Hybrid Decentralized Data Analytics in Edge Computing Empowered IoT Networks

Liang Zhao, Fangyu Li, and Maria Valero

***Abstract*—Edge computing is emerging as a new infrastructure for Internet of Things (IoT) networks by placing computation and analytics near to where data is generated. This paper presents a novel data analytics framework for edge computing. The framework is based on a new decentralized algorithm, which enables all the nodes to obtain the global optimal model without sharing raw data. The resulting scheme executes in a hybrid mode: local IoT nodes send computed information to edge nodes. The edge nodes cooperate with each other by exchanging analytics with their neighbors only. The presenting approach is analyzed and evaluated on various applications and the experiment results demonstrate the effectiveness of the proposed methodology in providing fast data analytics to edge computing infrastructure.**

***Index Terms*—Edge computing, Internet of Things, Decentralized algorithm, Data analytics.**

I. Introduction

THE Internet of Things (IoT) has the potential to represent the next evolution of the Internet as we advance by extracting and accumulating knowledge from huge volumes of data collected by IoT devices [1]. With the help of Cloud, many IoT applications nowadays simply transmit all the raw data into Cloud for processing and analytics. However, this approach has several limitations. First, the data is too much. With the deployment and development of IoT, the data volumes generated by IoT devices is increasing, and it may not be possible to move all the raw data over the network to Cloud due to bandwidth limitations [2]. Second, the latency is high. For certain time-sensitive applications, this Cloud-based solution is prohibited due to the delay caused by moving, processing, and analyzing the data in Cloud. Third, limited privacy. In domains such as healthcare, the medical data cannot be transferred and thus this "collecting and then processing" approach is not suitable for privacy-centric applications.

Recently, the infrastructure of edge computing has been proposed. The idea is to move computation and analytics close to where data is generated. This architecture is promising in addressing some of the concerns aforementioned, for example, reduce the latency by offloading the computation into edge nodes [3], [4]. In addition, some recent studies focusing on resource allocation and energy efficiency of computation offloading in edge computing have been conducted in literature [5], [6]. However, the privacy concern remains unsolved and is left as an open issue in these works as raw data created by IoT devices are still transferred to edge nodes. As a possible solution, Federated Learning (FL), which is a new paradigm in data analytics and machine learning can be applied to IoT devices to jointly learn a model without sharing their raw data [7]. But the limitation lies in its parameter-server architecture that a central server (e.g. remote Cloud) exists for aggregating the parameter estimates from IoT devices and thus it is vulnerable to single point of failure. That is, if the server is down, the data analytics process will be ceased. Considering the points above, a natural question is: can we design a data analytics scheme that can fit into the infrastructure of edge computing and capable of coping with all the aforementioned issues? To answer this question, we propose a new decentralized algorithm and adapt it into a hybrid protocol for implementation in edge computing empowered IoT networks. In our protocol, IoT nodes transfer their gradient information to their corresponding edge nodes according to proximity, and then those edge nodes collaborate to obtain the global optimal solution by exchanging their estimates with immediate neighbors. The proposed framework does not require raw data sharing in the entire analytics process. Moreover, there is no central fusion center and certain node's failure will not prevent other nodes from performing the analytics.

The structure of this paper is as follows. Section II discusses the related work. Section III describes the problem formulation. Section IV presents the design of the proposed decentralized algorithm and its convergence properties. Section V describes the decentralized algorithm based hybrid protocol for the edge computing infrastructure. Section VI demonstrates the evaluation of the proposed protocol on two applications. Finally, Section VII presents our conclusion.

II. Related Work

In decentralized computing paradigms, each node holds an objective function privately known and collaborates only with its immediate neighbors to exchange information for the global objective. Based on the operating mode of each node in computation and communication, decentralized algorithms can be categorized into synchronous or asynchronous. A series of synchronous algorithms for solving general convex optimization problems have been proposed in literature [8]–[13]. However, each node needs to wait for its slowest neighbor's information to proceed. An asynchronous algorithm based on broadcast gossip was first proposed in [14] dealing with

L. Zhao and M. Valero are with the Department of Information Technology, Kennesaw State University, Marietta, GA, 30060 USA. e-mail: (lzhao10@kennesaw.edu, mvalero2@kennesaw.edu.)

F. Li is with the Department of Electrical and Computer Engineering, Kennesaw State University, Marietta, GA 30067, USA. (e-mail: fli6@kennesaw.edu.)

the average consensus problem. Nedic developed new asynchronous algorithms in [15] that adopted the similar random broadcast scheme but considered solving convex optimization with "real" objective functions . The computation component of the algorithm in [15] relies on a simple but efficient (sub)gradient-based update in each node locally. Note that our proposed algorithm is focusing on improving the local update rule for each node in order to reduce the communication rounds required with neighbors. This is based on a natural fact in distributed/decentralized computing that more optimized local node update can allow the nodes to have less information exchange with each other towards convergence. To be specific, in addition to using gradient descent for local update, our design combines neighbors' gradient information and momentum [16] to accelerate the entire process for all the nodes to reach the optimal solution.

The main contribution of this paper is three-fold: 1) A novel decentralized data analytics algorithm is developed. 2) A hybrid decentralized protocol is designed for adapting the proposed algorithm into the edge computing infrastructure for IoT networks. 3) Convergence properties of the proposed algorithm have been analyzed ensuring that all the nodes can reach the same optimal solution eventually.

III. Problem Statement

We consider an undirected connected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ denotes the node set and $\mathcal{E}$ is the edge set. The number of nodes executing decentralized consensus is p and two nodes i and j are neighbors if $(i, j) \in \mathcal{E}$. There are m objective functions that defined by the data acquisition processes. That is, there are m data-holding places owning the data generated locally across the network. Note that when $p = m$, each node i is able to access a local private objective function $F_i : \mathbb{R}^n \to \mathbb{R}$. The goal is that each node can obtain the global optimal solution $x \in X$ minimizing the summation of m private convex objective functions. The resulting problem is described as follows.

$$\min_{x \in X} \left\{ F(x) := \sum_{i=1}^{m} F_i(x) \right\}. \tag{1}$$

The formulation in (1) is powerful in modeling various problems in signal processing [17], control [18], and statistical learning [19]. Examples of function F_i include least-squares, logistic regression, support vector machines, etc. Related applications have been investigated in literature such as electrical power systems, sensor networks, smart buildings, and smart manufacturing [20], [21]. In this work, we propose a novel decentralized algorithm and a hybrid protocol to solve (1) in an edge computing empowered IoT network [22] (see Fig. 1). We believe that this is an important addition to existing architectures for data analytics in edge computing based IoT networks.

IV. Algorithm

A. Decentralized algorithm design

We first propose an algorithm aiming to solve the problem in (1) in a fully decentralized manner. The number of nodes

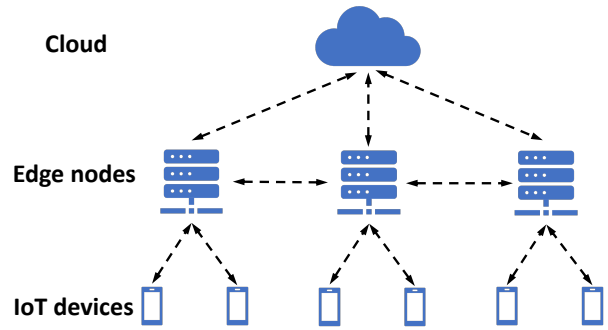


Fig. 1: Edge computing empowered IoT infrastructure [23].

p is set to m such that each node i can access a data-holding place, where the information is embedded in local private objective function F_i. Each node performs local computation and communication with its neighbors to obtain the global optimal solution. The main computation step in this proposed decentralized algorithm is as follows:

$$\begin{aligned} y_k^i &= \theta x_{k-1}^{i_k} + (1-\theta)\, x_{k-1}^i, \\ x_k^i &= P_X \Bigg[y_k^i - \alpha_{i,k} \tilde{\nabla} F_i(y_k^i) - \rho_{i,k} \Big(\sum_{u \in \mathcal{N}_i} \tilde{\nabla} F_u(x_{\tau_{u,k}}^u) \Big) \\ &\quad + \beta_{i,k} \left(x_{k-1}^i - x_{k-2}^i \right) \Bigg], \end{aligned} \tag{2}$$

where k is the virtual global iteration number. $\mathcal{N}_i$ is the set of neighbors of node i. x_k^i represents node i's solution at k-th iteration. $\tilde{\nabla} F_u(x_{\tau_{u,k}}^u)$ means the (sub)gradient of node u at point $x_{\tau_{u,k}}^u$. $\tau_{u,k}$ characterizes the possible outdated gradient information. If $\tau_{u,k} = k$, $\tilde{\nabla} F_u(x_{\tau_{u,k}}^u)$ will be the current gradient of node u. P_X is the projection operator onto the feasible set X. $\theta, \alpha_{i,k}, \rho_{i,k}$ and $\beta_{i,k}$ are parameters for the algorithm and would be discussed in the context later. The algorithm can be summarized as follows.

Algorithm 1 Accelerated decentralized algorithm based on (2) with $p = m$.

Input: Starting point $x_0^1, x_0^2, \cdots, x_0^p$.

while each node i, $i \in \{1, 2, \cdots, p\}$ asynchronously **do**
  if (node i_k's local clock ticks now) **then**
    Node i_k broadcasts its estimate $x_{k-1}^{i_k}$ and (sub)gradient $\tilde{\nabla} F_i(x_{k-1}^{i_k})$ to its neighbors;
    Node i who receives node i_k's broadcast updates its solution x_k^i based on (2).
  end if
end while

Note that the first equation in (2) fuses node i's solution with neighbor node i_k's using weighted average. This step is mainly adopted to push all the nodes to reach consensus on their estimates for the global solution. The second equation in (2) is the local optimization step for node i, in which $y_k^i - \alpha_{i,k} \tilde{\nabla} F_i(y_k^i)$ is a regular gradient descent step using F_i owned by node i [24]. Notice that the scheme proposed in

[15] uses $y_k^i - \alpha_{i,k}\tilde{\nabla}F_i(y_k^i)$ to update x_k^i. In our designed local update rule in (2), we have two extra terms. The term $\sum_{u\in\mathcal{N}_i}\tilde{\nabla}F_u(x_{\tau_{u,k}}^u)$ contains the neighbors' gradient information for node i. Together with the terms $y_k^i - \alpha_{i,k}\tilde{\nabla}F_i(y_k^i)$, it is approximately equivalent to run gradient descent using multiple nodes' data (multiple $F_i s$). In the ideal case that node i is a neighbor with all the other nodes in the system, node i will be directly optimizing the global function F in (1) and thus speeding up the process of obtaining the global optimal solution. The last term in the second equation of (2) $\left(x_{k-1}^i - x_{k-2}^i\right)$ is called "momentum". The momentum term brings history information into the current estimate for finding better gradient directions with the hope of gaining faster convergence [16]. Notice that parameters $\alpha_{i,k}, \rho_{i,k}, \beta_{i,k}$ are the step sizes of the aforementioned terms, respectively.

Remark 1. *There is a fundamental trade-off between computation and communication in distributed/decentralized computing frameworks [25]. It implies that it is possible to trade extra computing power to reduce the communication load. In this study, local updating rules (e.g.* (2)*) requiring only computation of function values and gradients (not Hessians since they are much more expensive to compute) are considered. We aim to let local nodes work harder in order to reduce the communication rounds needed towards convergence.*

B. Convergence results

We demonstrates the convergence results of our proposed decentralized algorithm in this subsection. Assumptions required for convergence analysis are described in Assumption 1-3.

Assumption 1. *Bounded (sub)gradient for each function F_i such that $\|\tilde{\nabla}F_i\| \leq G$ where $G > 0$ is some constant.*

Assumption 2. *The constraint set X is bounded such that the problem has finite number of solutions.*

Assumption 3. $\sum_{k=1}^{\infty}\frac{\rho_{i,k}}{k\alpha_{i,k}} < \infty,\ \sum_{k=1}^{\infty}\frac{\beta_{i,k}}{k\alpha_{i,k}} < \infty$ *almost surely.*

Theorem IV.1. *The sequence $\left\{x_k^i\right\}, \forall i \in \mathcal{V}, k \geq 0$ generated by Algorithm 1 for each node i has the following consensual property almost surely:*

$$\sum_{k=1}^{\infty}\frac{1}{k}\|x_{k-1}^i - \bar{x}_{k-1}\| < \infty, \text{ and } \lim_{k\to\infty}\|x_k^i - \bar{x}_k\| = 0,$$

where $\bar{x}_{k-1} = \frac{1}{m}\sum_{i=1}^{m} x_{k-1}^i$.

Proof. Please see Appendix B for details. □

Theorem IV.2. *The sequence $\left\{x_k^i\right\}, \forall i \in \mathcal{V}, k \geq 0$ generated by Algorithm 1 for each node i reaches to a same optimal point almost surely.*

Proof. Please see Appendix C for details. □

Theorem IV.3. *For Algorithm 1, if F_i is strongly convex then we have the following:*

$$\frac{1}{T}\sum_{k=1}^{T}\left(\mathbb{E}\left[\sum_{i=1}^{m}F_i(x_k^i)\right]\right) - F\left(x^*\right) \leq O\left(\frac{\log T}{T}\right),$$
$$\frac{1}{T}\sum_{k=1}^{T}\left(\mathbb{E}\left[\sum_{i=1}^{m}\left\|x_k^i - x^*\right\|\right]\right) \leq O\left(\frac{\log T}{T}\right).$$

Proof. Please see Appendix D for details. □

Remark 2. *Note that Assumption 1-2 are common conditions for analyzing distributed/decentralized algorithms [11]–[13]. For the proposed decentralized algorithm 1, the main goal is that all the nodes can obtain the optimal solution. Also, we are interested to see how fast all the nodes reach the optimal solution. Theorem IV.1 indicates that each node will converge to the average solution of all the nodes eventually, and thus it implies that all the nodes will reach consensus on their estimates. Based on Theorem IV.1, Theorem IV.2 confirms that our goal can be achieved such that all the nodes will reach the same optimal solution. The statement is derived by first showing certain node can reach the optimal solution and then applying the fact that all the nodes will be consensual on their estimates (Theorem IV.1). In general, Theorem IV.3 characterizes the dependence of the solution error (captured by the distance between nodes' solutions to the optimal solution) on the number of iterations. It implies how fast all the nodes obtain the optimal solution in terms of number of iterations.*

V. Implementation: Hybrid Protocol

In this section, we develop a hybrid decentralized protocol in order to implement and fit the proposed Algorithm 1 into the edge computing architecture (illustrated in Fig. 1). Assume that there are q edge nodes in the network running decentralized analytics. Each edge node is responsible for a set of IoT nodes near to it. Assuming each IoT is a raw data holder and IoT node j can then access the private objective function $F_j, \forall j \in \{1, 2, \cdots, m\}$. The IoT nodes will calculate their gradients and send them to their corresponding edge nodes. The edge nodes will then perform the fully decentralized algorithm with each other to obtain the global optimal solution. Since the aggregation process exists for the edge nodes to collect their IoTs' gradients, the whole protocol is considered to be executed in a hybrid decentralized fashion. The hybrid protocol consists of two parts: the procedures for edge nodes and IoT nodes, respectively. They are summarized in Algorithm 2 and 3 as follows.

Algorithm 2 Edge node procedure

Input: Starting point $x_0^1, x_0^2, \cdots, x_0^q$. Initialize the iteration number $k = 0$.

while the maximum number of iterations has not been reached and the change between the two most recent estimates are not within the pre-determined threshold, each edge node i asynchronously **do**
  if (node i_k's local clock ticks now) **then**
    Edge node i_k broadcasts its estimate $x_{k-1}^{i_k}$ and (sub)gradient $g_k^{i_k}$ to its neighbors;
    Edge node i who receives edge node i_k's broadcast updates its solution as follows.
    Edge node i mixes its current estimate with $x_{k-1}^{i_k}$ using the first equation in (2).
    Edge node i sends mixed estimate y_k^i to its corresponding IoT nodes.
    Edge node i waits for the IoT nodes to return their gradients and aggregate them (the summation) as g_k^i.
    Edge node i update its estimate based on the second equation in (2) with replacing local gradient $\tilde{\nabla} F_i(y_k^i)$ with g_k^i (aggregated from its IoT nodes), neighbors' gradient $\tilde{\nabla} F_u(x_{\tau_{u,k}}^u)$ with g_k^u, respectively.
  end if
  Increment k.
end while
Send EXIT signal.

Algorithm 3 IoT node procedure

while EXIT signal has not been received, each IoT node j with j belongs to the set of IoT nodes that associated with edge node i **do**
  IoT node j receives edge node i's mixed estimate y_k^i.
  IoT node j computes the gradient with respect to y_k^i using its local objective function F_j.
  IoT node j sends the computed gradient to its corresponding edge node i.
end while

The decentralized part of the hybrid protocol is that the edge nodes compute locally and exchange their estimates with each other. This decentralized computing framework is same as the counterpart described in Algorithm 1 and is depicted in Fig. 2. Note that an aggregator-based scheme is adopted for the interaction between each edge node and its IoT nodes. The edge node sends its parameter estimate to its IoT nodes and each IoT node compute the gradient with respect to the estimate received using its local objective function. These calculated gradients will then be returned to the edge node for aggregation. An illustration of this process is shown in Fig. 3. Notice that in our hybrid decentralized protocol, the raw data is kept in all the IoT nodes (where the data is generated) and has never been shared. Only the gradients and parameter estimates would be exchanged in the data analytics process.

The signaling overhead and computation complexity in Algorithm 2 and 3 are analyzed as follows. Assume there are m IoT nodes (raw data-holders) and p edge nodes in the system. In each iteration, there are $O(1)$ broadcast and $O(d)$ messages received by the edge nodes where d is the maximum degree of the network formed by the edge nodes (see Fig. 2). To be specific, if the edge nodes form a mesh network, then $d = p - 1$. The size of each communication is twice the size of the decision vector (i.e. x). For the interaction between edge nodes and their IoT nodes (see Fig. 3), in each iteration, there are at most $O(m)$ communication from IoT nodes to their corresponding edge nodes and $O(m)$ communication vice versa. The size of each communication is same as the size of the decision vector. Regarding computation complexity, there are $O(d)$ edge node updates performed in parallel in each iteration where d is the maximum degree aforementioned. Each edge node update involves gradient evaluations conducted by its IoT nodes. Hence, there are at most $O(m)$ gradient calculations in each iteration.

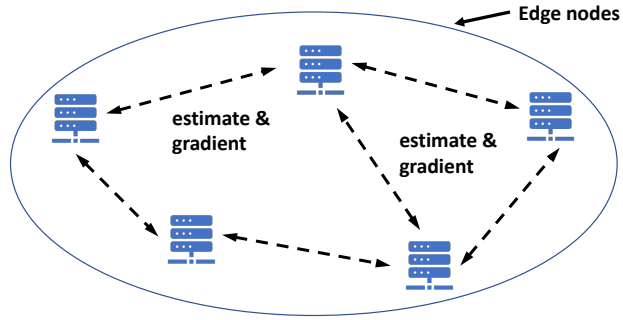


Fig. 2: Interaction among the edge nodes. The edge nodes exchange their estimates and gradient information in a decentralized fashion.

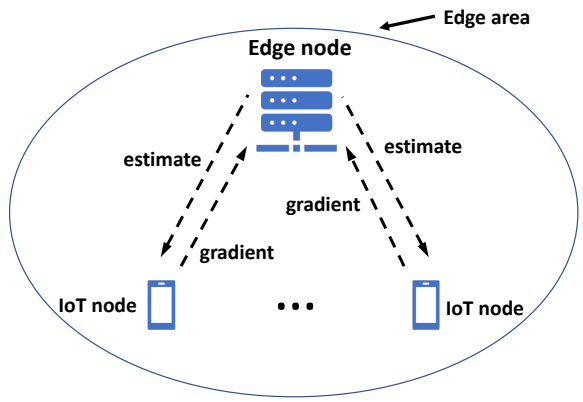


Fig. 3: Interaction between edge node and its IoT nodes in an edge area due to proximity. The edge node sends its mixed estimate to the IoT nodes. IoT nodes calculate the gradient using their local objective functions and then return the gradients to the edge node for its update.

Remark 3. *In certain time-sensitive IoT applications, a solution needs to be generated as quickly as possible and we might not wait until the edge nodes to converge. But in early stage, the edge nodes' estimates might be different from each other and it is difficult to determine which one is better. Thus*

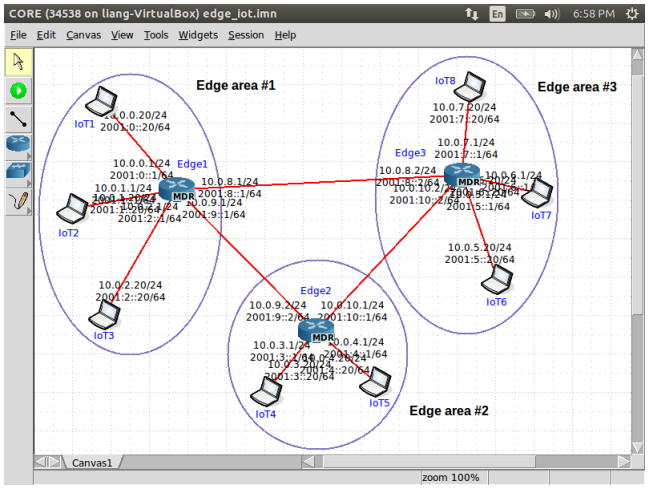


Fig. 4: An example of CORE GUI. In each edge area, there is one edge node and some IoT nodes.

a common strategy is to average all the estimates available.

VI. Experiment Evaluation

A. Experiment Setup

We conduct experiments for our proposed hybrid decentralized protocol on two applications: regularized least-square and computed tomography. The simulations are tested in Common Open Research Emulator (CORE), which is a distributed network emulator [26]. All the experiments are simulated on a MacBook Pro with an 2.6 GHz 6-Core Intel Core i7 CPU and 16 GB memory. An example of the CORE GUI is shown in Fig. 4. Regarding the parameters in the protocol (see (1)): a). Any $\theta \in [0, 1]$ is valid and we choose $\theta = 0.5$ as the "mixing" parameter for all the tests in this paper. b). The step-size $\alpha_{i,k}$ for the local gradient is set to the reciprocal of the number of updates edge node i has performed till iteration k. c). To satisfy Assumption 3, both $\rho_{i,k}$ and $\beta_{i,k}$ are set to $\alpha_{i,k}^2$ for simplicity. In addition, we adapt two decentralized algorithms in [15] (named Nedic) and [27] (named FDDA) into our hybrid architecture and compare our proposed protocol with them. Three main metrics are used to test and compare the convergence characteristics of the decentralized methods aforementioned: objective value, relative error, and disagreement. They are defined as follows.

1) objective value: $F(\bar{x}_k)$, where $\bar{x}_k = \frac{1}{q}\sum_{i=1}^{q} x_k^i$ is the average value of all the edge nodes at iteration k.
2) relative error: $\frac{\|\bar{x}_k - x^*\|_2}{\|x^*\|_2}$, where x^* is the optimal solution pre-computed by a centralized solver. This quantity tracks the distance between the obtained average solution to the optimal one.
3) disagreement: $\frac{1}{q}\sum_{i=1}^{q} \|x_k^i - \bar{x}_k\|_2$. This quantity measures the disagreement among the edge nodes on their estimates.

B. Test on regularized least-square

we first test our proposed hybrid protocol on the application of regularized least-square [28], which is a ubiquitous problem in statistics, computer science, economics [29]. The formulation of the regularized (Tikhonov regularization [30]) least-square problem can be expressed as follows.

$$\min_x \ \frac{1}{2}\|Ax - b\|_2^2 + \gamma\|x\|_2^2, \tag{3}$$

where A is a matrix whose rows usually represent data. b is a vector and γ is the regularization parameter used to control the trade-off between the data fitting term (first one) and the regularization part. To fit it into our hybrid decentralized framework, (3) is decomposed as follows.

$$\min_x \ \sum_{i=1}^{m} \frac{1}{2}\|A_i x - b_i\|_2^2 + \frac{\gamma}{m}\|x\|_2^2. \tag{4}$$

Hence, the local objective function for IoT node i is:

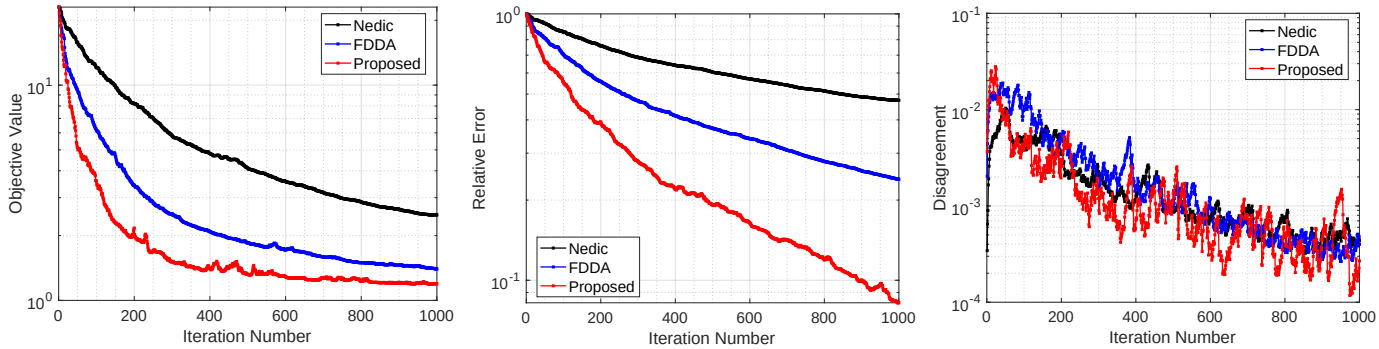
$$F_i(x) = \frac{1}{2}\|A_i x - b_i\|_2^2 + \frac{\gamma}{m}\|x\|_2^2 \tag{5}$$

For the set up of our network, we perform our test on 50 IoT nodes (data-holders, $m = 50$) and 10 edge nodes. In each edge area, there are an edge node and 5 IoT nodes associated with it. For the connectivity among all the edge nodes, random topologies with average degree 3 and 5 are tested, respectively. Matrix A is randomly generated with size 50×80. A and b are evenly decomposed for all the nodes thus each IoT node has one row of the data in A and b. Each element in matrix A and vector b is uniformly sampled over $[0, 1)$. The regularization parameter γ is set to 1.

The experiment results are illustrated in Fig. 5-7. In Fig. 5 and 6, it can be seen that our proposed protocol outperforms the benchmarks in terms of the convergence speed for average objective value, relative error (accuracy) and the disagreement among all the edge nodes. Comparing Fig. 5 and 6, we can see that when the connectivity among the edge nodes is higher (change average degree from 3 to 5), all the methods can reach a same accuracy faster. This demonstrate the effect of connectivity in decentralized algorithms. Also, notice that in the metric of disagreement of estimate among all the edge nodes (IoT nodes in the same edge area have the same estimate as its edge node), higher connectivity smooth out the volatility. This is expected because higher connectivity can make information propagation faster in a decentralized environment. Fig. 7 demonstrates the messages exchanged for each edge node with other neighbors. Notice that in both cases (degree is 3 and 5), the communication among all the nodes are balanced, this is one of the characteristics of decentralized algorithms. In addition, it can be observed that each edge node exchanges more messages in the higher connectivity setting as they can receive messages from more neighbors.

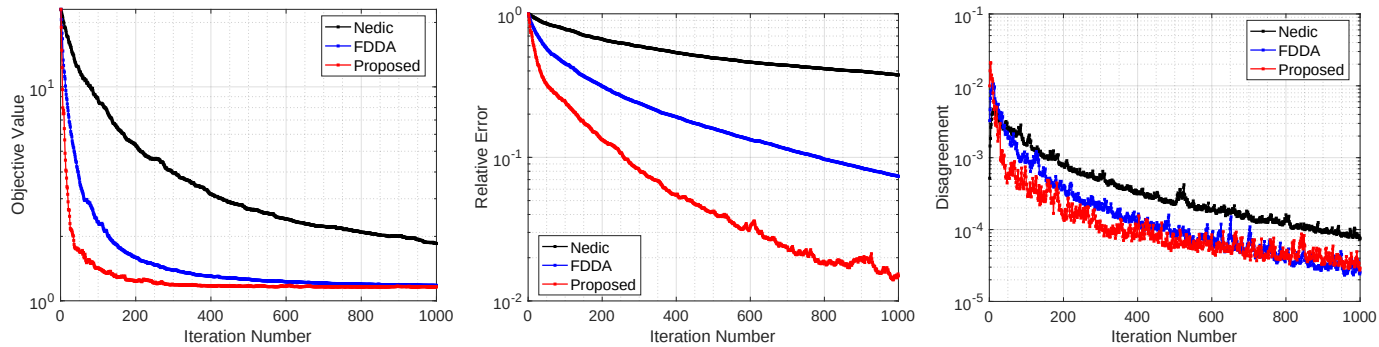
C. Test on computed tomography

We conduct experiments of our proposed protocol on the application of computed tomography [31] in this subsection. We use the code in the AIR package [32] for this test. A 2D tomography test problem is generated using parallel beams. For simplicity, we use the same model in (3) to reconstruct the tomography. In the test problem, the dimension of the tomography is 50×50 (The resolution of the image), the size



(a) (b) (c)

Fig. 5: Comparison of convergence behavior on the regularized least-square problem. 10 edge nodes are randomly connected with average degree = 3. In each edge area, there are 5 IoT nodes.



(a) (b) (c)

Fig. 6: Comparison of convergence behavior on the regularized least-square problem. 10 edge nodes are randomly connected with average degree = 5. In each edge area, there are 5 IoT nodes.

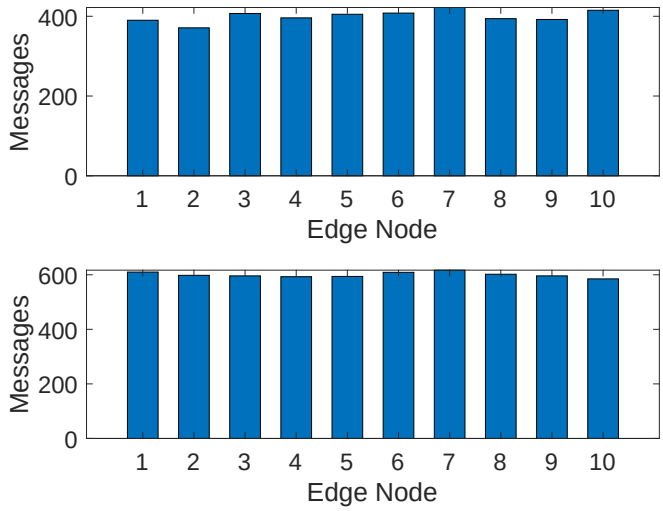
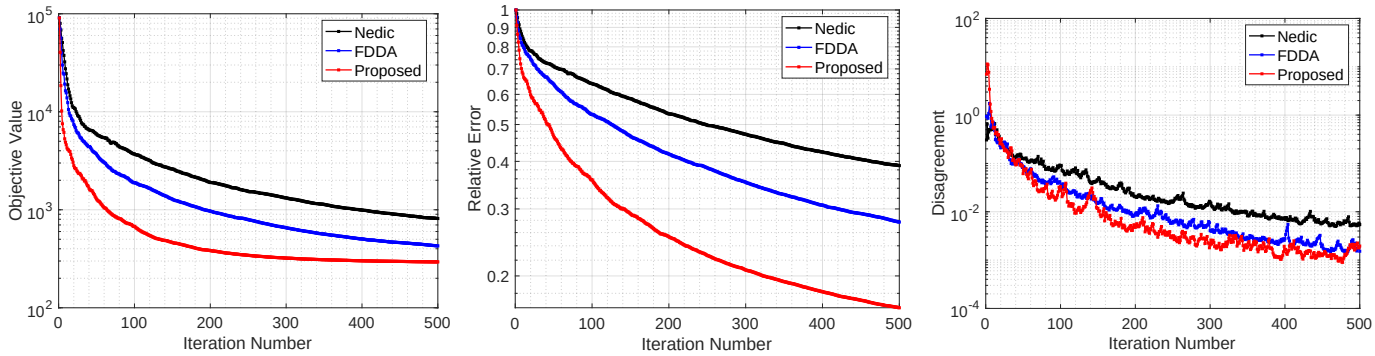


Fig. 7: Communication cost on the regularized least-square problem with average node degree equals 3 (upper) and 5 (bottom) for the edge nodes topology, respectively.

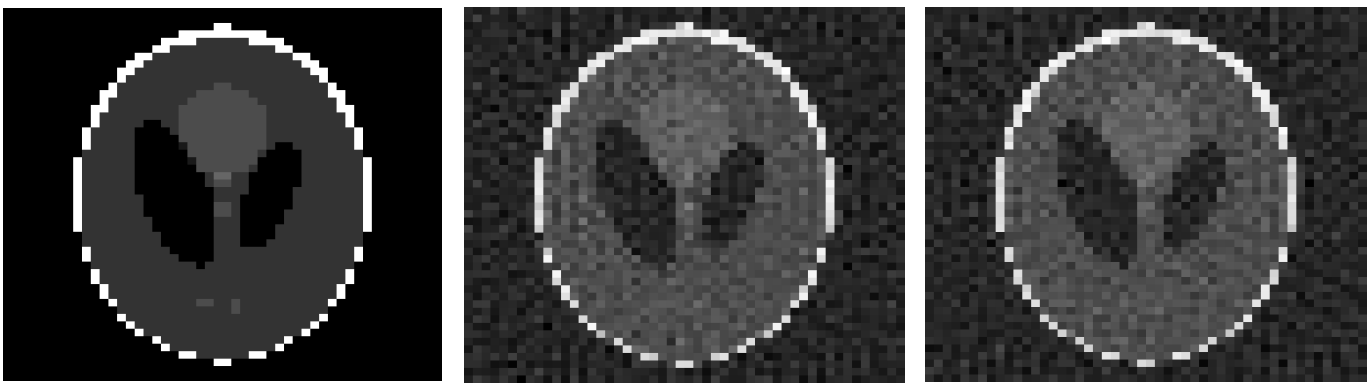
of matrix A is 5400×2500 and the original vector b has been added a 5% random noise to create the "noisy" problem. The number of edge areas is this test is 10. The neighbors of each edge node is randomly generated and the average degree of each node is 5. Each edge area has 10 IoT nodes and every IoT node contains 54 rows of data in matrix A.

The experiment results are illustrated in Fig. 8-10. Fig. 8 demonstrates again that our proposed hybrid protocol is superior to the other benchmarks in the speed of obtaining the global optimal solution for all the nodes in the network. Fig. 9 shows the tomography results of the 2D test problem and it can be observed that our decentralized solution is close to the centralized counterpart visually. Notice that the centralized solution (fig (b)) is not identical to the groundtruth (fig (a)) due to the added noise. The model used to reconstruct the tomography affect the tomography results we obtained comparing to the groundtruth and how to design reconstruction model is out of scope of this presenting work. Our goal is to obtain the same optimal solution (as the centralized one) using the proposed hybrid decentralized protocol. In Fig. 10, we take a closer look on the convergence performance of each edge node individually. It can be seen that after around 60 iterations, all the edge nodes' estimates are very close to each other and it is a desirable property that we can pick up any edge node's



(a) (b) (c)

Fig. 8: Comparison of convergence behavior on the computed tomography problem. 10 edge nodes are randomly connected with average degree = 5. In each edge area, there are 10 IoT nodes.



(a) ground-truth. (b) centralized solution. (c) proposed method at iteration 500.

Fig. 9: Tomography results for the 2D parallel-beam tomography test problem.

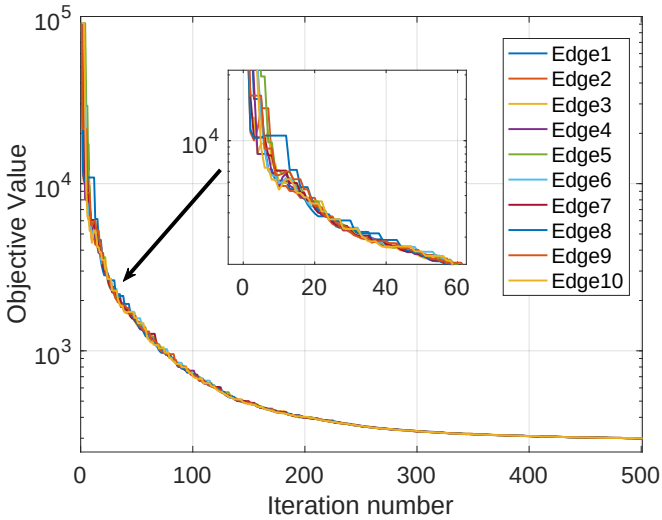


Fig. 10: Convergence behavior of each edge node in terms of the objective value on the computed tomography problem.

estimate as our final solution since they are all consensual. Behind Fig. 10, note that there are 50 gradient evaluations performed by IoT nodes in parallel in each iteration. This is due to the fact that there are 5 edge nodes selected for updating in each iteration and each edge area contains 10 IoT nodes.

D. Test on logistic regression

In this subsection, we perform tests on logistic regression task to further demonstrate the applicability of our decentralized protocol. We use the enron email dataset [33] and train a logistic regression model to classify an email as spam or ham. The task can be formulated in a decentralized optimization fashion as follows.

$$\min_{x} \sum_{i=1}^{m} \sum_{j=1}^{m_i} \log\left(1+\exp\left(-b_i^j \left(A_i^j\right)^T x\right)\right). \quad (6)$$

The local objective function held by IoT node i is:

$$F_i(x) = \sum_{j=1}^{m_i} \log\left(1+\exp\left(-b_i^j \left(A_i^j\right)^T x\right)\right), \quad (7)$$

where m_i is the number of the instances owned by IoT node i. b_i^j is the j-th element of vector b_i, which contains the binary

outcomes at i-th IoT. Similarly, $\left(A_i^j\right)^T$ denotes the j-th row of matrix A_i. Each row in A_i represents an instance and the columns are feature variables. We use the first two folders of the emails in this scenario. The first 5000 emails are used for training and the remaining 6029 emails for testing. The emails are pre-proceesed and 7997 frequent words are kept as features. For the setup about the edge computing environment, 100 IoT nodes ($m = 100$) and 10 edge nodes are used. In each edge area, there are 10 IoT nodes. The number of immediate neighbors for each edge node is set to 5. The raw data is evenly divided into the IoT nodes and thus each one contains 50 records ($m_i = 50$). Hence, the dimension of A_i and b_i are 50×7997 and 50×1, respectively. The experiment results are shown in Fig. 11. The log loss and classification error are adopted to measure the performance of the model. The average model of the 10 edge nodes is used to capture the characteristics of the proposed decentralized analytics process on the training and testing datasets, respectively. It can be seen that both training and testing error are below 0.02 (accuracy 0.98) after around 20 iterations.

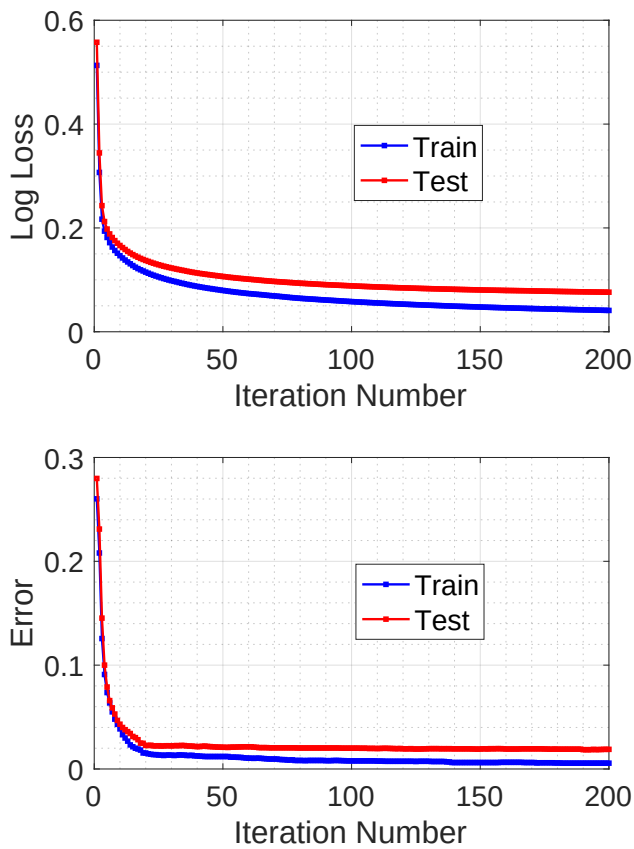


Fig. 11: Logistic regression for the Enron email dataset classification problem. There are 10 edge nodes and 100 IoT nodes. The average model of all the 10 edge nodes is tested for the two performance metrics: log loss (value of the loss function) and error (fraction that the model is is wrong in classification).

VII. Conclusion

With large-scale deployment of IoT devices, traditional Cloud-based solution is prohibited for associated IoT data analytics tasks due to three main limitations: large data movement, high latency, and limited privacy. Edge computing emerges as a promising infrastructure for IoT networks while careful design is needed for building the data analytics pipeline. We proposed a hybrid decentralized framework for edge computing empowered IoT networks aiming to address all the issues above. Theoretical analysis and experimental evaluations elucidate that the proposed approach is capable of offering efficient data analytics with all the raw data localized. Possible future work includes investigating new analysis for our decentralized scheme that can lift conditions such as Assumption 1 and embedding other privacy-preserving techniques to further enhance the privacy level of the framework.

Appendix A
Lemmas needed for the proof

In this section, we first list the lemmas required for the proof of Theorem IV.1-IV.3.

Lemma A.1. *(The supermartingale convergence theorem [Proposition 8.2.10 in [34]]) Assume σ_k, φ_k, ω_k, and ε_k are nonnegative random variables and assume the following hold*

$$\mathbb{E}\left(\sigma_{k+1}|\Omega_k\right) \leq \sigma_k - \varphi_k + \varepsilon_k \quad almost\ surely,$$
$$\sum_{k=1}^{\infty} \varepsilon_k < \infty \quad almost\ surely$$

where $\mathbb{E}\left(\sigma_{k+1}|\Omega_k\right)$ represents the conditional expectation given all the past history of σ_k, φ_k, and ε_k up to iteration k. Then it concludes that

$$\sigma_k \to \sigma \quad almost\ surely, \quad \sum_{k=1}^{\infty} \varphi_k < \infty \quad almost\ surely$$

where $\sigma \geq 0$ is some random variable.

Lemma A.2. *([15]) The upperbounds of step size $\alpha_{i,k}$ are obtained as follows when k is large enough ($k > \tilde{k}(m,q)$)*

$$\alpha_{i,k} \leq \frac{2}{k\delta_i}, \ \alpha_{i,k}^2 \leq \frac{4m^2}{k^2 p_*^2}, \ \left|\alpha_{i,k} - \frac{1}{k\delta_i}\right| \leq \frac{2}{k^{\frac{3}{2}-q} p_*^2},$$

where δ_i is the total probability that node i updates. p_* denotes the minimum among all p_{ij}'s. $q \in \left(0, \frac{1}{2}\right)$ is some constant. $\tilde{k}(m,q)$ is an integer determined by the number of nodes m and q.

Lemma A.3. *(Proposition 1.1.9 in [35]) The projection used in (2) is nonexpansive such that*

$$\left\|P_X\left(x\right) - P_X\left(y\right)\right\| \leq \left\|x - y\right\|,$$

where P_X is the projection operator on set X and x, y are two points.

Lemma A.4. *(lemma 6 in [27]) The following two relations hold*

$$\sum_{i=1}^{m} \mathbb{E}\left[\left\|y_k^i - x\right\| |\Omega_{k-1}\right] \leq \sum_{i=1}^{m} \left\|x_{k-1}^i - x\right\|,$$
$$\sum_{i=1}^{m} \mathbb{E}\left[\left\|y_k^i - x\right\|^2 |\Omega_{k-1}\right] \leq \sum_{i=1}^{m} \left\|x_{k-1}^i - x\right\|^2.$$

APPENDIX B
PROOF OF THEOREM IV.1

Proof. We prove Theorem IV.1 for Algorithm 1. The proof follows the framework for Theorem VI.7 in [27]. The main difference is the error term e_k^i and also its bounds. We provide a sketch for the sake of completeness. Let s_k^t to be avector with components $\left[x_k^i\right]_t, \forall i \in \mathcal{V}$, where $\left[x_k^i\right]_t$ is the t-th element of node i's estimate at iteration k. It follows that:

$$s_k^t = W_k s_{k-1}^t + d_k^t, \tag{8}$$

where d_k^t is a vector defined as follows.

$$\left[d_k^t\right]_i = \left[-\alpha_{i,k}\left(\tilde{\nabla} F_i(y_k^i) + e_k^i\right)\right]_t, i \in J_k \tag{9}$$

And $\left[d_k^t\right]_i = 0$ otherwise. J_k is the set of nodes who receive the estimate and will perform update locally in iteration k. We can express the error term e_k^i as follows.

$$e_k^i = \frac{\rho_{i,k}}{\alpha_{i,k}}\left(\sum_{u\in\mathcal{N}_i}\tilde{\nabla} F_u(x_{\tau_{u,k}}^u)\right) - \frac{\beta_{i,k}}{\alpha_{i,k}}\left(x_{k-1}^i - x_{k-2}^i\right).$$

Following the proof in Appendix A in [27], we can obtain the following.

$$\begin{aligned}
&\mathbb{E}\left[\left\|s_k^t - \left[\bar{x}_k\right]_t \mathbf{1}\right\|^2 |\Omega_{k-1}\right] \\
&\leq \mu\left\|s_{k-1}^t - \left[\bar{x}_{k-1}\right]_t \mathbf{1}\right\|^2 + \frac{4m^3}{k^2 p_*^2}\left\|\tilde{\nabla} F_i(y_k^i) + e_k^i\right\|^2 \\
&+ \sqrt{\mu}\left\|s_{k-1}^t - \left[\bar{x}_{k-1}\right]_t \mathbf{1}\right\| \frac{2m\sqrt{m}}{kp_*}\left\|\tilde{\nabla} F_i(y_k^i) + e_k^i\right\|
\end{aligned} \tag{10}$$

where Ω_k is the σ-algebra containing the past history up to iteration k, i.e.

$$\Omega_k = \left\{x_0^i, i_t, j_t, \forall i \in \mathcal{V}, t = 0, 1, \cdots k\right\}.$$

. The different thing about our proposed Algorithm 1 is that the following terms in (10) can be bounded as follows.

$$\begin{aligned}
&\left\|\tilde{\nabla} F_i(y_k^i) + e_k^i\right\|^2 \leq 2\left\|\tilde{\nabla} F_i(y_k^i)\right\|^2 + 2\left\|e_k^i\right\|^2 \\
&\leq 2G^2 + 4\left(\frac{\rho_{i,k}\left|\mathcal{N}_i\right| G}{\alpha_{i,k}}\right)^2 + 4\left(\frac{\beta_{i,k}}{\alpha_{i,k}}\left\|x_{k-1}^i - x_{k-2}^i\right\|\right)^2 \\
&\leq 2G^2 + 4\left(\frac{\rho_{i,k}\left|\mathcal{N}_i\right| G}{\alpha_{i,k}}\right)^2 + 4\left(\frac{2\beta_{i,k} d_X}{\alpha_{i,k}}\right)^2,
\end{aligned}$$

$$\begin{aligned}
\left\|\tilde{\nabla} F_i(y_k^i) + e_k^i\right\| \leq& G + \alpha_{i,k}\left(NG + 2d_X\right) \\
\leq& \frac{2mG}{kp_*} + \frac{4m^2\left(NG + 2d_X\right)}{k^2 p_*^2},
\end{aligned}$$

where we use the relation that

$$\begin{aligned}
\left\|x_{k-1}^i - x_{k-2}^i\right\| &= \left\|\left(x_{k-1}^i - x^*\right) + \left(x^* - x_{k-2}^i\right)\right\| \\
&\leq \left\|x_{k-1}^i - x^*\right\| + \left\|x_{k-2}^i - x^*\right\| \\
&\leq 2d_X,
\end{aligned}$$

d_X is the diameter of the bounded set X such that $d_X \equiv \max_{a,b\in X}\|a - b\|$, $\left|\mathcal{N}_i\right|$ is the cardinality of set $\mathcal{N}_i$.

The remaining deduction is similar to the proof of Theorem VI.7 in [27]. □

APPENDIX C
PROOF OF THEOREM IV.2

Proof. Since Algorithm 1 is a stochastic algorithms and associated proofs are usually based on the supermartingale convergence theorem in Lemma A.1 is show the almost sure convergence. The attack plan is to bound the terms and fit them into Lemma A.1. We prove Theorem IV.2 for Algorithm 1. The proof follows the framework for Theorem VI.8 in [27]. Again, the error term e_k^i is defined differently and also we need to find the bounds for that. We provide a sketch for the sake of completeness.

Taking a look at node i with $i \in J_k$. We first subtract some x in the feasible set on both sides of the second equation in (2) and then take the square norm on both sides. Based on the property of the projection operation in Lemma A.3, we can obtain the following.

$$\begin{aligned}
\left\|x_k^i - x\right\|^2 \leq& \left\|y_k^i - x\right\|^2 + \alpha_{i,k}^2\left\|\tilde{\nabla} F_i(y_k^i) + e_k^i\right\|^2 \\
&- 2\alpha_{i,k}\left(\tilde{\nabla} F_i(y_k^i) + e_k^i\right)^T\left(y_k^i - x\right),
\end{aligned} \tag{11}$$

where e_k^i is defined as follows.

$$e_k^i = \frac{\rho_{i,k}}{\alpha_{i,k}}\left(\sum_{u\in\mathcal{N}_i}\tilde{\nabla} F_u(x_{\tau_{u,k}}^u)\right) - \frac{\beta_{i,k}}{\alpha_{i,k}}\left(x_{k-1}^i - x_{k-2}^i\right).$$

Using the facts that $\left|a^T b\right| \leq \|a\|\|b\|$, $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ and the assumption of bounded (sub)gradient, the terms associated with e_k^i can be bounded as follows.

$$\begin{aligned}
&-\left(e_k^i\right)^T\left(y_k^i - x\right) = \frac{\rho_{i,k}}{\alpha_{i,k}}\left(\sum_{u\in\mathcal{N}_i}\tilde{\nabla} F_u(x_{\tau_{u,k}}^u)\right)^T\left(y_k^i - x\right) \\
&- \frac{\beta_{i,k}}{\alpha_{i,k}}\left(x_{k-1}^i - x_{k-2}^i\right)^T\left(y_k^i - x\right) \\
&\leq \frac{\rho_{i,k}\left|\mathcal{N}_i\right| G}{\alpha_{i,k}}\left\|y_k^i - x\right\| + \frac{\beta_{i,k}}{\alpha_{i,k}}\left\|x_{k-1}^i - x_{k-2}^i\right\|\left\|y_k^i - x\right\| \\
&\leq \frac{\rho_{i,k}\left|\mathcal{N}_i\right| G}{\alpha_{i,k}}\left\|y_k^i - x\right\| + \frac{2\beta_{i,k} d_X}{\alpha_{i,k}}\left\|y_k^i - x\right\|,
\end{aligned} \tag{12}$$

$$\begin{aligned}
&\left\|\tilde{\nabla} F_i(y_k^i) + e_k^i\right\|^2 \leq 2\left\|\tilde{\nabla} F_i(y_k^i)\right\|^2 + 2\left\|e_k^i\right\|^2 \\
&\leq 2G^2 + 4\left(\frac{\rho_{i,k}\left|\mathcal{N}_i\right| G}{\alpha_{i,k}}\right)^2 + 4\left(\frac{\beta_{i,k}}{\alpha_{i,k}}\left\|x_{k-1}^i - x_{k-2}^i\right\|\right)^2 \\
&\leq 2G^2 + 4\left(\frac{\rho_{i,k}\left|\mathcal{N}_i\right| G}{\alpha_{i,k}}\right)^2 + 4\left(\frac{2\beta_{i,k} d_X}{\alpha_{i,k}}\right)^2,
\end{aligned}$$

Following the proof in Appendix B in [27], we can obtain the following.

$$\begin{aligned}&\sum_{i=1}^{m}\mathbb{E}\left[\left\|x_k^i-x^*\right\|^2|\Omega_{k-1}\right]\\&\leq(1+b_k)\sum_{i=1}^{m}\left\|x_{k-1}^i-x^*\right\|^2-\frac{2}{k}\left(F(\bar{x}_{k-1})-F(x^*)\right)\\&+\frac{2mG}{kp_*}\sum_{i=1}^{m}\left\|x_{k-1}^i-\bar{x}_{k-1}\right\|+\sum_{i=1}^{m}\delta_i\left(\alpha_{i,k}^2+b_k\right)C_k^i\\&+\frac{2m^2\left(NGd_X\rho_{i,k}+2d_X^2\beta_{i,k}\right)}{kp_*\alpha_{i,k}},\end{aligned}\tag{13}$$

where $N=\max_s|\mathcal{N}_s|$, $b_k=\frac{2}{k^{3/2-q}p_*^2}$, $C_k^i=2G^2+4\left(\frac{\rho_{i,k}NG}{\alpha_{i,k}}\right)^2+4\left(\frac{2\beta_{i,k}d_X}{\alpha_{i,k}}\right)^2$. Select $\rho_{i,k}$ and $\beta_{i,k}$ satisfying the assumption that $\sum_{k=1}^{\infty}\frac{\rho_{i,k}}{k\alpha_{i,k}}<\infty$, $\sum_{k=1}^{\infty}\frac{\beta_{i,k}}{k\alpha_{i,k}}<\infty$ with probability 1. From the definitions of $\alpha_{i,k},b_k$, it can be seen that

$$\sum_{k=1}^{\infty}b_k<\infty,\ \sum_{k=1}^{\infty}\left(\alpha_{i,k}^2+b_k\right)<\infty.\tag{14}$$

The remaining deduction is similar to the proof of Theorem VI.8 in [27].

□

APPENDIX D
PROOF OF THEOREM IV.3

Proof. We prove the convergence rate Theorem IV.3 here. From (11), it follows that for $i\in J_k$,

$$\begin{aligned}\left\|x_k^i-x^*\right\|^2\leq&\left\|y_k^i-x^*\right\|^2+\alpha_{i,k}^2\left\|\tilde{\nabla}F_i(y_k^i)+e_k^i\right\|^2\\&-2\alpha_{i,k}\left(\tilde{\nabla}F_i(y_k^i)+e_k^i\right)^T\left(y_k^i-x^*\right),\end{aligned}\tag{15}$$

where $x^*\in X^*$ is an optimal point for (1). Assume function F_i is λ_i-strongly convex, then the following inequality holds:

$$\left(\tilde{\nabla}F_i(y_k^i)\right)^T\left(y_k^i-x^*\right)\geq F_i(y_k^i)-F_i(x^*)+\frac{\lambda_i}{2}\left\|y_k^i-x^*\right\|^2.\tag{16}$$

Plugging the inequalities (12) and (16) with the property about the diameter d_X into (15) yields

$$\begin{aligned}&\left\|x_k^i-x^*\right\|^2\leq\left\|y_k^i-x^*\right\|^2\\&+2\alpha_{i,k}^2\left[G^2+2\left(\frac{\rho_{i,k}|\mathcal{N}_i|G}{\alpha_{i,k}}\right)^2+2\left(\frac{2\beta_{i,k}d_X}{\alpha_{i,k}}\right)^2\right]\\&-2\alpha_{i,k}\left(F_i(y_k^i)-F_i(x^*)+\frac{\lambda_i}{2}\left\|y_k^i-x^*\right\|^2\right)\\&+2\rho_{i,k}|\mathcal{N}_i|Gd_X+4\beta_{i,k}d_X^2.\end{aligned}\tag{17}$$

At this point, assume we set $\rho_{i,k}=\beta_{i,k}=\alpha_{i,k}^2$. Together with the fact that $\alpha_{i,k}\geq\frac{1}{k}$ and $\alpha_{i,k}^2\leq\frac{4m^2}{k^2p_*^2}$ (Lemma A.2), we can obtain as follows.

$$\begin{aligned}&\left\|x_k^i-x^*\right\|^2\leq\left\|y_k^i-x^*\right\|^2\\&+\frac{8m^2}{k^2p_*^2}\left[G^2+2\left(\frac{\rho_{i,k}|\mathcal{N}_i|G}{\alpha_{i,k}}\right)^2+2\left(\frac{2\beta_{i,k}d_X}{\alpha_{i,k}}\right)^2\right]\\&-\frac{2}{k}\left(F_i(y_k^i)-F_i(x^*)+\frac{\lambda_i}{2}\left\|y_k^i-x^*\right\|^2\right)\\&+\frac{8m^2}{k^2p_*^2}\left[|\mathcal{N}_i|Gd_X+2d_X^2\right]\\&\leq\left(1-\frac{\lambda}{k}\right)\left\|y_k^i-x^*\right\|^2-\frac{2}{k}\left(F_i(y_k^i)-F_i(x^*)\right)\\&+\frac{8m^2}{k^2p_*^2}\left(G^2+NGd_X+2d_X^2+\frac{4m^2\left(2N^2G^2+8d_X^2\right)}{k^2p_*^2}\right).\end{aligned}\tag{18}$$

where $\lambda=\min_s\lambda_s$. Taking conditional expectation on both sides of (18) and considering the fact that $x_k^i=y_k^i$ if $i\notin J_k$ and node i updates with probability δ_i, we can obtain the relation for $i\in\mathcal{V}$ as follows.

$$\begin{aligned}&\mathbb{E}\left[\left\|x_k^i-x^*\right\|^2|\Omega_{k-1}\right]\\&\leq\left(1-\frac{\lambda}{k}\right)\mathbb{E}\left[\left\|y_k^i-x^*\right\|^2|\Omega_{k-1}\right]-\frac{2\delta_i}{k}\left(F_i(y_k^i)-F_i(x^*)\right)\\&+\frac{8m^2\delta_i}{k^2p_*^2}\left(G^2+NGd_X+2d_X^2+\frac{4m^2\left(2N^2G^2+8d_X^2\right)}{k^2p_*^2}\right)\\&\leq\left(1-\frac{\lambda}{k}\right)\mathbb{E}\left[\left\|y_k^i-x^*\right\|^2|\Omega_{k-1}\right]-\frac{2p_*}{km}\left(F_i(y_k^i)-F_i(x^*)\right)\\&+\frac{8m^2}{k^2p_*^2}\left(G^2+NGd_X+2d_X^2+\frac{4m^2\left(2N^2G^2+8d_X^2\right)}{k^2p_*^2}\right),\end{aligned}\tag{19}$$

where the last inequality is based on the facts that $\delta_i\geq\frac{p_*}{m}$ and $\delta_i\leq\frac{N}{m}$. Summing up both sides of (19) over all the nodes $i\in\mathcal{V}$, applying Lemma A.4 and the definition in (1) yields

$$\begin{aligned}&\sum_{i=1}^{m}\mathbb{E}\left[\left\|x_k^i-x^*\right\|^2|\Omega_{k-1}\right]\leq\left(1-\frac{\lambda}{k}\right)\sum_{i=1}^{m}\left\|x_{k-1}^i-x^*\right\|^2\\&-\frac{2p_*}{km}\left[\left(\sum_{i=1}^{m}F_i(y_k^i)\right)-F(x^*)\right]\\&+\frac{8m^2N}{k^2p_*^2}\left(G^2+NGd_X+2d_X^2+\frac{4m^2\left(2N^2G^2+8d_X^2\right)}{k^2p_*^2}\right).\end{aligned}\tag{20}$$

By the convexity of function F_i and the bounded (sub)gradient assumption, the following relation can be obtained.

$$\begin{aligned}F_i\left(y_k^i\right)\geq&F_i\left(x_k^i\right)+\left(\tilde{\nabla}F_i(x_k^i)\right)^T\left(y_k^i-x_k^i\right)\\\geq&F_i\left(x_k^i\right)-G\left\|y_k^i-x_k^i\right\|.\end{aligned}\tag{21}$$

Now we need to bound the term $\left\|y_k^i - x_k^i\right\|$. Based on the nonexpansive property of the projection operation and the main computation step in the algorithm, we can have

$$\begin{aligned}\left\|y_k^i - x_k^i\right\| \leq & \alpha_{i,k}\left\|\tilde{\nabla} F_i(y_k^i) + e_k^i\right\| \\ \leq & \alpha_{i,k}\left(G + \alpha_{i,k}\left(NG + 2d_X\right)\right) \\ \leq & \frac{2mG}{kp_*} + \frac{4m^2\left(NG + 2d_X\right)}{k^2 p_*^2},\end{aligned} \tag{22}$$

where the last inequality is based on the relations in Lemma A.2. Applying (22) to (21) and then plugging it into (20) yields

$$\begin{aligned}&\sum_{i=1}^m \mathbb{E}\left[\left\|x_k^i - x^*\right\|^2 |\Omega_{k-1}\right] \leq \left(1 - \frac{\lambda}{k}\right)\sum_{i=1}^m \left\|x_{k-1}^i - x^*\right\|^2 \\ &+ \frac{8m^2 N}{k^2 p_*^2}\left(C_1 + \frac{p_* G^2}{km} + \frac{2p_* G d_X}{kmN} + \frac{C_2}{k^2 p_*^2}\right) \\ &- \frac{2p_*}{km}\left(\left[\sum_{i=1}^m F_i(x_k^i)\right] - F(x^*)\right),\end{aligned} \tag{23}$$

where $C_1 \equiv G^2 + NGd_X + 2d_X^2 + \frac{p_*^2 G^2}{2m^2 N}$, $C_2 \equiv 4m^2\left(2N^2G^2 + 8d_X^2\right)$.

Taking expectation on both sides yields

$$\begin{aligned}&\left(\mathbb{E}\left[\sum_{i=1}^m F_i(x_k^i)\right] - F(x^*)\right) \\ &\leq \frac{(k-\lambda)m}{2p_*}\mathbb{E}\sum_{i=1}^m \left\|x_{k-1}^i - x^*\right\|^2 - \frac{km}{2p_*}\mathbb{E}\sum_{i=1}^m \left\|x_k^i - x^*\right\|^2 \\ &+ \frac{8m^3 N}{kp_*^3}\left(C_1 + \frac{p_* G^2}{km} + \frac{2p_* G d_X}{kmN} + \frac{C_2}{k^2 p_*^2}\right).\end{aligned} \tag{24}$$

By summing from $k = 1$ to $k = T$, we obtain

$$\begin{aligned}&\frac{1}{T}\sum_{k=1}^T\left(\mathbb{E}\left[\sum_{i=1}^m F_i(x_k^i)\right]\right) - F(x^*) \\ &\leq \frac{O(1)}{T} + \frac{8m^3 N C_1}{Tp_*^3}\sum_{k=1}^T \frac{1}{k} \\ &+ \frac{8m^2 G\left(NG + d_X\right)}{Tp_*^2}\sum_{k=1}^T \frac{1}{k^2} + \frac{8m^3 N C_2}{Tp_*^3}\sum_{k=1}^T \frac{1}{k^3} \\ &\leq \frac{O(1) + O(\log T)}{T} \\ &= O\left(\frac{\log T}{T}\right).\end{aligned} \tag{25}$$

The first inequality in (25) is based on the relation as follows.

$$\begin{aligned}\frac{1}{T}\sum_{k=1}^T\Bigg(&\frac{(k-\lambda)m}{2p_*}\mathbb{E}\sum_{i=1}^m \left\|x_{k-1}^i - x^*\right\|^2 \\ &- \frac{km}{2p_*}\mathbb{E}\sum_{i=1}^m \left\|x_k^i - x^*\right\|^2\Bigg) \leq \frac{O(1)}{T}.\end{aligned}$$

The associated term does not affect the convergence rate result and is thus omitted in (25) in order to simply the notation. The last two terms in (25) are p-series with $p = 2$ and 3 and they are known to converge.

By the strongly convexity of function F_i, we can also infer that

$$\frac{1}{T}\sum_{k=1}^T\left(\mathbb{E}\left[\sum_{i=1}^m \left\|x_k^i - x^*\right\|\right]\right) \leq \left(\frac{\log T}{T}\right).$$

This completes the proof for Theorem IV.3. □

REFERENCES

[1] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," *CISCO white paper*, vol. 1, pp. 1–11, 2011.

[2] C. S. Inc., "Fog computing and the internet of things: Extend the cloud to where the things are," *CISCO white paper*, vol. 1, pp. 1–6, 2016.

[3] L. Lin, X. Liao, H. Jin, and P. Li, "Computation offloading toward edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1584–1607, 2019.

[4] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Toward computation offloading in edge computing: A survey," *IEEE Access*, vol. 7, pp. 131 543–131 558, 2019.

[5] H. Liao, Z. Zhou, X. Zhao, L. Zhang, S. Mumtaz, A. Jolfaei, S. H. Ahmed, and A. K. Bashir, "Learning-based context-aware resource allocation for edge-computing-empowered industrial iot," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4260–4277, 2020.

[6] Z. Zhou, H. Yu, C. Xu, Z. Chang, S. Mumtaz, and J. Rodriguez, "Begin: Big data enabled energy-efficient vehicular edge computing," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 82–89, 2018.

[7] J. Konečný, H. McMahan, F. Yu, P. Richtárik, A. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 10 2016.

[8] I. Matei and J. Baras, "Performance evaluation of the consensus-based distributed subgradient method under random communication topologies," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 4, pp. 754–771, Aug 2011.

[9] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *Automatic Control, IEEE Transactions on*, vol. 54, no. 1, pp. 48–61, Jan 2009.

[10] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, Dec 2013, pp. 6855–6860.

[11] ——, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *arXiv:1406.2075*, 2014.

[12] I.-A.Chen, "Fast distributed first-order methods," *PhD thesis, Massachusetts Institute of Technology*, 2012.

[13] J. M. F. M. Dusan Jakovetic, Joao Xavier, "Fast distributed gradient methods," *arXiv:1112.2972v4*, 2014.

[14] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2748–2761, July 2009.

[15] A. Nedic, "Asynchronous broadcast-based convex optimization over a network," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1337–1351, June 2011.

[16] N. Loizou and P. Richtárik, "Momentum and stochastic momentum for stochastic gradient, newton, proximal point and subspace descent methods," 2017.

[17] T. van Waterschoot and G. Leus, "Distributed estimation of static fields in wireless sensor networks using the finite element method," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 2853–2856.

[18] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 77–103, 2018. [Online]. Available: https://doi.org/10.1146/annurev-control-060117-105131

[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1561/2200000016

[20] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[21] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278 – 305, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1367578819300082

[22] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct 2016.
[23] ——, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
[24] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific; 2nd edition, 1999.
[25] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 109–128, 2018.
[26] J. Ahrenholz, "Comparison of core network emulation platforms," in *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*, Oct 2010, pp. 166–171.
[27] L. Zhao, "Fast decentralized data analytics in iot wireless networks," *IEEE Access*, vol. 7, pp. 181 531–181 545, 2019.
[28] G. del Pino, "The unifying role of iterative generalized least squares in statistical algorithms," *Statist. Sci.*, vol. 4, no. 4, pp. 394–403, 11 1989. [Online]. Available: https://doi.org/10.1214/ss/1177012408
[29] X. Yan and X. G. Su, *Linear Regression Analysis: Theory and Computing*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2009.
[30] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00401706.1970.10488634
[31] *Principles of Computerized Tomographic Imaging*. USA: Society for Industrial and Applied Mathematics, 2001.
[32] P. C. Hansen and M. Saxild-Hansen, "Air tools - a matlab package of algebraic iterative reconstruction methods," *Journal of Computational and Applied Mathematics*, vol. 236, no. 8, pp. 2167 – 2178, 2012, inverse Problems: Computation and Applications. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377042711005188
[33] "Enron email dataset," https://www.cs.cmu.edu/∼./enron/, accessed: 2020-10-22.
[34] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, 2003.
[35] D. P. Bertsekas, *Convex Optimization Theory*. Athena Scientific, 2009.

Liang Zhao received his Ph.D in computer science from Georgia State University, GA, USA and M.S in Electrical Engineering from Lehigh University, PA, USA in 2016 and 2012, respectively. He was a research data scientist at IBM (U.S). He is currently an assistant professor of Information Technology at Kennesaw State University, GA, USA. His current research interests are in the area of optimization and data analytics for distributed systems.

Fangyu Li is an assistant professor with the Department of Electrical and Computer Engineering at the Kennesaw State University (KSU). Before joining KSU, he was a postdoctoral fellow with the College of Engineering, University of Georgia. He received his PhD in Geophysics from University of Oklahoma in 2017. His Master (2013) and Bachelor (2009) degrees were both in Electrical Engineering, obtained from Tsinghua University and Beihang University, respectively. His research interests include seismic interpretation, subsurface imaging, signal processing, seismic imaging, machine learning, deep learning, distributed computing, Internet of things (IoT), and cyber-physical systems (CPS).

Maria Valero is an assistant professor in the College of Computing and Software Engineering, Department of Information Technology at Kennesaw State University (KSU). She received her Ph.D in Electrical and Computer Engineering at University of Georgia in 2019 and Master in Computer Science from University of Los Andes (Venezuela) in 2009. She was an associate professor at the University of Tachira (Venezuela) from 2004 to 2015. Her research interests include IoT for smarth healthcare, distributed computing, signal processing, wireless sensor networks, and cyber-physical systems.