Communication-Efficient and Privacy-Aware Distributed LMS Algorithm

Vinay Chakravarthi Gogineni*, Ashkan Moradi*, Naveen K. D. Venkategowda§, Stefan Werner*
*Dept. of Electronic Systems, Norwegian University of Science and Technology-NTNU, Norway
§Dept. of Science and Technology, Linköping University, Sweden
E-mails: {vinay.gogineni, ashkan.moradi, stefan.werner}@ntnu.no, naveen.venkategowda@liu.se

***Abstract*—This paper presents a private-partial distributed least mean square (PP-DLMS) algorithm that offers energy efficiency while preserving privacy and is suitable for applications with limited resources and strict security requirements. The proposed PP-DLMS allows every agent to exchange only a fraction of their perturbed data with neighbors during the collaboration process to minimize communication costs and guarantee privacy simultaneously. In order to understand how partial-sharing of perturbed data affects the learning performance, we conduct mean convergence analysis. Moreover, to investigate the privacy-preserving properties of the proposed algorithm, we characterize agent privacy in the presence of an honest-but-curious (HBC) adversary. Analytical results show that the proposed PP-DLMS is resilient against an HBC adversary by providing a fair energy-privacy trade-off compared to the conventional LMS algorithm. Numerical simulations corroborate the analytical findings.**

***Index Terms*—Distributed learning, energy-efficiency, privacy-preservation, average consensus, multiagent systems.**

I. Introduction

In the past decade, distributed computing systems have played a significant role in advancing signal processing and machine learning over multiagent networks [1]–[5]. The distributed network structure facilitates local communication between agents and their neighbors, thus enhancing the learning performance and robustness against dynamic changes in network topology. The local interactions among agents are realized via radio communication, which consumes large amounts of power and bandwidth. Local interactions are not only energy-intensive but also vulnerable to potential adversaries [6]. Thus, a distributed learning procedure that reduces the communication load as much as possible without significantly impairing the privacy of agents and overall estimation performance is always preferred.

Cryptography-based methods can provide secure communication between agents. However, they add substantial communication overhead and require considerable amounts of power [7]–[9], prohibiting their use in resource-constrained networks. Furthermore, cryptographic techniques are ineffective against privacy theft by dishonest network agents. Instead, low-complexity methods like noise injection-based mechanisms are attractive alternatives for preserving the privacy of individual agents [10]–[17]. In this category, differential-privacy techniques inject uncorrelated noise sequences into the information exchanged to ensure data privacy [10], [11]. The privacy-accuracy trade-off was improved in [15]–[18] by injecting correlated noise sequences with decaying variances into the exchanged information. Meanwhile, decomposition-based privacy-preserving techniques divide the private information into two substates, of which only one is shared among agents, hence making inference more difficult for adversaries [19], [20].

Distributed computing systems are often associated with limited computational and power resources, so resource-intensive local interactions should be minimized. This can be accomplished by performing dimensionality reduction [21] and 1-bit quantization [22] on the information before exchanging. Although these methods reduce communication costs, they are time-consuming and add additional computational burden to agents. Employing a probabilistic communication strategy is also an alternative solution to reduce local communication among agents [23]. Furthermore, partial-sharing concepts proposed in [27]–[29] reduce the consumption of resources by allowing agents to share only a fraction of information during each inter-agent interaction. The ease of implementation has made partial-sharing concepts popular in distributed learning. These communication-efficient methods, however, have not been investigated for privacy protection. To this end, in this paper, we propose a distributed learning framework that simultaneously attains both energy efficiency and privacy preservation.

This paper presents a private-partial distributed LMS (PP-DLMS) algorithm that enables agents to participate in local interactions by sharing only a fraction of their perturbed information, thus reducing resource consumption as well as preserving privacy. To investigate the impact of partial-sharing of perturbed data on the performance of distributed learning, we analyze the mean convergence and study the privacy of agents in the presence of an honest-but-curious (HBC) adversary. The HBC agent is a legitimate agent in the network that is curious about the private information of other agents. Since an HBC agent is a member of the network, it has access to the information exchanged in the neighborhood as well as to the information of the partial-sharing-based communication mechanism. As a result, the network becomes more vulnerable to information leakage. The privacy analysis shows that the proposed PP-DLMS provides a fair energy-privacy trade-off against HBC adversaries. Finally, we provide

The first two authors contributed equally to this work.

This work was supported by the Research Council of Norway.

numerical simulations that corroborate our analytical findings.

Mathematical notation: Scalars are denoted by lowercase letters, column vectors by bold lowercase, and matrices by bold uppercase. Superscripts $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{-1}$ denote the transpose and inverse operators, respectively. The symbol $\mathbf{1}_K$ represents the $K \times 1$ column vector with all entries equal to one and $\mathbf{I}_K$ is the $K \times K$ identity matrix. The right Kronecker product of two matrices is denoted by $\otimes$, while $\lambda_i(\mathbf{A})$ denotes the ith eigenvalue of matrix $\mathbf{A}$.

II. Background and Problem Formulation

Consider a sensor network modeled as a connected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where the node set $\mathcal{N}$ represents the agents of the network and $\mathcal{E}$ is the set of edges that represent bidirectional communication links between the nodes, i.e., $(k,l) \in \mathcal{E}$ if nodes k and l are connected. Additionally, the set $\mathcal{N}_k$ indicates the neighborhood of the node k that includes itself and the cardinality of the set $\mathcal{N}_k$ is denoted by $|\mathcal{N}_k|$, while $K = |\mathcal{N}|$ is the number of agents in the network. At time instant n and agent k, the input signal $\mathbf{x}_{k,n}$ and desired signal $y_{k,n}$ are assumed to be described as

$$y_{k,n} = \mathbf{x}_{k,n}^{\mathrm{T}}\, \mathbf{w}^{\star} + \epsilon_{k,n}, \tag{1}$$

where $\mathbf{w}^{*} \in \mathbb{R}^{L}$ is an optimal parameter vector to be estimated, $\mathbf{x}_{k,n} = [x_{k,n}, x_{k,n-1}, \ldots, x_{k,n-L+1}]^{\mathrm{T}}$ is the input signal vector, and the observation noise $\epsilon_{k,n}$ is a zero-mean Gaussian random sequence. The estimate of $\mathbf{w}^{\star}$ at time instant n, i.e., $\mathbf{w}_n$ is chosen so that it minimizes

$$\mathcal{J}_n = \frac{1}{K} \sum_{k \in \mathcal{N}} \mathbb{E}[e_{k,n}^2], \tag{2}$$

where $e_{k,n} = y_{k,n} - \hat{y}_{k,n}$ with $\hat{y}_{k,n}$ as the estimated filter output at agent k. At every time instant n, $\mathbf{w}_n$ can be updated via steepest-descent approach as

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \frac{\eta}{2} \nabla \mathcal{J}_n = \mathbf{w}_n + \eta \sum_{k \in \mathcal{N}} e_{k,n} \mathbf{x}_{k,n}, \tag{3}$$

where η is the step size. The operation in (3) can be modeled as $\mathbf{w}_{n+1} = \frac{1}{K} \sum\limits_{k \in \mathcal{N}} \boldsymbol{\psi}_{k,n+1}$ with

$$\boldsymbol{\psi}_{k,n+1} = \mathbf{w}_n + \mu\, e_{k,n}\, \mathbf{x}_{k,n}, \tag{4}$$

being the intermediate estimate of $\mathbf{w}^{\star}$ at node k and time instant n, and $\mu = \eta K$ is the new step size. The average of the intermediate estimate $\boldsymbol{\psi}_{k,n+1}$ across the entire network can be evaluated in a distributed manner using an average consensus filter (ACF) [24]–[26].

In the process of obtaining an average consensus, agents exchange local information $\boldsymbol{\psi}_{k,n+1}$ with their neighbors that contains node-sensitive information and might be exploited by potential adversaries. To protect the node-sensitive information from being inferred by adversaries, agents exchange perturbed versions of their private information [15]–[17]. Thus, the state of the ACF after m consensus iterations is

$$\mathbf{h}_{k,(m)} = \sum_{l \in \mathcal{N}_k} a_{lk} \tilde{\mathbf{h}}_{l,(m-1)}, \tag{5}$$

where a_{lk} is the consensus weight between agents l and k, $\tilde{\mathbf{h}}_{l,(m-1)} = \mathbf{h}_{l,(m-1)} + \boldsymbol{\omega}_{l,(m-1)}$ is the perturbed local information with $\mathbf{h}_{l,(0)} = \boldsymbol{\psi}_{l,n+1}$, and $\boldsymbol{\omega}_{l,(m-1)}$ is the perturbation noise at agent l and $(m-1)$th consensus iteration [15]. The perturbation noise at agent l and consensus iteration m is given by

$$\boldsymbol{\omega}_{l,(m)} = \begin{cases} \boldsymbol{\nu}_{l,(0)}, & m = 0 \\ \phi^{m} \boldsymbol{\nu}_{l,(m)} - \phi^{m-1} \boldsymbol{\nu}_{l,(m-1)}, & \text{otherwise}, \end{cases} \tag{6}$$

where constant $\phi \in (0,1)$ is same for all agents, and $\boldsymbol{\nu}_{l,(m)} \in \mathbb{R}^{L}$ is a zero-mean Gaussian sequence with $\mathbb{E}[\boldsymbol{\nu}_{l,(m)} \boldsymbol{\nu}_{l,(m)}^{\mathrm{T}}] = \sigma_{\nu}^2 \mathbf{I}_L$. If $\mathbf{A}$ with $[\mathbf{A}]_{l,k} = a_{lk}$ is a doubly stochastic matrix that satisfies the conditions stated in [25] and the perturbation noise follows (6), all agents reach consensus on the exact average, given by

$$\lim_{m \to \infty} \mathbf{h}_{k,(m)} = \frac{1}{K} \sum_{l \in \mathcal{N}} \mathbf{h}_{l,(0)}, \tag{7}$$

asymptotically.

III. PP-DLMS Algorithm

As shown in (5), the collaboration between agents is vital for distributed learning. Privacy-preserving distributed learning techniques are no exception. However, although collaboration among agents improves learning accuracy, it is resource-intensive. As nodes in sensor networks have limited battery power, reducing the inter-node communication overhead is essential while maintaining inter-node cooperation benefits. By promoting partial-sharing [27]–[29] among agents in privacy-preserving distributed learning systems, we aim to achieve both privacy and energy efficiency in a single framework.

In the proposed PP-DLMS, during each consensus iteration m, every agent shares only a portion of the perturbed version of its private information with neighbors (i.e., M out of L entries in $\mathbf{h}_{k,(m)}$) to reduce the communication load while maintaining privacy. The entry selection procedure at each agent k is characterized by a diagonal selection matrix of size $L \times L$, the main diagonal of which consists of M numbers of ones and $L - M$ numbers of zeros. The selection matrix of agent k at time instant n and consensus iteration m is denoted by $\mathbf{S}_{k,n,(m)}$, where the position of ones indicates which entries of the private information are to be shared with neighbors. The selection of M out of L entries can be made stochastically, or, sequentially as in [27], [28]. We adopt a coordinated partial-sharing scheme, which is a special case of sequential and stochastic partial-sharing methods [28]. In coordinated partial-sharing, all agents are initialized with the same selection matrices, i.e., $\mathbf{S}_{1,0,(0)} = \mathbf{S}_{2,0,(0)} \cdots \mathbf{S}_{K,0,(0)}$. Since we are using the coordinated partial-sharing, we drop node index in $\mathbf{S}_{k,n,(m)}$ and continue with $\mathbf{S}_{n,(m)}$. Additionally, the selection matrix at the current consensus iteration, i.e., $\mathbf{S}_{n,(m)}$, can be obtained by applying a right-circular shift operation on the main diagonal elements of the selection matrix during the previous consensus iteration, i.e., $\mathbf{S}_{n,(m-1)}$. We also consider $\mathbf{S}_{n,(0)} = \mathbf{S}_{n-1,(m)}$ at each time index n. This process has an entry-sharing probability of $p = \frac{M}{L}$ because each entry will be

Algorithm 1: Private-Partial DLMS (PP-DLMS)

```
• For each agent $k \in \mathcal{N}$
Initialize: $\mathbf{S}_{n,(0)}, \tau$,
$\hat{y}_{k,n} = \mathbf{x}_{k,n}^{\mathrm{T}} \mathbf{w}_{k,n}$
$e_{k,n} = y_{k,n} - \hat{y}_{k,n}$
Local Update:
    $\boldsymbol{\psi}_{k,n+1} = \mathbf{w}_{k,n} + \mu\, \mathbf{x}_{k,n}\, e_{k,n}$
Average Consensus Update:
Set $\mathbf{h}_{k,(0)} = \boldsymbol{\psi}_{k,n+1}$
For $m = 1$ to $T$
Perturb the local data $\tilde{\mathbf{h}}_{k,(m-1)} = \mathbf{h}_{k,(m-1)} + \boldsymbol{\omega}_{k,(m-1)}$
Share $\mathbf{S}_{n,(m-1)} \tilde{\mathbf{h}}_{k,(m-1)}$
Receive $\left\{ \mathbf{S}_{n,(m-1)} \tilde{\mathbf{h}}_{l,(m-1)} : \forall l \in \mathcal{N}_k^- \right\}$
 $\mathbf{h}_{k,(m)} = a_{kk} \tilde{\mathbf{h}}_{k,(m-1)} + \sum_{l \in \mathcal{N}_k^-} a_{lk} \left( \mathbf{S}_{n,(m-1)} \tilde{\mathbf{h}}_{l,(m-1)} + (\mathbf{I} - \mathbf{S}_{n,(m-1)}) \tilde{\mathbf{h}}_{k,(m-1)} \right)$
$\mathbf{S}_{n,(m)} = \text{circularshift}\left( \mathbf{S}_{n,(m-1)}, \tau \right)$
Endfor
$\mathbf{w}_{k,n+1} = \mathbf{h}_{k,(T)}$
```

shared M times during L subsequent iterations. By using the selection matrices, the privacy-preserving average consensus state update at each agent k can be expressed alternatively as

$$\mathbf{h}_{k,(m)} = a_{kk} \tilde{\mathbf{h}}_{k,(m-1)} + \sum_{l \in \mathcal{N}_k^-} a_{lk} \big(\mathbf{S}_{n,(m-1)} \tilde{\mathbf{h}}_{l,(m-1)} + (\mathbf{I} - \mathbf{S}_{n,(m-1)}) \tilde{\mathbf{h}}_{l,(m-1)} \big), \tag{8}$$

where $\mathcal{N}_k^-$ indicates the neighborhood of node k excluding itself. As a result of partial information sharing, agents do not have access to the portion of the information that was not shared. However, by allowing each node to use its own internal information instead of the unshared information of neighboring agents, this challenge can be solved. At each agent k, we therefore substitute $(\mathbf{I} - \mathbf{S}_{n,(m-1)}) \tilde{\mathbf{h}}_{k,(m-1)}$ in the place of $(\mathbf{I} - \mathbf{S}_{n,(m-1)}) \tilde{\mathbf{h}}_{l,(m-1)}$ for each $l \in \mathcal{N}_k^-$ as

$$\mathbf{h}_{k,(m)} = a_{kk} \tilde{\mathbf{h}}_{k,(m-1)} + \sum_{l \in \mathcal{N}_k^-} a_{lk} \left(\mathbf{S}_{n,(m-1)} \tilde{\mathbf{h}}_{l,(m-1)} + (\mathbf{I} - \mathbf{S}_{n,(m-1)}) \tilde{\mathbf{h}}_{k,(m-1)} \right) \cdot \tag{9}$$

After a sufficient number of consensus iterations, say T, the parameter vector $\mathbf{w}_{k,n}$ is updated to $\mathbf{w}_{k,n+1} = \mathbf{h}_{k,(T)}$. The workflow of the proposed PP-DLMS is summarized in Algorithm 1.

IV. Performance Analysis

In this section, we examine the impact of partial sharing of information on convergence and privacy.

A. Network Global Model

At each time instant n, we define the optimal model parameter vector $\mathbf{w}_{net}^{\star} = \mathbf{1}_K \otimes \mathbf{w}^{\star}$, estimated model parameter vector $\mathbf{w}_{net,n} = \text{col}\{\mathbf{w}_{1,n}, \mathbf{w}_{2,n}, \ldots, \mathbf{w}_{K,n}\}$, input data matrix $\mathbf{X}_n = \text{blockdiag}\{\mathbf{x}_{1,n}, \mathbf{x}_{2,n}, \ldots, \mathbf{x}_{K,n}\}$, observation noise vector $\boldsymbol{\epsilon}_{net,n} = \text{col}\big\{\epsilon_{1,n}, \epsilon_{2,n}, \ldots, \epsilon_{K,n}\big\}$, and private information

$$\begin{aligned} \mathbf{h}_{(0)} &= \text{col}\{\mathbf{h}_{1,(0)}, \mathbf{h}_{2,(0)}, \ldots, \mathbf{h}_{K,(0)}\} \\ &= \text{col}\{\boldsymbol{\psi}_{1,n}, \boldsymbol{\psi}_{2,n}, \ldots, \boldsymbol{\psi}_{K,n}\}, \end{aligned} \tag{10}$$

where the column-wise stacking and block diagonalization operations are represented by col$\{\cdot\}$ and blockdiag$\{\cdot\}$, respectively. Using the above definitions, data model and error vector at network-level are

$$\begin{aligned} \mathbf{y}_n &= \text{col}\{y_{1,n}, y_{2,n}, \ldots, y_{K,n}\} = \mathbf{X}_n^{\mathrm{T}} \mathbf{w}_{net}^{\star} + \boldsymbol{\epsilon}_n \\ \mathbf{e}_n &= \text{col}\big\{e_{1,n}, e_{2,n}, \ldots, e_{K,n}\big\} = \mathbf{y}_n - \mathbf{X}_n^{\mathrm{T}} \mathbf{w}_{net,n}. \end{aligned} \tag{11}$$

According to definitions in (11), the average consensus state update in (9), and

$$\boldsymbol{\psi}_{k,n+1} = \mathbf{w}_{k,n} + \mu\, \mathbf{x}_{k,n}\, e_{k,n}, \tag{12}$$

the network-level model of the PP-DLMS can be stated as

$$\mathbf{w}_{net,n+1} = \boldsymbol{\mathcal{B}}_n \big(\mathbf{w}_{net,n} + \mu \mathbf{X}_n \mathbf{e}_n \big) + \mathbf{c}_n \tag{13}$$

with

$$\boldsymbol{\mathcal{B}}_n = \prod_{i=0}^{m-1} \boldsymbol{\mathcal{B}}_{n,(i)} \quad \text{and} \quad \mathbf{c}_n = \sum_{i=0}^{m-1} \Big(\prod_{j=i}^{m-1} \boldsymbol{\mathcal{B}}_{n,(j)} \Big) \boldsymbol{\omega}_{(i)}, \tag{14}$$

where $\boldsymbol{\mathcal{B}}_{n,(m)} = \mathbf{A} \otimes \mathbf{S}_{n,(m)} + \mathbf{I}_K \otimes (\mathbf{I}_L - \mathbf{S}_{n,(m)})$, $\boldsymbol{\omega}_{(i)} = \text{col}\{\boldsymbol{\omega}_{1,(i)}, \boldsymbol{\omega}_{2,(i)}, \ldots, \boldsymbol{\omega}_{K,(i)}\}$, and the network-level perturbation noise vector is given by

$$\boldsymbol{\omega}_{(i)} = \begin{cases} \boldsymbol{\nu}_{(0)}, & i = 0 \\ \phi^i \boldsymbol{\nu}_{(i)} - \phi^{i-1} \boldsymbol{\nu}_{(i-1)}, & \text{otherwise}, \end{cases} \tag{15}$$

where $\boldsymbol{\nu}_{(i)} = \text{col}\{\boldsymbol{\nu}_{1,(i)}, \boldsymbol{\nu}_{2,(i)}, \ldots, \boldsymbol{\nu}_{K,(i)}\}$. In order to obtain the convergence condition for PP-DLMS, we assume the following:

- **A1.** For all $k \in \mathcal{N}$, the input signal vector $\mathbf{x}_{k,n}$ is drawn from a WSS multivariate random sequence with correlation matrix $\mathbf{R}_k = \mathrm{E}[\mathbf{x}_{k,n} \mathbf{x}_{k,n}^{\mathrm{T}}]$; in addition, the input signal vectors $\mathbf{x}_{k,n}$ and $\mathbf{x}_{l,m}$ are independent for all $k \neq l$ and $n \neq m$.
- **A2.** The noise process $\epsilon_{k,n}$ is assumed to be zero-mean i.i.d. and independent of any other quantity.
- **A3.** For all $k \in \mathcal{N}$, the selection matrix $\mathbf{S}_{n,(m)}$ is assumed to be independent of any other data.

B. First-order Convergence

Considering $\tilde{\mathbf{w}}_{net,n} = \mathbf{w}_{net}^{\star} - \mathbf{w}_{net,n}$, and using the fact that $\mathbf{w}_{net}^{\star} = \boldsymbol{\mathcal{B}}_n \mathbf{w}_{net}^{\star}$ (since $\boldsymbol{\mathcal{B}}_{n,(m)} \mathbf{w}_{net}^{\star} = \mathbf{w}_{net}^{\star}$ for all m), then form (13), $\tilde{\mathbf{w}}_{net,n+1}$ can be recursively expressed as

$$\tilde{\mathbf{w}}_{net,n+1} = \boldsymbol{\mathcal{B}}_n \big(\mathbf{I}_{LK} - \mu \mathbf{X}_n \mathbf{X}_n^{\mathrm{T}} \big) \tilde{\mathbf{w}}_{net,n} - \mu \boldsymbol{\mathcal{B}}_n \mathbf{X}_n \boldsymbol{\epsilon}_{net,n} - \mathbf{c}_n. \tag{16}$$

Applying expectation $\mathbb{E}[\cdot]$ on the both sides of (16) and using the assumptions $\mathbf{A1}-\mathbf{A3}$, we obtain

$$\mathbb{E}[\tilde{\mathbf{w}}_{net,n+1}] = \mathbb{E}[\boldsymbol{\mathcal{B}}_n]\big(\mathbf{I}_{LK} - \mu\boldsymbol{\mathcal{R}}\big)\mathbb{E}[\tilde{\mathbf{w}}_{net,n}], \tag{17}$$

where $\boldsymbol{\mathcal{R}} = \mathbb{E}[\mathbf{X}_n\mathbf{X}_n^{\mathrm{T}}] = \text{blockdiag}\{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_K\}$. From (17), one can see that $\lim_{n\to\infty}\mathbb{E}[\tilde{\mathbf{w}}_{net,n}]$ attains finite value if and only if $\|\mathbb{E}[\boldsymbol{\mathcal{B}}_n]\big(\mathbf{I}_{LK} - \mu\boldsymbol{\mathcal{R}}\big)\| < 1$ for all n, where $\|\cdot\|$ is any matrix norm. Here, we use the block maximum norm of the matrix, i.e., $\|\cdot\|_{b,\infty}$ in [30], to obtain the mean convergence condition. From the properties of block maximum norm, one can obtain

$$\|\mathbb{E}[\boldsymbol{\mathcal{B}}_n]\big(\mathbf{I}_{LK} - \mu\boldsymbol{\mathcal{R}}\big)\|_{b,\infty} \leq \|\mathbb{E}[\boldsymbol{\mathcal{B}}_n]\|_{b,\infty}\|\mathbf{I}_{LK} - \mu\boldsymbol{\mathcal{R}}\|_{b,\infty}\cdot$$

Additionally, we have

$$\|\mathbb{E}[\boldsymbol{\mathcal{B}}_n]\|_{b,\infty} = \|\prod_{i=0}^{m-1}\mathbb{E}\left[\mathbf{B}_{n,(i)}\right]\|_{b,\infty} \leq \prod_{i=0}^{m-1}\|\mathbb{E}[\mathbf{B}_{n,(i)}]\|_{b,\infty} \leq 1,$$

and using the similar procedure in [27], [28], one can prove that

$$\|\mathbb{E}[\mathbf{B}_{n,(i)}]\|_{b,\infty} = \|p(\mathbf{A}\otimes\mathbf{I}_L) + (1-p)\mathbf{I}_{LK}\|_{b,\infty} \leq 1\cdot$$

By using [31, Lemma D. 5], it is seen that $\mathbb{E}\big[\tilde{\mathbf{w}}_{net,n}\big]$ converges under the condition $\rho\big(\mathbf{I}_{LK}-\mu\boldsymbol{\mathcal{R}}\big) < 1$, or, equivalently, $\forall k, i : |1-\mu\lambda_i(\mathbf{R}_k)| < 1$, where $\rho(\cdot)$ denotes the spectral radius of the argument matrix. As a result, we obtain the mean convergence condition as

$$0 < \mu < \frac{2}{\max\limits_{\forall i,k}\{\lambda_i(\mathbf{R}_k)\}}\cdot \tag{18}$$

Accordingly, as long as the step size μ satisfies (18), the operations will converge in the mean.

C. Privacy Analysis

This section examines the privacy of agents in the presence of an HBC agent. The HBC agent is an adversary, but a legitimate agent of the network that has access to information associated with the selection of elements in the partial sharing process and consequently increases the likelihood of information leakage. Let us assume that agent k is an HBC agent trying to estimate the private information of other agents at each time instant n, i.e., $\mathbf{h}_{l,(0)} = \boldsymbol{\psi}_{l,n+1}$ for $l \in \mathcal{N}\setminus\{k\}$. The privacy of agent l is defined as the mean squared estimation error at the adversary attempting to infer the private information as

$$\mathcal{E}_{l,(m)} \triangleq \text{tr}\left(\mathbb{E}[(\hat{\mathbf{h}}_{l,(m)} - \mathbf{h}_{l,(0)})(\hat{\mathbf{h}}_{l,(m)} - \mathbf{h}_{l,(0)})^{\mathrm{T}}]\right) \tag{19}$$

where $\hat{\mathbf{h}}_{l,(m)}$ denotes the estimate of the private information $\mathbf{h}_{l,(0)}$ after m consensus iterations at the adversary.

The HBC agent has access to its own information and the information exchanged in the neighborhood at each consensus iteration m, i.e., $\{\mathbf{h}_{k,(m)}, \mathbf{S}_{n,(m)}, \mathbf{S}_{n,(m)}\tilde{\mathbf{h}}_{l,(m)}\}$, for $l \in \mathcal{N}_k^-$. Since the HBC agent already knows its own information, the corresponding entries are removed from $\boldsymbol{\omega}_{(m)}, \boldsymbol{\nu}_{(m)}, \mathbf{h}_{(0)}, \text{and}, \boldsymbol{\mathcal{B}}_{n,(m)}$, and denote the quantities with reduced dimensions as $\breve{\boldsymbol{\omega}}_{(m)}, \breve{\boldsymbol{\nu}}_{(m)}, \breve{\mathbf{h}}_{(0)}, \text{and}, \breve{\boldsymbol{\mathcal{B}}}_{n,(m)}$, respectively. From (9), the network-level consensus operation with reduced dimensions can be stated as

$$\tilde{\mathbf{h}}_{(m)} = \Big(\prod_{i=0}^{m}\breve{\boldsymbol{\mathcal{B}}}_{n,(i)}\Big)\breve{\mathbf{h}}_{(0)} + \sum_{i=0}^{m}\Big(\prod_{j=i}^{m}\breve{\boldsymbol{\mathcal{B}}}_{n,(j)}\Big)\breve{\boldsymbol{\omega}}_{(i)}\cdot \tag{20}$$

Without loss of generality, we consider the case where agent K is an HBC agent. At the HBC agent, let $\boldsymbol{\theta}_{(m)} = \mathbf{C}\tilde{\mathbf{h}}_{(m)}$ be the observation vector that comprises the information captured at mth consensus iteration with $\mathbf{C} = \bar{\mathbf{C}}^{\mathrm{T}}\otimes\mathbf{I}_L$ where columns of $\bar{\mathbf{C}} \in \mathbb{R}^{(K-1)\times|\mathcal{N}_K^-|}$ consist of the canonical vectors corresponding to neighbors of agent K. The canonical vector corresponding to agent l, $\mathbf{e}_l \in \mathbb{R}^{K-1}$, is a vector with 1 in the lth entry and zeros elsewhere. Then, following similar procedure as in [15] and substituting (6) in (9), observation model at the HBC agent, after m consensus iterations, is described as

$$\boldsymbol{\vartheta}_{(m)} = \mathbf{H}_{(m)}\breve{\mathbf{h}}_{(0)} + \mathbf{F}_{(m)}\boldsymbol{\upsilon}_{(m)} \tag{21}$$

where $\boldsymbol{\vartheta}_{(m)} = \text{col}\{\boldsymbol{\theta}_{(0)}, \cdots, \boldsymbol{\theta}_{(m)}\}$, $\mathbf{H}_{(m)} = \text{col}\{\boldsymbol{\mathcal{H}}_{(0)}, \cdots, \boldsymbol{\mathcal{H}}_{(m)}\}$ with $\boldsymbol{\mathcal{H}}_{(m)} = \mathbf{C}\prod_{i=0}^{m}\breve{\boldsymbol{\mathcal{B}}}_{n,(i)}$, $\breve{\mathbf{h}}_{(0)} = \text{col}\{\mathbf{h}_{1,(0)}, \cdots, \mathbf{h}_{K-1,(0)}\}$, $\boldsymbol{\upsilon}_{(m)} = \text{col}\{\breve{\boldsymbol{\nu}}_{(0)}, \cdots, \breve{\boldsymbol{\nu}}_{(m)}\}$, and

$$\mathbf{F}_{(m)} = \begin{bmatrix} \mathbf{C}\breve{\boldsymbol{\mathcal{B}}}_{n,(0)} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}\mathbf{F}_{(1),(0)} & \phi\mathbf{C}\breve{\boldsymbol{\mathcal{B}}}_{n,(1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}\mathbf{F}_{(2),(0)} & \phi\mathbf{C}\mathbf{F}_{(2),(1)} & \phi^2\mathbf{C}\breve{\boldsymbol{\mathcal{B}}}_{n,(2)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}\mathbf{F}_{(m),(0)} & \phi\mathbf{C}\mathbf{F}_{(m),(1)} & \phi^2\mathbf{C}\mathbf{F}_{(m),(2)} & \cdots & \phi^m\mathbf{C}\breve{\boldsymbol{\mathcal{B}}}_{n,(m)} \end{bmatrix}$$

with $\mathbf{F}_{(m),(i)} = \prod_{t=i+1}^{m}\breve{\boldsymbol{\mathcal{B}}}_{n,(t)}(\breve{\boldsymbol{\mathcal{B}}}_{n,(i)} - \mathbf{I})$. Using the model in (21) the HBC agent can obtain the maximum likelihood (ML) estimate of $\breve{\mathbf{h}}_{(0)}$, with associated error covariance

$$\mathbf{P}_{(m)} = \Big(\mathbf{H}_{(m)}^{\mathrm{T}}\big(\mathbf{F}_{(m)}\boldsymbol{\Gamma}\mathbf{F}_{(m)}^{\mathrm{T}}\big)^{-1}\mathbf{H}_{(m)}\Big)^{-1} \tag{22}$$

where $\boldsymbol{\Gamma} = \mathbb{E}\{\boldsymbol{\upsilon}_{(m)}\boldsymbol{\upsilon}_{(m)}^{\mathrm{T}}\} = \sigma_\nu^2\mathbf{I}$. As the HBC agent collects more information from neighbors, the mean squared error of the ML estimator decreases and the privacy metric (19) at each agent k is obtained as

$$\mathcal{E}_{k,(m)} = \text{tr}\left((\mathbf{e}_k^{\mathrm{T}}\otimes\mathbf{I}_L)\mathbf{P}_{(m)}(\mathbf{e}_k\otimes\mathbf{I}_L)\right). \tag{23}$$

V. Numerical Simulations

To demonstrate the effectiveness of PP-DLMS, we conducted simulations for identifying an unknown system of length $L = 32$. For this, we considered a network of $K = 5$ agents with the adjacency matrix of

$$\mathbf{E} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix},$$

as in [15]. The input signal $x_{k,n}$ and observation noise sequence $\epsilon_{k,n}$, were drawn from zero-mean Gaussian distribution with variance $\sigma_x^2 = 1$ and $\sigma_\epsilon^2 \in \mathcal{U}(0.008, 0.03)$ where $\mathcal{U}(\cdot)$ is the uniform distribution. The average consensus weights

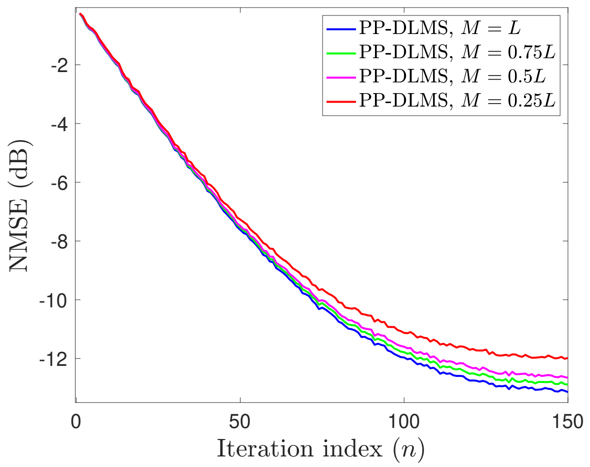

Fig. 1: Network-level MSE (in dB) versus time.

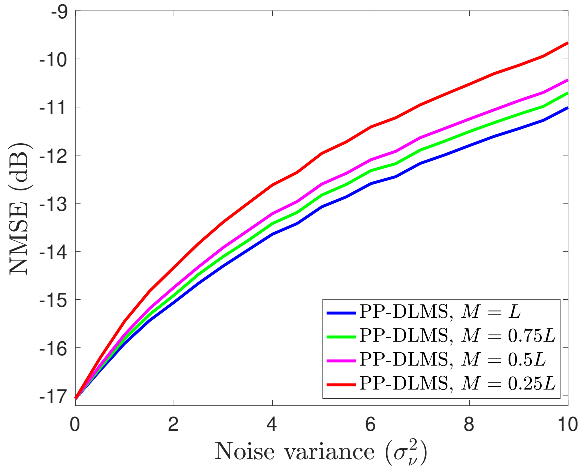

Fig. 2: Network-level MSE (in dB) for different values of M, i.e., the portion of the shared information, versus injected noise variance σ_ν^2.

are non-negative coefficients and were obtained through the Metropolis rule [25]. The ACF was iterated for $T = 40$ iterations to approximate the required averages and the perturbation noise sequence at each agent follows (6) with $\phi = 0.9$. The proposed PP-DLMS was simulated under coordinated partial-sharing scheme for different values of M (say $0.75L$, $0.5L$, $0.25L$) and the network-level MSE (NMSE) was considered as the performance metric. The results were obtained against the injected noise variance σ_ν^2, by averaging over 500 independent experiments.

Firstly, the learning curves (i.e., NMSE in dB vs iteration index n) for perturbation noise variance $\sigma_\nu^2 = 5$ are shown in the Fig. 1. Next, for different values of σ_ν^2, the steady-state NMSE is displayed in Fig. 2. From these plots, it can be observed that the proposed PP-DLMS scheme simultaneously achieves energy efficiency and privacy at the cost of a slight degradation in the NMSE. This performance degradation is inversely proportional to the amount of information shared during the average consensus operations. The degradation in performance increases with less information shared at each iteration, smaller M, resulting in a larger NMSE.

Finally in the presence of an HBC agent, agent 5 in the network, the privacy metric (23) versus σ_ν^2 for different values of M is illustrated in Fig. 3. A similar breach of privacy occurs with agent 4 as in [15], and agent 3 obtains identical privacy as agent 1 due to symmetric topology, they are omitted in Fig. 3. From Fig. 3, it can be seen that the proposed PP-DLMS provides a reasonable privacy-energy trade-off. For the case of sharing $M = 0.75L$, the algorithm achieves the same level of privacy as in the case of full information sharing. In the case of sharing less information, $M = 0.5L$ and $M = 0.25L$, the level of privacy decreases, however since smaller portions of information are shared at each consensus iteration, the HBC agent must collect information for more consensus iterations to accurately estimate the private information of other agents.

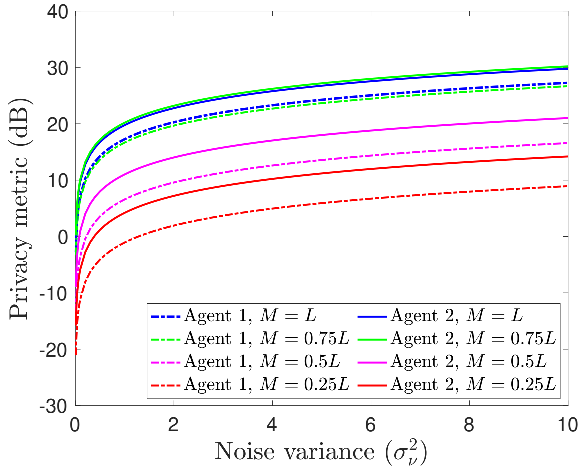

Fig. 3: Agent privacy (in dB) for different values of M versus injected noise variance σ_ν^2.

VI. Conclusions

This paper proposed an energy-efficient and privacy-preserving distributed LMS algorithm. By allowing each agent to share only a fragment of perturbed local information with its neighbors, the proposed private-partial distributed LMS (PP-DLMS) simultaneously achieved both energy-efficiency and privacy-preservation. A mean-convergence analysis of the proposed PP-DLMS algorithm has been conducted to examine the impact of partial-sharing of information on the estimation performance. Further, agent privacy has been characterized in the presence of an honest-but-curious (HBC) adversary, in order to investigate the privacy-preserving properties of the proposed algorithm. Analytical results revealed that the PP-DLMS is resilient to the perturbation sequence and provides a fair energy-privacy trade-off against HBC agents. Numerical simulations have validated the analytical findings.

References

[1] J. B Predd, S. B Kulkarni, and H. V Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Process. Mag.*, Vol. 23, no. 4, pp. 56–69, Jul., 2006.

[2] T.H. Chang, M. Hong, H.T. Wai, X. Zhang, and S. Lu, "Distributed learning in the nonconvex world: From batch data to streaming and beyond," *IEEE Signal Process. Mag.*, Vol. 37, no. 3, pp. 26–38, May, 2020.

[3] T.H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Trans. Signal Process.*, Vol. 63, no. 2, pp. 482–497, Jan., 2014.

[4] K. Yuan, B. Ying, X. Zhao, and A. H Sayed, "Exact diffusion for distributed optimization and learning—Part I: Algorithm development," *IEEE Trans. Signal Process.*, Vol. 67, no. 3, pp. 708–723, Feb., 2018.

[5] K. Yuan, B. Ying, X. Zhao, and A. H Sayed, "Exact diffusion for distributed optimization and learning—Part II: Convergence analysis," *IEEE Trans. Signal Process.*, Vol. 67, no. 3, pp. 724–739, Feb., 2018.

[6] Q. Li, J. S. Gundersen, R. Heusdens, and M. G. Christensen, "Privacy-preserving distributed processing: metrics, bounds and algorithms," *IEEE Trans. Inf. Forensics Security.*, Vol. 16, no. 3, pp. 2090–2103, Jan., 2021.

[7] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Process. Mag.*, Vol. 30, no. 1, pp. 82–105, Jan., 2013.

[8] K. Kogiso, and T. Fujita, "Cyber-security enhancement of networked control systems using homomorphic encryption," *Proc. 54th IEEE Conf. Decis. and Control*, pp. 6836–6843, 2015.

[9] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," *Springer Annu. Cryptology Conf.*, pp. 643–662, 2012.

[10] J. He and L. Cai and X. Guan, "Differential private noise adding mechanism and its application on consensus algorithm," *IEEE Trans. Signal Process.*, Vol. 68, pp. 4069-4082, Jul., 2020.

[11] E. Nozari, P. Tallapragada, J. Cortés, "Differentially private average consensus: obstructions, trade-offs, and optimal algorithm design," *Elsevier Automatica*, Vol. 81, pp. 221–231, Jul., 2017.

[12] N. K. D. Venkategowda, S. Werner, "Privacy-preserving distributed maximum consensus," *IEEE Signal Process. Lett.*, Vol. 27, pp. 1839–1843, Oct., 2020.

[13] A. Moradi, N. K. Venkategowda, S. Werner, "Coordinated data- falsification attacks in consensus-based distributed Kalman filtering," *Proc. 8th IEEE Int. Workshop Comput. Advances Multi-Sensor Adaptive Process.*, pp. 495–499, 2019.

[14] A. Moradi, N. K. Venkategowda, S. P. Talebi, S. Werner, "Distributed Kalman Filtering with Privacy against Honest-but-Curious Adversaries," *Proc. 55th IEEE Asilomar Conf. Signals, Syst., Computers*, pp. 790–794, 2021.

[22] S. Xie, H. Li, "Distributed LMS estimation over networks with quantized communications," *Int. J. Control*, Vol. 86, no. 3, pp. 478–492, Apr., 2013.

[15] Y. Mo and R.M. Murray, "Privacy preserving average consensus," *IEEE Trans. Autom. Control*, Vol. 62, no. 2, pp. 753–765, Feb., 2017.

[16] J. He, L. Cai, X. Guan, "Preserving data-privacy with added noises: Optimal estimation and privacy analysis," *IEEE Trans. Inf. Theory*, Vol. 64, no. 8, pp. 5677–5690, Aug., 2018.

[17] J. He, L. Cai, C. Zhao, P. Cheng, and X. Guan, "Privacy-preserving average consensus: privacy analysis and algorithm design," *IEEE Trans. Signal Inf. Process. Netw.*, Vol. 5, no. 1, pp. 127–138, Mar., 2019.

[18] A. Moradi, N. K. Venkategowda, S. P. Talebi, S. Werner, "Securing the Distributed Kalman Filter Against Curious Agents," *Proc. 24th IEEE Int. Conf. Inf. Fusion*, pp. 1–7, 2021.

[19] Y. Wang, "Privacy-preserving average consensus via state decomposition," *IEEE Trans. Autom. Control*, Vol. 64, no. 11, pp. 4711–4716, Nov., 2019.

[20] W. Wang, D. Li, X. Wu, and S. Xue, "Average consensus for switching topology networks with privacy protection," *Proc. IEEE Chinese Automat. Congr.*, pp. 1098–1102, 2019.

[21] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Trading off complexity with communication costs in distributed adaptive learning via Krylov subspaces for dimensionality reduction," *IEEE J. Sel. Topics Signal Process.*, Vol. 7, no. 2, pp. 257–273, Apr., 2013.

[23] C. G. Lopes, and A. H. Sayed, "Diffusion adaptive networks with changing topologies," *IEEE Int. Conf. Acoust., Speech Signal Process.*, pp. 3285–3288, 2008.

[24] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Trans. Signal Process.*, Vol. 57, no. 6, pp. 2365–2382, June, 2009.

[25] L. Xiao, S. Boyd and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. Int. Conf. Info. Process. in Sensor Networks*, 2005, pp. 63–70.

[26] L. Xiao, Stephen Boyd, "Fast linear iterations for distributed averaging," *Syst. & Control Lett.*, Vol. 53, no. 1, pp. 65-78, 2004.

[27] R. Arablouei, S. Werner, Y. F. Huang and K. Doğançay, "Distributed least mean-square estimation with partial diffusion," *IEEE Trans. Signal Process.*, Vol. 62, no. 2, pp. 472–484, Jan., 2013.

[28] R. Arablouei,K. Doğançay, S. Werner and Y. F. Huang, "Adaptive distributed estimation based on recursive least-squares and partial diffusion," *IEEE Trans. Signal Process.*, Vol. 62, no. 14, pp. 3510–3522, Jul., 2014.

[29] V. C. Gogineni and M. Chakraborty, "Partial diffusion affine projection algorithm over clustered multitask networks," in *Proc. IEEE Int. Symp. Circuits and Syst.*, 2019, pp. 1-5.

[30] A. H. Sayed, "*Adaptation, learning, and optimization over networks*," Found. Trends Mach. Learn., vol. 7, no. 4–5, pp. 311–801, 2014.

[31] A. H . Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Process.,* vol. 3, pp. 322-453, Elsevier, 2014.