu}^k-(\boldsymbol{\mu}^\star)^k\|^2-\|\boldsymbol{\mu}^{k-1}-(\boldsymbol{\mu}^\star)^k\|^2$

$$\frac{2\rho\mu L}{\mu+L}\|\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k\|^2+\frac{2\rho\mu}{L(\mu+L)}\|\nabla\phi^k(\boldsymbol{d}^k)-\nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2 \leq -\|\boldsymbol{\mu}^k-\boldsymbol{\mu}^{k-1}\|^2-\|\boldsymbol{\mu}^k-(\boldsymbol{\mu}^\star)^k\|^2+\|\boldsymbol{\mu}^{k-1}-(\boldsymbol{\mu}^\star)^k\|^2-2\rho^2(\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T\boldsymbol{L}_+(\boldsymbol{d}^k-\boldsymbol{d}^{k-1}) \quad (47)$$

To begin with, let's simplify the last term $(\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T\boldsymbol{L}_+(\boldsymbol{d}^k-\boldsymbol{d}^{k-1})$ by expressing $\boldsymbol{L}+$ as $2\boldsymbol{D}-\boldsymbol{L}_-$.

$$\begin{aligned}&(\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T\boldsymbol{L}_+(\boldsymbol{d}^k-\boldsymbol{d}^{k-1})\\&=2(\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T\boldsymbol{D}(\boldsymbol{d}^k-\boldsymbol{d}^{k-1})\\&\quad-(\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T\boldsymbol{L}_-(\boldsymbol{d}^k-\boldsymbol{d}^{k-1})\\&=2(\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T\boldsymbol{D}(\boldsymbol{d}^k-\boldsymbol{d}^{k-1})\\&\quad-\frac{1}{\rho}(\boldsymbol{\lambda}^k-\boldsymbol{\lambda}^{k-1})^T(\boldsymbol{d}^k-\boldsymbol{d}^{k-1})\end{aligned} \quad (48)$$

Now using the identity in (46), we can expand (48) further given that $\boldsymbol{a}=\boldsymbol{0}$

$$\begin{aligned}&(\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T\boldsymbol{L}_+(\boldsymbol{d}^k-\boldsymbol{d}^{k-1})\\&=2(\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k)^T\boldsymbol{D}(\boldsymbol{d}^k-\boldsymbol{d}^{k-1})\\&\quad-\frac{1}{2\rho}\|\boldsymbol{\lambda}^k-\boldsymbol{\lambda}^{k-1}\|^2-\frac{1}{2\rho}\|\boldsymbol{d}^k-\boldsymbol{d}^{k-1}\|^2\\&\quad+\frac{1}{2\rho}\|(\boldsymbol{\lambda}^k-\boldsymbol{\lambda}^{k-1})-(\boldsymbol{d}^k-\boldsymbol{d}^{k-1})\|^2\end{aligned} \quad (49)$$

Substituting (49) in (47) we can write it as

$$\frac{2\rho\mu L}{\mu+L}\|\boldsymbol{d}^k-(\boldsymbol{d}^\star)^k\|^2+\frac{2\rho\mu}{L(\mu+L)}\|\nabla\phi^k(\boldsymbol{d}^k)-\nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2 \leq -\|\boldsymbol{\mu}^k-\boldsymbol{\mu}^{k-1}\|^2-\|\boldsymbol{\mu}^k-(\boldsymbol{\mu}^\star)^k\|^2+\|\boldsymbol{\mu}^{k-1}-(\boldsymbol{\mu}^\star)^k\|^2$$

$$- 4\rho^2(\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k)^T\boldsymbol{D}(\boldsymbol{d}^k - \boldsymbol{d}^{k-1}) + \rho\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|^2 + \rho\|\boldsymbol{d}^k - \boldsymbol{d}^{k-1}\|^2 - \rho\|(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}) - (\boldsymbol{d}^k - \boldsymbol{d}^{k-1})\|^2 \tag{50}$$

Furthermore, we can expand the term $-4\rho^2(\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k)^T\boldsymbol{D}(\boldsymbol{d}^k - \boldsymbol{d}^{k-1})$ in (50) following the same way

$$\begin{aligned}
&- 4\rho^2(\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k)^T\boldsymbol{D}(\boldsymbol{d}^k - \boldsymbol{d}^{k-1}) \\
&\leq -4\rho^2\delta_{\min}(\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k)^T(\boldsymbol{d}^k - \boldsymbol{d}^{k-1}) \\
&= -2\rho^2\delta_{\min}\Big(\|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|^2 + \|\boldsymbol{d}^k \\
&\quad -\boldsymbol{d}^{k-1}\|^2 - \|\boldsymbol{d}^{k-1} - (\boldsymbol{d}^\star)^k\|^2\Big)
\end{aligned} \tag{51}$$

Finally, substituting (51) in (50), we can have the following

$$\begin{aligned}
&\frac{2\rho\mu L}{\mu+L}\|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|^2 + \frac{2\rho\mu}{L(\mu+L)}\|\nabla\phi^k(\boldsymbol{d}^k) - \nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2 \\
&\leq -\|\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1}\|^2 - \|\boldsymbol{\mu}^k - (\boldsymbol{\mu}^\star)^k\|^2 + \|\boldsymbol{\mu}^{k-1} - (\boldsymbol{\mu}^\star)^k\|^2 \\
&\quad - 2\rho^2\delta_{\min}\|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|^2 - 2\rho^2\delta_{\min}\|\boldsymbol{d}^k - \boldsymbol{d}^{k-1}\|^2 \\
&\quad + 2\rho^2\delta_{\min}\|\boldsymbol{d}^{k-1} - (\boldsymbol{d}^\star)^k\|^2 + \rho\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|^2 \\
&\quad + \rho\|\boldsymbol{d}^k - \boldsymbol{d}^{k-1}\|^2
\end{aligned} \tag{52}$$

We can write the term $\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|^2$ using (44) as $\|\boldsymbol{A}^T(\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1})\|^2$ which has an upper bound of $\sigma_{\max}(\boldsymbol{A})\|\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1}\|^2$,where $\sigma_{\max}(\boldsymbol{A})$ is the largest singular value of the extended incidence matrix $\boldsymbol{A}$ and by regrouping the terms in (52) we have

$$\begin{aligned}
&\frac{2\rho\mu L}{\mu+L}\|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|^2 + \frac{2\rho\mu}{L(\mu+L)}\|\nabla\phi^k(\boldsymbol{d}^k) - \nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2 \\
&\leq \|\boldsymbol{\mu}^{k-1} - (\boldsymbol{\mu}^\star)^k\|^2 - \|\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1}\|^2_{(1-\rho\sigma_{\max}(\boldsymbol{A}))\boldsymbol{I}} \\
&\quad - \|\boldsymbol{\mu}^k - (\boldsymbol{\mu}^\star)^k\|^2 \\
&\quad - 2\rho^2\delta_{\min}\|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|^2 + 2\rho^2\delta_{\min}\|\boldsymbol{d}^{k-1} - (\boldsymbol{d}^\star)^k\|^2 \\
&\quad - (2\rho^2\delta_{\min} - \rho)\|\boldsymbol{d}^k - \boldsymbol{d}^{k-1}\|^2
\end{aligned} \tag{53}$$

Using the definition of the variables $\boldsymbol{u}$ and $\boldsymbol{G}$ in (27), we can express $\|\boldsymbol{\mu}^{k-1} - (\boldsymbol{\mu}^\star)^k\|^2 - \|\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1}\|^2 + 2\rho^2\delta_{\min}\|\boldsymbol{d}^{k-1} - (\boldsymbol{d}^\star)^k\|^2 - 2\rho^2\delta_{\min}\|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|^2$ by $\|\boldsymbol{u}^{k-1} - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}} - \|\boldsymbol{u}^k - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}}$. Furthermore, the term $\|\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1}\|^2_{(1-\rho\sigma_{\max}(\boldsymbol{A}))\boldsymbol{I}}$ can be written as $\|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|^2_{\rho^2(1-\rho\sigma_{\max}(\boldsymbol{A}))\boldsymbol{L}_-}$ using (44). Finally, we can rewrite (53) as

$$\begin{aligned}
&\frac{2\rho\mu L}{\mu+L}\|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|^2 + \frac{2\rho\mu}{L(\mu+L)}\|\nabla\phi^k(\boldsymbol{d}^k) - \nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2 \\
&\leq \|\boldsymbol{u}^{k-1} - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}} - \|\boldsymbol{u}^k - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}} - (2\rho^2\delta_{\min} - \rho)\|\boldsymbol{d}^k \\
&\quad - \boldsymbol{d}^{k-1}\|^2 - \|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|^2_{\rho^2(1-\rho\sigma_{\max}(\boldsymbol{A}))\boldsymbol{L}_-}
\end{aligned} \tag{54}$$

By rearranging the terms in (54), we get a lower bound for the difference $\|\boldsymbol{u}^{k-1} - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}} - \|\boldsymbol{u}^k - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}}$,

$$\begin{aligned}
&\|\boldsymbol{u}^{k-1} - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}} - \|\boldsymbol{u}^k - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}} \\
&\geq \frac{2\rho\mu}{L(\mu+L)}\|\nabla\phi^k(\boldsymbol{d}^k) - \nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2 \\
&\quad + (2\rho^2\delta_{\min} - \rho)\|\boldsymbol{d}^k - \boldsymbol{d}^{k-1}\|^2 \\
&\quad + \|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|^2_{\frac{2\rho\mu L}{\mu+L}\boldsymbol{I} + \rho^2(1-\rho\sigma_{\max}(\boldsymbol{A}))\boldsymbol{L}_-}
\end{aligned} \tag{55}$$

D. PROOF OF THEOREM 1

The result in (55) provides a lower bound for $\|\boldsymbol{u}^{k-1} - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}} - \|\boldsymbol{u}^k - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}}$. We need to show that for a positive constant ζ we have $\|\boldsymbol{u}^{k-1} - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}} - \|\boldsymbol{u}^k - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}} \geq \zeta\|\boldsymbol{u}^k - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}}$. Therefore the inequality $\|\boldsymbol{u}^k - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}} \leq \frac{1}{1+\zeta}\|\boldsymbol{u}^{k-1} - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}}$ is satisfied if we can show that the lower bound in (55) is greater than $\zeta\|\boldsymbol{u}^k - \boldsymbol{u}^\star\|^2_{\boldsymbol{G}}$ or we write

$$\begin{aligned}
&\zeta\|\boldsymbol{\mu}^k - \boldsymbol{\mu}^\star\| + 2\zeta\rho^2\delta_{\min}\|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\| \\
&\leq \frac{2\rho\mu}{L(\mu+L)}\|\nabla\phi^k(\boldsymbol{d}^k) - \nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2 \\
&\quad + (2\rho^2\delta_{\min} - \rho)\|\boldsymbol{d}^k - \boldsymbol{d}^{k-1}\|^2 \\
&\quad + \|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|^2_{\frac{2\rho\mu L}{\mu+L}\boldsymbol{I} + \rho^2(1-\rho\sigma_{\max}(\boldsymbol{A}))\boldsymbol{L}_-}
\end{aligned} \tag{56}$$

To show that the inequality in (55) holds for some $\zeta > 0$, we need to first find an upper bound for the squared norm $\|\boldsymbol{\mu}^k - \boldsymbol{u}^\star\|$ in terms of $\|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|$ and $\|\nabla\phi^k(\boldsymbol{d}^k) - \nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2$ in the right-hand side of (56). Observing the definition of $\boldsymbol{\lambda}^k$ as $\boldsymbol{A}^T\boldsymbol{\mu}^k$ we can write

$$\|\boldsymbol{A}^T(\boldsymbol{\mu}^k - (\boldsymbol{\mu}^\star)^k)\| \leq \sigma_{\min}(\boldsymbol{A})\|\boldsymbol{\mu}^k - (\boldsymbol{\mu}^\star)^k\| \tag{57}$$

Furthermore, Considering the expression in (26) and (57) we can show that the term $\|\boldsymbol{\mu}^k - (\boldsymbol{\mu}^\star)^k\|$ is bounded from above by

$$\begin{aligned}
\|\boldsymbol{\mu}^k - (\boldsymbol{\mu}^\star)^k\|^2 \leq \frac{\gamma}{\sigma_{\min}(\boldsymbol{A})}\|\nabla\phi^k(\boldsymbol{d}^k) - \nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2 \\
+ \frac{2\gamma\rho^2\delta_{\max}}{(\gamma-1)\sigma_{\min}(\boldsymbol{A})}\|(\boldsymbol{d}^k - \boldsymbol{d}^{k-1})\|^2.
\end{aligned} \tag{58}$$

where $\gamma > 1$ is a tuning parameter. For (42) to satisfy the inequality in (56), we need to show that,

$$\begin{aligned}
&\frac{2\rho\mu}{L(\mu+L)}\|\nabla\phi^k(\boldsymbol{d}^k) - \nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2 \\
&\quad + (2\rho^2\delta_{\min} - \rho)\|\boldsymbol{d}^k - \boldsymbol{d}^{k-1}\|^2 \\
&\quad + \|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|^2_{\frac{2\rho\mu L}{\mu+L}\boldsymbol{I} + \rho^2(1-\rho\sigma_{\max}(\boldsymbol{A}))\boldsymbol{L}_-} \\
&\geq \frac{\zeta\gamma}{\sigma_{\min}(\boldsymbol{A})}\|\nabla\phi^k(\boldsymbol{d}^k) - \nabla\phi^k((\boldsymbol{d}^\star)^k)\|^2 \\
&\quad + \frac{2\zeta\gamma\rho^2\delta_{\max}}{(\gamma-1)\sigma_{\min}(\boldsymbol{A})}\|(\boldsymbol{d}^k - \boldsymbol{d}^{k-1})\|^2 \\
&\quad + 2\zeta\rho^2\delta_{\min}\|\boldsymbol{d}^k - (\boldsymbol{d}^\star)^k\|.
\end{aligned} \tag{59}$$

To ensure that (59) holds and consequently enable (56), we simply need to make sure the existence of $\zeta > 0$

$$\begin{aligned}
&\frac{2\rho\mu}{L(\mu+L)} \geq \frac{\zeta\gamma}{\sigma_{\min}(\boldsymbol{A})}, \quad 2\rho\delta_{\min} - 1 \geq \frac{2\zeta\gamma\rho\delta_{\max}}{(\gamma-1)\sigma_{\min}(\boldsymbol{A})}, \\
&\frac{2\rho\mu L}{\mu+L}\boldsymbol{I} + \rho^2(1-\rho\sigma_{\max}(\boldsymbol{A}))\boldsymbol{L}_- \succcurlyeq 2\zeta\rho^2\delta_{\min}
\end{aligned} \tag{60}$$

To satisfy the condition in (60), we choose ζ as

$$\zeta = \min\left\{\frac{2\rho\mu\sigma_{\min}(\boldsymbol{A})}{\gamma L(\mu+L)}, \frac{(2\rho\delta_{\min}-1)(\gamma-1)\sigma_{\min}(\boldsymbol{A})}{2\gamma\rho\delta_{\max}}, \frac{\mu L}{\rho\delta_{\min}(\mu+L)}\right\} \tag{61}$$

which guarantees that

$$\|\boldsymbol{u}^k - \boldsymbol{u}^\star\|_{\boldsymbol{G}}^2 \leq \frac{1}{1+\zeta}\|\boldsymbol{u}^{k-1} - \boldsymbol{u}^\star\|_{\boldsymbol{G}}^2 \tag{62}$$

REFERENCES

[1] S. Nabavi, J. Zhang, and A. Chakrabortty, "Distributed optimization algorithms for wide-area oscillation monitoring in power systems using interregional PMU-PDC architectures," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2529–2538, Sep. 2015.

[2] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, Jun. 2021.

[3] A. Chavez, A. Moukas, and P. Maes, "Challenger: A multi-agent system for distributed resource allocation," in *Proc. 1st Int. Conf. Auto. Agents (AGENTS)*, 1997, pp. 323–331.

[4] A. M. Elbir, B. Soner, S. Coleri, D. Gunduz, and M. Bennis, "Federated learning in vehicular networks," in *Proc. IEEE Int. Medit. Conf. Commun. Netw. (MeditCom)*, Athens, Greece, Sep. 2022, pp. 72–77.

[5] S. Liu, P.-Y. Chen, B. Kailkhura, G. Zhang, A. O. Hero III, and P. K. Varshney, "A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications," *IEEE Signal Process. Mag.*, vol. 37, no. 5, pp. 43–54, Sep. 2020.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, vol. 54, A. Singh and J. Zhu. Eds., Fort Lauderdale, FL, USA, 2017, pp. 1273–1282.

[7] R. Zhang and J. Kwok, "Asynchronous distributed ADMM for consensus optimization," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1701–1709.

[8] X. Yin, B. W.-H. Ng, J. He, Y. Zhang, and D. Abbott, "Accurate image analysis of the retina using Hessian matrix and binarisation of thresholded entropy with application of texture mapping," *PLoS ONE*, vol. 9, no. 4, Apr. 2014, Art. no. e95943.

[9] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.

[10] S. Pu and A. Nedić, "Distributed stochastic gradient tracking methods," *Math. Program.*, vol. 187, pp. 409–457, May 2021.

[11] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM J. Optim.*, vol. 26, no. 3, pp. 1835–1854, Jan. 2016.

[12] M. Hong, D. Hajinezhad, and M. Zhao, "Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1529–1538.

[13] A. Ruszczynski, *Nonlinear Optimization*. Princeton, NJ, USA: Princeton Univ. Press, 2011.

[14] C. Guille-Escuret, M. Girotti, B. Goujaud, and I. Mitliagkas, "A study of condition numbers for first-order optimization," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 1261–1269.

[15] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. Vincent Poor, "Fast-convergent federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 201–218, Jan. 2021.

[16] J. Wang, V. Tantia, N. Ballas, and M. Rabbat, "SlowMo: Improving communication-efficient distributed SGD with slow momentum," 2019, *arXiv:1910.00643*.

[17] W. Liu, L. Chen, Y. Chen, and W. Zhang, "Accelerating federated learning via momentum gradient descent," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 8, pp. 1754–1766, Aug. 2020.

[18] S. Reddi et al., "Adaptive federated optimization," 2020, *arXiv:2003.00295*.

[19] A. Elgabli, J. Park, A. S. Bedi, C. B. Issaid, M. Bennis, and V. Aggarwal, "Q-GADMM: Quantized group ADMM for communication efficient decentralized machine learning," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 164–181, Jan. 2021.

[20] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2021–2031.

[21] C. B. Issaid, A. Elgabli, J. Park, M. Bennis, and M. Debbah, "Communication efficient decentralized learning over bipartite graphs," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 4150–4167, Jun. 2022.

[22] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1131–1146, May 2014.

[23] R. Xin, S. Kar, and U. A. Khan, "Gradient tracking and variance reduction for decentralized optimization and machine learning," 2020, *arXiv:2002.05373*.

[24] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 1322–1333.

[25] M. Safaryan, R. Islamov, X. Qian, and P. Richtárik, "FedNL: Making Newton-type methods applicable to federated learning," 2021, *arXiv:2106.02969*.

[26] A. Elgabli, C. B. Issaid, A. S. Bedi, K. Rajawat, M. Bennis, and V. Aggarwal, "FedNew: A communication-efficient and privacy-preserving Newton-type method for federated learning," in *Proc. ICML*, 2022, pp. 5861–5877.

[27] Y. Qu et al., "Decentralized federated learning for UAV networks: Architecture, challenges, and opportunities," *IEEE Netw.*, vol. 35, no. 6, pp. 156–162, 2021.

[28] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1146–1159, Feb. 2020.

[29] A. Mokhtari, Q. Ling, and A. Ribeiro, "Network Newton distributed optimization methods," *IEEE Trans. Signal Process.*, vol. 65, no. 1, pp. 146–161, Jan. 2017.

[30] J. Zhang, Q. Ling, and A. M. So, "A Newton tracking algorithm with exact linear convergence for decentralized consensus optimization," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 7, pp. 346–358, 2021.

[31] M. Eisen, A. Mokhtari, and A. Ribeiro, "Decentralized quasi-Newton methods," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2613–2628, May 2017.

[32] J. Zhang, H. Liu, A. M. So, and Q. Ling, "Variance-reduced stochastic quasi-Newton methods for decentralized learning," *IEEE Trans. Signal Process.*, vol. 71, pp. 311–326, 2023.

[33] A. Ghalkha, C. Ben Issaid, A. Elgabli, and M. Bennis, "DIN: A decentralized inexact Newton algorithm for consensus optimization," in *Proc. IEEE Int. Conf. Commun.*, May 2023, pp. 4391–4396.

[34] R. Islamov, X. Qian, and P. Richtárik, "Distributed second order methods with fast rates and compressed communication," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 4617–4628.

[35] U. Marteau-Ferey, F. Bach, and A. Rudi, "Globally convergent Newton methods for ill-conditioned generalized self-concordant losses," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–7.

[36] K. Mishchenko, "Regularized Newton method with global $o(1/k^2)$ convergence," 2021, *arXiv:2112.02089*.

[37] S. P. Karimireddy, S. U. Stich, and M. Jaggi, "Global linear convergence of Newton's method without strong-convexity or Lipschitz gradients," 2018, *arXiv:1806.00413*.

[38] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, May 2011, doi: 10.1145/1961189.1961199.

[39] A. Bhardwaj. (2022). *Framingham Heart Study Dataset*. [Online]. Available: https://www.kaggle.com/dsv/3493583

[40] S. Savazzi, V. Rampa, S. Kianoush, and M. Bennis, "An energy and carbon footprint analysis of distributed and federated learning," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 1, pp. 248–264, Mar. 2023.

[41] S. Cass, "Nvidia makes it easy to embed AI: The Jetson nano packs a lot of machine-learning power into DIY projects–[hands on]," *IEEE Spectr.*, vol. 57, no. 7, pp. 14–16, Jul. 2020.

[42] S. Budennyy et al., "Eco2AI: Carbon emissions tracking of machine learning models as the first step towards sustainable AI," 2022, *arXiv:2208.00406*.

ABDULMOMEN GHALKHA received the B.Sc. degree (Hons.)in telecommunications and electronics engineering from the University of Tripoli in 2019 and the M.Sc. degree in wireless communication engineering from the University of Oulu, Finland, where he is currently pursuing the Ph.D. degree. In 2021, he joined, as a Research Assistant, the Centre for Wireless Communication (CWC), University of Oulu. His research interests include distributed machine learning and machine learning applications for wireless communications.

CHAOUKI BEN ISSAID received the Diplôme d'Ingénieur degree from the l'École Polytechnique de Tunisie, La Marsa, Tunisia, in 2013, the M.Sc. degree in applied mathematics and computational science and the Ph.D. degree in statistics from the King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia, in 2015 and 2019, respectively. He is a Senior Researcher and an Adjunct Professor (docent) at the University of Oulu, Finland. Before that, he was a Post-Doctoral Researcher at the Centre for Wireless Communications, University of Oulu, from 2020 to 2023. His current research interests include communication-efficient distributed machine learning with applications to wireless communication.

ANIS ELGABLI (Senior Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from the University of Tripoli, Libya, in 2004, the M.Eng. degree from UKM, Malaysia, in 2007, and the M.Sc. and Ph.D. degrees from the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, in 2015 and 2018, respectively. He is currently an Assistant Professor with the ISE Department, KFUPM, affiliated with the Interdisciplinary Research Center for Communication Systems and Sensing (IRC-CSS), and guest affiliated with the Interdisciplinary Research Center for Intelligent Secure Systems (IRC-ISS). Before that, he spent five years as a Post-Doctoral Researcher with the Centre for Wireless Communications, University of Oulu, Finland. His main research interests include distributed optimization and machine learning, heterogeneous networks, radio resource management, vehicular communications, and video streaming. He was a recipient of the Best Paper Award in HotSpot Workshop, in 2018 (Infocom 2018), and the most JUFO points in 2020 at the Center of Wireless Communication, University of Oulu.

MEHDI BENNIS (Fellow, IEEE) is a Professor at the Centre for Wireless Communications, University of Oulu, Finland, Academy of Finland Research Fellow and Head of the intelligent connectivity and networks/systems group (ICON). He has published more than 200 research papers in international conferences, journals and book chapters. His main research interests are in radio resource management, heterogeneous networks, game theory, and distributed machine learning in 5G networks and beyond. He has been the recipient of several prestigious awards including the 2015 Fred W. Ellersick Prize from the IEEE Communications Society, the 2016 Best Tutorial Prize from the IEEE Communications Society, the 2017 EURASIP Best paper Award for the Journal of Wireless Communications and Networks, the all-University of Oulu award for research, the 2019 IEEE ComSoc Radio Communications Committee Early Achievement Award and the 2020 Clarviate Highly Cited Researcher by the Web of Science. He is an editor of IEEE TCOM and Specialty Chief Editor for *Data Science for Communications in the Frontiers in Communications and Networks journal*.