An Adversary-Resilient Doubly-Compressed Diffusion LMS Algorithm for Distributed Estimation

Hadi Zayyani, Fatemeh Oruji, Inbar Fijalkow

Noname manuscript No.
(will be inserted by the editor)

An Adversary-Resilient Doubly-Compressed Diffusion LMS Algorithm for Distributed Estimation

Hadi Zayyani · Fatemeh Oruji · Inbar Fijalkow

Received: date / Accepted: date

Abstract This paper proposes an adversary resilient communication-efficient distributed estimation algorithm for time-varying networks. It is a generalization of the doubly-compressed Diffusion Least Mean Square (DLMS) algorithm that isn't adversary-resilient. The major drawback in existing adversary detectors in the literature is that they suggested the detection criterion heuristically. In this paper, an adversary detector is suggested theoretically based on a Bayesian Hypothesis Test (BHT). It is proved that the test statistics of the detectors is a distance metric compared to a threshold similarly to related papers in the literature. Hence, we prove the validity of the detection criterion based on BHT. The other difficulty encountered in existing works is the determination of thresholds. In this paper, the optimum thresholds are derived in closed-form. Since the optimum thresholds need the values of unknown parameters, it is not feasible to derive them. Hence, suboptimal procedures for determining the thresholds are provided. Moreover, convergence of the mean of the algorithm is investigated analytically. In addition, the Cramer-Rao Bound (CRB) of the problem of distributed estimation based on all nodes observations in the presence of adversaries is calculated. The simulation results show the effectiveness of the proposed algorithms and demonstrate that the proposed algorithms reach the performance of the algorithm when the adversaries are ideally known in advance, with some delay.

Keywords Distributed estimation · adversary resilient · Bayesian hypothesis testing · communication reduction · diffusion LMS

Hadi Zayyani
Faculty of Electrical and Computer Engineering, Qom University of Technology (QUT), Qom, Iran E-mail: zayyani@qut.ac.ir

Fatemeh Oruji
Qom University of Technology (QUT), Qom, Iran

Inbar Fijalkow
ETIS, UMR 8051, CY Cergy Paris University, ENSEA, CNRS, Cergy, France

1 Introduction

Distributed estimation problems have been vastly investigated in recent years [1]-[32], with potential applications in networks such as Wireless Sensor Networks (WSN) [33] and Internet of Things (IoT) networks [29]. In most distributed estimation problems, a collection of physically distributed nodes aim to collaboratively estimate an unknown vector parameter (of an underlying physical phenomenon) from linear measurements observed by all nodes. There are three methods: incremental, consensus and diffusion strategies for distributed estimation where diffusion strategy is reported to have more advantages [1]. Diffusion-based algorithms for distributed estimation are used extensively in the literature [1]-[32] in which the neighboring nodes diffuse their estimates and measurements to adapt and combine their estimates.

In these networks, energy consumption [34], [35] is an important issue. To reduce the energy consumption, distributed estimation algorithms can reduce the amount of data to be communicated as suggested in [6]-[18]. In pioneering works [6], [7], a partial diffusion algorithm is suggested in which a partial part of intermediate estimations are sent to the neighbors. In [8], a random subset of neighboring nodes are omitted to communicate with the corresponding node and they are substituted with the own estimate of the node. Moreover, [9] devised a data-selective diffusion-LMS algorithm to reduce the communication overhead in which a dynamic diffusion method is presented which shares only the dynamic neighborhood information. Besides the selective approaches described above, in [10] and [11], the authors suggested a compressive diffusion strategy to reduce the communication load. In addition, [12] uses motifs which are local structural patterns common in wireless sensor networks. In [13], the probabilities of data fusion from neighboring nodes is controlled by minimizing the Mean-Square-Deviation (MSD) to reduce the communication cost. Also, [14] suggested a new version of sparse diffusion LMS algorithm taking both communications and error cost into account. Furthermore, a data reserved periodic diffusion LMS is presented in [15] which has a low communication cost. Recently, a doubly compressed diffusion LMS was proposed using compression by removing the entries in both adaptation and combination steps [16]. Moreover, a neighbor-partial diffusion LMS was recently derived using a random neighbor node estimation to replace the estimation of random removing nodes [17]. Also, an smart selection of nodes is suggested for communication reduction in which an optimum linear combination of available estimates is used instead of removing node estimates [18].

In wireless and IoT networks, security issues should be considered in designing the signal processing tasks such as distributed estimation. So, designing adversary-resilient distributed estimation algorithms is a necessity [36]. To this end, some distributed estimation algorithms are suggested in the literature [37]-[44]. In [37], a flag raising distributed estimator is proposed that allows the agents under attack to perform accurate parameter estimation and detect the adversarial agents simultaneously. [38] suggests a hybrid algorithm composed of a non-cooperative LMS (nc-LMS) algorithm and a correction-based diffu-

sion LMS. Also, [40] proposes a distributed algorithm based on the Kullback-Leibler divergence to detect false data injection attacks. Moreover, in a recent work [41], a resilient distributed diffusion algorithm is suggested which is robust to any data falsification attack when the number of compromised agents in the local neighborhood of a normal agent is bounded. In addition, in [42], by determining an adaptive threshold, a distributed detection algorithm is proposed based on the correlation between tasks and a safe multi-task diffusion least mean square, SM-DLMS. Not only do the authors reduce communication costs, they also effectively decrease the impact of attacks. Furthermore, a secure diffusion least mean squares (S-DLMS) algorithm is proposed which can be considered as a hybrid system, i.e. a nc-LMS subsystem and a DLMS subsystem [43]. [44] proposed an attack detection mechanism and also a reputation model which is utilized by the agents to estimate the other agents' trustworthiness based on their past interactions. Recent papers are dedicated to the influence of disturbances, modeling errors, various uncertainties in real world systems as well as robust and filtering techniques [45]-[48].

In this paper, an adversary-resilient yet communication reduced distributed estimation algorithm is devised. We modify the doubly-compressed diffusion LMS (DC-DLMS) algorithm in [16] to become robust to adversary agents. This paper is different from our past papers on one bit distributed estimation [49], robust distributed estimation against impulsive noise [49], [30], [28], [25] and communication reducing distributed estimation [18], in that it is our first attempt in secure distributed estimation problem [37]-[44]. The strategy of communication reduction in [18] is different from this paper because [18] uses an smart node selection rather than random node selection strategy is used in this paper. In the proposed Adversary-Resilient DC-DLMS (AR-DC-DLMS) algorithm, an adaptation-detection-combination strategy is derived in which the adversary detection is performed by a Bayesian Hypothesis Test (BHT). The BHT adversary detector compares a test statistics to an optimal threshold. The test statistics is a distance metric and the optimal threshold depends on adversary attack parameters which are not available. So, a suboptimal procedure to select the threshold is proposed. In this paper, two adversary false data injection attack models are considered. The first attack model, injects the false data into the measurements at the nodes. The second attack model injects the false data into the intermediate estimations exchanged between nodes in the network. The adversary detector based on the measurements is performed in a centralized manner since all the measurements are needed for detection. The adversary detector based on intermediate estimations is performed in a decentralized manner and each node detects its own neighboring's adversaries. In addition, a theoretical mean convergence analysis of the proposed AR-DC-DLMS algorithm is provided. Moreover, an approximated closed form CRB formula for the distributed estimation problem in the presence of adversaries is provided. Simulation results show that the proposed algorithm reaches the final error performance of the ideal algorithm when the adversaries are known in advance.

In brief, the main contributions and novelties of the paper are:

– Proposing a BHT for adversary detection in two false data injection attack models,
– Calculating analytically the optimum thresholds of BHT and proposing a suboptimal procedure to determine the thresholds practically,
– Proposing the AR-DC-DLMS algorithm for a distributed estimation resilient to adversaries,
– Calculating the CRB of the distributed estimation problem in presence of adversaries,
– Providing the mean convergence analysis of the proposed algorithm.

The organization of the remaining of the paper is as follows. Section 2 introduces the problem and model used in the paper. Section 3 reviews the basics of diffusion LMS and doubly compressed diffusion LMS algorithm. In section 4, the proposed adversary-resilient distributed estimation algorithm is derived. In section 5, the convergence analysis of the proposed algorithm is derived. Section 6 calculates the closed form CRB for the problem in presence of adversaries. Simulation results are presented in Section 7. Finally, conclusions are drawn in Section 8.

2 Distributed system model and problem formulation

2.1 Distributed estimation problem

Consider a network consisting of N sensors observing a linear scalar measurement of a common $L \times 1$ unknown vector $\mathbf{w}_o$. The measurements are

$$x_{k,i} = \mathbf{u}_{k,i}^T \mathbf{w}_o + n_{k,i}, \tag{1}$$

where $1 \leq k \leq N$ is the index of the sensor, $1 \leq i \leq I$ is the index of time, $\mathbf{u}_{k,i}$ is the $L \times 1$ zero-mean regression vector of sensor k at time index i and is known, and $n_{k,i}$ is the zero-mean measurement noise. The noise and the regression vectors are assumed to be independent of each other. The objective of the distributed estimation is to collaboratively and adaptively estimate the unknown vector $\mathbf{w}_o$ using the known sequences of measurements and regression vectors $\{x_{k,i}, \mathbf{u}_{k,i}\}$ for $1 \leq k \leq N$ and $1 \leq i \leq I$. Each node k can communicate the data with its neighboring nodes collected in the set $\mathcal{N}_k$.

In this paper, we assume that some of the sensors are attacked by adversaries whose attack models are presented in the following.

2.2 Adversary attack models

Although there are some important practical attacks such as denial of service attacks in WSN [50], we only consider the false data injection attacks discussed in [37]-[44]. There are weak and strong false data injection attacks knowing partially or completely the information of the network and the compromised agent, respectively [41]. In [41], the single node attack model and the network

attack model are designed. In this paper, since the focus is not on designing attacks, we use two simple false data injection models for the attacks [37]-[44]. In the first one, an adversary attacker drifts the unknown vector $\mathbf{w}$ by an error $\mathbf{q}_{k,i}$. So, the attack hypothesis (H_1) can be defined as

$$x_{k,i} = \begin{cases} \mathbf{u}_{k,i}^T(\mathbf{w} + \mathbf{q}_{k,i}) + n_{k,i} & \gamma_{lk,i} = 1 \quad (H_1) \\ \mathbf{u}_{k,i}^T\mathbf{w} + n_{k,i} & \gamma_{lk,i} = 0 \quad (H_0) \end{cases} \tag{2}$$

where the attack error $\mathbf{q}$ is independent of the measurements vector, and H_0 is the no-attack hypothesis. In the first model of adversary, we should resort to an adversary detector based on the measurements. In the second attack model, the adversary takes control of the communication link and injects the false data into intermediate estimations $\boldsymbol{\psi}_{l,i}$ by an error equal to $\mathbf{e}_{l,i}$. Therefore, the attack hypothesis is defined as

$$\boldsymbol{\Psi}_{l,i} = \begin{cases} \boldsymbol{\psi}_{l,i} + \mathbf{e}_{l,i} + \mathbf{v}_{l,i} & \gamma_{lk,i} = 1 \quad (\tilde{H}_1), \\ \boldsymbol{\psi}_{l,i} + \mathbf{v}_{l,i} & \gamma_{lk,i} = 0 \quad (\tilde{H}_0), \end{cases} \tag{3}$$

where the intermediate estimation $\boldsymbol{\psi}_{l,i}$ are used as in the DLMS (4) as described in section 3. In Section 4, we derive the adversary detector based on the measurements and the detector based on the intermediate estimations.

3 Diffusion LMS and Doubly-compressed diffusion LMS algorithms

3.1 Diffusion LMS

In the Diffusion LMS (DLMS), a mean square error convex cost function is considered at each node. For the k^{th} node, it is $J_k(\mathbf{w}) = E\{|x_{k,i} - \mathbf{u}_{k,i}^T\mathbf{w}|^2\}$. DLMS seeks the minimization of the aggregate global cost function defined as $J^{\text{glob}}(\mathbf{w}) = \sum_{k=1}^{N} J_k(\mathbf{w})$. In the Adapt-Then-Combine (ATC) strategy of DLMS, this is obtained cooperatively by updating the local estimations in the adaptation step and then combining the local estimations to yield a new estimation. Hence, the overall DLMS algorithm is a two-step algorithm as follows [1]:

$$\begin{cases} \boldsymbol{\psi}_{k,i+1} = \mathbf{w}_{k,i} + \mu_k \sum_{l\in\mathcal{N}_k} c_{lk}\hat{\mathbf{g}}_l(\mathbf{w}_{k,i}), \\ \mathbf{w}_{k,i+1} = \sum_{l\in\mathcal{N}_k} a_{lk}\boldsymbol{\psi}_{l,i+1}, \end{cases} \tag{4}$$

where $\hat{\mathbf{g}}_l(\mathbf{w}_{k,i}) = [x_{l,i} - \mathbf{u}_{l,i}^T\mathbf{w}_{k,i}]\mathbf{u}_{l,i} = -\nabla_{\mathbf{w}}(J_l(\mathbf{w}_{k,i}))$ is the negative of the gradient of $J_l(\mathbf{w})$ with $\mathbf{w}$ replaced by the locally estimated $\mathbf{w}_{k,i}$. $\mathcal{N}_k$ is the neighborhood set of the k'th sensor. $\boldsymbol{\psi}_{k,i+1}$ is the intermediate estimation of the k'th sensor at the next time index. It is calculated through the adaptation step of the algorithm. c_{lk} and a_{lk} are the combination coefficients from node l to node k in the adaptation and combination steps, respectively.

3.2 Doubly-compressed DLMS

DC-DLMS [16] is a variant of the DLMS algorithm where two random diagonal entry-selective matrices $\mathbf{H}_{k,i}$ and $\mathbf{Q}_{k,i}$ are used at node k and time index i. Matrices $\mathbf{H}_{k,i}$ and $\mathbf{Q}_{k,i}$ have M and M_Δ ones on their diagonals. The other diagonal entries are set to zero. The ones select the entries and the zeros remove the entries. The matrix $\mathbf{H}_{k,i}$ selects the entries of the estimation and the matrix $\mathbf{Q}_{k,i}$ selects the entries of the gradient. The overall DC-DLMS algorithm is as follows [16]:

$$\begin{cases} \boldsymbol{\psi}_{k,i+1} = \mathbf{w}_{k,i} + \mu_k \sum_{l\in\mathcal{N}_k} c_{lk}\mathbf{g}_{l,i}, \\ \mathbf{w}_{k,i+1} = a_{kk}\boldsymbol{\psi}_{k,i+1} + \\ \sum_{l\in\mathcal{N}_k^-} a_{lk}\Big[\mathbf{H}_{l,i}\mathbf{w}_{l,i} + (\mathbf{I}-\mathbf{H}_{l,i})\boldsymbol{\psi}_{k,i+1}\Big], \end{cases} \tag{5}$$

where $\mathbf{g}_{l,i}$, the negative of the gradient is defined as [16]:

$$\mathbf{g}_{l,i} = \mathbf{Q}_{l,i}\mathbf{u}_{l,i}\Big[x_{l,i} - \mathbf{u}_{l,i}^T\Big(\mathbf{H}_{k,i}\mathbf{w}_{k,i} + (\mathbf{I}-\mathbf{H}_{l.i})\mathbf{w}_{l,i}\Big)\Big]$$

$$+(\mathbf{I}_L - \mathbf{Q}_{l,i})\mathbf{u}_{k,i}\Big[x_{k,i} - \mathbf{u}_{k,i}^T\mathbf{w}_{k,i}\Big], \tag{6}$$

and $\mathcal{N}_k^-$ stands for the neighborhood set of node k except k itself. The false measurement data (in first false data injection model) and false intermediate estimate (in second false data injection model) deviates the common estimate of the parameter vector in aforementioned DLMS and DC-DLMS algorithms. So, unfortunately, the DLMS and DC-DLMS algorithms are sensitive to attacks of the adversaries and their performances are degraded in the presence of adversaries. Therefore, the main aim of this paper is to devise an adversary resilient yet communication reduced diffusion algorithm. This is accomplished in the next sections.

4 The proposed adversary-resilient doubly-compressed DLMS algorithm

4.1 Basic idea

The main purpose of this paper is to modify the DC-DLMS [16] to be resilient to adversary. In the adaptation and combination steps, we suggest to exclude the nodes attacked by adversaries. So, we propose the AR-DC-DLMS algorithm described as follows:

$$\begin{cases} \boldsymbol{\psi}_{k,i+1} = \mathbf{w}_{k,i} + \mu_k \sum_{l\in\mathcal{N}_k} c_{lk}(1-\gamma_{lk,i})\mathbf{g}_{l,i}, \\ \mathbf{w}_{k,i+1} = a_{kk}\boldsymbol{\psi}_{k,i+1} + \\ \sum_{l\in N_k^-} a_{lk}\Big[(1-\gamma_{lk,i})\boldsymbol{\varphi}_{lk,i} + \gamma_{lk,i}\mathbf{w}_{k,i-1}\Big], \end{cases} \tag{7}$$

where $\gamma_{lk,i} = 0$ or 1 is the adversary l indicator of node k coefficient at instant i and $\boldsymbol{\varphi}_{lk,i}$ is defined as

$$\boldsymbol{\varphi}_{lk,i} = \mathbf{H}_{l,i}\mathbf{w}_{l,i-1} + (\mathbf{I}_L - \mathbf{H}_{l,i})\boldsymbol{\psi}_{k,i}. \tag{8}$$

To design the $\gamma_{lk,i}$, we should detect the adversary nodes. To do this, we suggest to use a Bayesian Hypothesis Test (BHT). To devise the detectors, we need to assume some attack model for the adversaries. Two different attack models can be considered as will be discussed in the next subsections.

4.2 Node adversary detector based on measurements

Next, we should consider the hypothesis testing. BHT selects $\hat{\gamma}_{lk,i} = 1$ if the a posteriori probabilities satisfy $p(H_1|\mathbf{x}_k) \geq p(H_0|\mathbf{x}_k)$ and $\hat{\gamma}_{lk,i} = 0$ otherwise where $\mathbf{x}_k = [x_{k,1}, ..., x_{k,i}]^T$. We have $p(H_1|\mathbf{x}_k) \propto p(H_1)p(\mathbf{x}_k|H1)$ and $P(H_0|\mathbf{x}_k) \propto p(H_0)p(\mathbf{x}_k|H_0)$. If the prior probability is assumed to be $p(H_1) = p_a$, then $p(H_0) = 1 - p_a$. Also, the likelihoods $p(\mathbf{x}_k|H_1)$ and $p(\mathbf{x}_k|H0)$ are equal to

$$\begin{aligned} p(\mathbf{x}_k|H_1) &= p(\mathbf{x}_{i-1,k}|H_1)p(x_{k,i}|H_1, \mathbf{x}_{i-1,k}) \\ p(\mathbf{x}_k|H_0) &= p(\mathbf{x}_{i-1,k}|H_0)p(x_{k,i}|H_0, \mathbf{x}_{i-1,k}) \end{aligned} \tag{9}$$

where we denote $\mathbf{x}_{i-1,k} = [x_{k,1}, ..., x_{k,i-1}]^T$. Assuming that $\mathbf{u}_{k,i-1}$ is independent of $\mathbf{u}_{k,i}$, and $\mathbf{q}_{k,i}$ is independent of $\mathbf{q}_{k,i-1}$, then we have $p(x_{k,i}|H_1, \mathbf{x}_{i-1,k}) = p(x_{k,i}|H_1)$ and $p(x_{k,i}|H_0, \mathbf{x}_{i-1,k}) = p(x_{k,i}|H_0)$. So that, we have $\log(p(\mathbf{x}_k|H_1)) = \sum_{j=1}^{i} \log(p(x_{k,j}|H_1))$. Similarly, we have $\log(p(\mathbf{x}_k|H_0)) = \sum_{j=1}^{i} \log(p(x_{k,j}|H_0))$. The hypothesis in (2) can be written as:

$$x_{k,i} = \begin{cases} \mathbf{u}_{k,i}^T\mathbf{w} + \tilde{n}_{k,i} & \text{If } k \text{ is adversary } (H_1) \\ \mathbf{u}_{k,i}^T\mathbf{w} + n_{k,i} & \text{otherwise } (H_0) \end{cases} \tag{10}$$

where $\tilde{n}_{k,i} = \mathbf{u}_{k,i}^T\mathbf{q}_{k,i} + n_{k,i}$. If the length of the unknown vector $\mathbf{w}$, equal to L, is large, then from the Central Limit Theorem (CLT), the distribution of $\tilde{n}_{k,i}$ is Gaussian with zero mean and variance $\tilde{\sigma}_n^2 = L\sigma_u^2\sigma_q^2 + \sigma_n^2$, in which it is assumed that the elements of $\mathbf{q}_{k,i}$ are identically distributed with zero mean and variance σ_q^2 and are independent of $n_{k,i}$. So, we have

$$p(x_{k,j}|H_1) = \frac{1}{\tilde{\sigma}_n\sqrt{2\pi}} \exp\left(\frac{-1}{2\tilde{\sigma}_n^2}(x_{k,j} - \mathbf{u}_{k,j}^T\mathbf{w})^2\right), \tag{11}$$

and

$$p(x_{k,j}|H_0) = \frac{1}{\sigma_n\sqrt{2\pi}} \exp\left(\frac{-1}{2\sigma_n^2}(x_{k,j} - \mathbf{u}_{k,j}^T\mathbf{w})^2\right). \tag{12}$$

So, we have $\log(p(\mathbf{x}_k|H_1)) = \sum_{j=1}^{i} -\frac{1}{2}\log(2\pi\tilde{\sigma}_n^2) - \frac{1}{\tilde{\sigma}_n^2}(x_{k,j} - \mathbf{u}_{k,j}^T\mathbf{w})^2$ and $\log(p(\mathbf{x}_k|H_0)) = \sum_{j=1}^{i} -\frac{1}{2}\log(2\pi\sigma_n^2) - \frac{1}{\sigma_n^2}(x_{k,j} - \mathbf{u}_{k,j}^T\mathbf{w})^2$. Back to the detector, with some calculations, the BHT detector decides the hypothesis H_1 of

presence of an adversary when we have

$$\sum_{j=1}^{i}(x_{k,j}-\mathbf{u}_{k,j}^{T}\mathbf{w})^{2}\geq \mathrm{Th}_{i}, \tag{13}$$

where the optimum threshold is equal to $\mathrm{Th}_i \triangleq \frac{\sigma_n^2\tilde{\sigma}_n^2}{\tilde{\sigma}_n^2-\sigma_n^2}\Big[log(\frac{1-p_a}{p_a})+i\log(\frac{\tilde{\sigma}_n}{\sigma_n})\Big]$. So, the adversary detector based on BHT needs the prior probability of adversary. This can be circumvented by a suboptimal procedure for determining the threshold which will be discussed later. Replacing $\tilde{\sigma}_n^2 = L\sigma_u^2\sigma_q^2+\sigma_n^2$ in the above formula leads to the following final formula for the threshold which is increasing with the time index i:

$$\mathrm{Th}_i = \sigma_n^2(1+\frac{\sigma_n^2}{L\sigma_q^2\sigma_u^2})\Big[\log(\frac{1-p_a}{p_a})+i\log(\frac{\tilde{\sigma}_n}{\sigma_n})\Big] \tag{14}$$

To simplify the final detector, from (13), we decide on the presence of the adversary if we have

$$||\mathbf{x}_k-\mathbf{U}_k^{T}\mathbf{w}||^2\geq \mathrm{Th}_i \tag{15}$$

where $\mathbf{U}_k=[\mathbf{u}_{k,1}|\mathbf{u}_{k,2}|...|\mathbf{u}_{k,i}]^T$ in which $\mathbf{u}_{k,j}$ is the column of $\mathbf{U}_k$. The disadvantage of the adversary detector in (15) is that the computational complexity grows with the increasing the time index i. So, we can confine our observation vector to $\tilde{\mathbf{x}}_{k,i}=[x_{k,i-R+1},...,x_{k,i}]^T$ which consists of last R terms of the observations. Similar derivations lead to the final adversary detection criterion

$$||\tilde{\mathbf{x}}_k-\tilde{\mathbf{U}}_k^{T}\mathbf{w}||^2\geq \mathrm{Th}_R, \tag{16}$$

where $\tilde{\mathbf{U}}_k=[\mathbf{u}_{k,i-R+1}|...|\mathbf{u}_{k,i}]^T$, the threshold Th_R is equal to:

$$\mathrm{Th}_R = \sigma_n^2(1+\frac{\sigma_n^2}{L\sigma_q^2\sigma_u^2})\Big[\log(\frac{1-p_a}{p_a})+R\log(\frac{\tilde{\sigma}_n}{\sigma_n})\Big] \tag{17}$$

The problem of the detectors in (15) and (16) is that we do not know the unknown vector $\mathbf{w}$ in advance. Therefore, we use the estimated $\hat{\mathbf{w}}$ instead of the $\mathbf{w}$. So, the detection criterion is equivalent to

$$||\tilde{\mathbf{x}}_k-\tilde{\mathbf{U}}_k^{T}\hat{\mathbf{w}}||^2\geq \mathrm{Th}_f, \tag{18}$$

where Th_f is a threshold which should be determined. So, we resort to a suboptimal practical approach for finding the threshold Th_f. Let's set the test statistics as

$$T=||\tilde{\mathbf{x}}_k-\tilde{\mathbf{U}}_k^{T}\hat{\mathbf{w}}||^2. \tag{19}$$

Next, we assume that the above test statistics are distributed as a mixture of Gaussian as below

$$p(T)=(1-p_a)N(\mu_0,\sigma_0^2)+p_aN(\mu_1,\sigma_1^2), \tag{20}$$

where $p(T|H_0) = N(\mu_0, \sigma_0^2)$ and $p(T|H_1) = N(\mu_1, \sigma_1^2)$. This assumption is verified in practice by the simulations. To determine the threshold Th_f, the means μ_0 and μ_1 can be determined by finding the peaks of the pdf $p(T)$. The optimum threshold can be determined by minimizing the detection error probability. The probability of binary error detection is equal to

$$\begin{aligned} p_e &= (1-p_a)p(T > \text{Th}|H_0) + p_a p(T < \text{Th}|H_1) \\ &= (1-p_a)Q(\frac{\text{Th}-\mu_0}{\sigma_0}) + p_a(1-Q(\frac{\text{Th}-\mu_1}{\sigma_1})), \end{aligned} \tag{21}$$

where $Q(.)$ is the Q-function [51]. Taking the derivative of the probability of error with respect to the threshold and enforce it to be equal to zero, leads to the following quadratic equation

$$\frac{(\text{Th}-\mu_0)^2}{2\sigma_0^2} - \frac{(\text{Th}-\mu_1)^2}{2\sigma_1^2} = \log(\frac{1-p_a}{p_a}\frac{\sigma_1}{\sigma_0}). \tag{22}$$

To derive a simple solution, we assume $\sigma_0 = \sigma_1$. Then, the final threshold is obtained as

$$\text{Th}_o = \frac{\mu_1^2 - \mu_0^2 + 2\sigma_0^2 \log(\frac{1-p_a}{p_a})}{2(\mu_1 - \mu_0)}. \tag{23}$$

If we assume that $\mu_1^2 - \mu_0^2 \gg 2\sigma_0^2 \log(\frac{1-p_a}{p_a})$ i.e. the two Gaussian are far apart from each other and that σ_0^2 is small with respect to $\mu_1^2 - \mu_0^2$, a straightforward formula for the threshold is $\text{Th}_o \approx \frac{\mu_0+\mu_1}{2}$. It is equivalent to tell that the two Gaussian distribution should be separated enough with respect to their variances to enable detecting the peaks of the Gaussian mixture as a valid estimate of the mean values. It is shown in the simulation results that this is sufficient for reaching acceptable results. Since we need to compute the pdf of the test statistics for both adversary and non-adversary measurements, we use this detector in the centralized version of the proposed method. The details of the procedure is shown in Algorithm 1 (Centralized AR-DLMS algorithm) in which the adversary detection based on measurements is demonstrated in Algorithm 2.

4.3 Link adversary detector based on intermediate estimations

Following the model in (3), the BHT detector decides on adversary link if we have

$$p(\tilde{H}_1)p(\boldsymbol{\Psi}_l|\tilde{H}_1) \geq p(\tilde{H}_0)p(\boldsymbol{\Psi}_l|\tilde{H}_0), \tag{24}$$

where index i is omitted for the sake of brevity. Denoting $p(\tilde{H}_1) = p_a$, $p(\tilde{H}_0) = 1 - p_a$, and the likelihoods are as

$$p(\boldsymbol{\Psi}_l|\tilde{H}_1) = \frac{1}{(2\pi\sigma_1^2)^{\frac{L}{2}}} \exp(\frac{-1}{2\sigma_1^2}||\boldsymbol{\Psi}_l - \boldsymbol{\psi}_l||^2) = N(\boldsymbol{\psi}_l, \sigma_1^2\mathbf{I}), \tag{25}$$

and

$$p(\boldsymbol{\Psi}_l|\tilde{H}_0) = \frac{1}{(2\pi\sigma_0^2)^{\frac{L}{2}}} \exp(\frac{-1}{2\sigma_0^2}||\boldsymbol{\Psi}_l - \boldsymbol{\psi}_l||^2) = N(\boldsymbol{\psi}_l, \sigma_0^2\mathbf{I}), \tag{26}$$

where the error vector $\mathbf{e}_l$ in (3) is assumed to be a zero-mean Gaussian random variable with variance σ_e^2 and independent from $\mathbf{v}_l$ which is a zero-mean Gaussian noise with variance σ_v^2, $\sigma_0^2 = \sigma_v^2$, and $\sigma_1^2 = \sigma_v^2 + \sigma_e^2$. Hence, with some calculations and simplifications, (24) results in the final link adversary detection criterion as

$$||\boldsymbol{\Psi}_l - \boldsymbol{\psi}_l||^2 \geq \mathrm{Th}_L, \tag{27}$$

where the optimum threshold is equal to

$$\mathrm{Th}_L = 2\sigma_v^2(1 + \frac{\sigma_v^2}{\sigma_e^2})\Big[\log(\frac{1-p_a}{p_a}) + L\log(\frac{\sigma_1}{\sigma_0})\Big]. \tag{28}$$

The issue in the detector of (27) is that we do not know $\boldsymbol{\psi}_l$ in advance and the threshold value depends on parameters which are not known beforehand. Moreover, if the node l is not a link adversary and knows that it is not, then we can resort to $\boldsymbol{\psi}_l = \boldsymbol{\Psi}_l = \boldsymbol{\psi}_0$. But, we can not really know that the link is not captured by the adversary before detection. So, if we wait for a fixed delay of D samples to be sure that we have a rough estimate of the unknown vector, and if the intermediate estimations of adversaries are far apart from the true estimation, we can detect the adversaries by outlier detection. Therefore, if we find the most distant values of the test statistics and detect to which nodes they belong, we can detect the adversaries. The details of the proposed AR-DC-DLMS is shown in Algorithm 3 (Decentralized algorithm) in which the adversary detection based on intermediate estimations is explained in Algorithm 4.

4.4 The proposed Algorithms

The general pseudocode of centralized adversary-resilient distributed estimation algorithm AR-DLMS is shown in Algorithm 1. In this algorithm, the detected adversary nodes are excluded in updating the centralized estimation. In algorithm 1, the adversary detection is done via Algorithm 2. Moreover, the proposed decentralized AR-DC-DLMS algorithm is provided in Algorithm 3. This is a three step adaptation-detection-combination algorithm in which the adversary detection is done in Algorithm 4. The total communication reduction ratio is equal to $\frac{1}{N}\sum_k \frac{(|\mathcal{N}_k| - N_{\max})}{|\mathcal{N}_k|}$ and communication reduction ratio at node k is equal to $\frac{|\mathcal{N}_k| - N_{\max}}{|\mathcal{N}_k|}$. This communication reduction ratio leads to lower energy consumption of the entire network.

5 Convergence analysis

In this section, the convergence of the mean of the algorithm is discussed. The matrices $\mathbf{H}_{l,i}$ and $\mathbf{Q}_{l,i}$ are assumed to be random with M and M_Δ nonzero

Algorithm 1 Centralized adversary-resilient distributed estimation AR-DLMS

Input: **Observations** $x_{l,i}$; **Regression vectors** $\mathbf{u}_{l,i}$.
Parameters: **Step-size** μ_{glob}, **Fixed-delay** D.
Initialize $i = 0, \hat{\mathbf{w}}_i = \mathbf{0}_{L\times 1}$.

repeat
- i=i+1;
- For $k = 1 : N$
- Compute $T_{k,i} = ||\tilde{\mathbf{x}}_k - \tilde{\mathbf{U}}_k^T \hat{\mathbf{w}}_i||$.
- if $i \geq D$
- **Adversary detection based on measurements** based on algorithm 2.
- end if;
- end for;
- Update: $\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_{i-1} + \mu_{\text{glob}} \sum_{l\in \text{Non}-\text{Adv}} \mathbf{u}_{l,i}(d_{l,i} - \mathbf{u}_{l,i}^T \hat{\mathbf{w}}_{i-1})$

until A stopping criterion is reached

Algorithm 2 Adversary detection based on measurements

- Calculate the mixture of Gaussian pdf of T.
- Calculate the means of two Gaussian variables by peak finding (μ_0, μ_1) in the $p(T)$.
- Calculate the threshold $\text{Th}_{o,f} \approx \frac{\mu_0+\mu_1}{2}$.
- Adversary-detector: $T_{k,i} \geq \text{Th}_{o,f}$.

Algorithm 3 Decentralized adversary-resilient doubly-compressed diffusion LMS (AR-DC-DLMS)

Input: **Observations** $x_{l,i}$; **Regression vectors** $\mathbf{u}_{l,i}$.
Parameters: **Step-sizes** μ_k, **Fixed-delay** D, **Parameters** $n_{\max}, N_{\max}$.
Initialize $i = 0, \hat{\mathbf{w}}_{k,i} = \mathbf{0}_{L\times 1}$.

repeat
- i=i+1;
- For $k = 1 : N$
- Generate random $\mathbf{H}_{k,i}$ and $\mathbf{Q}_{k,i}$

  Adaptation:
- Compute $\boldsymbol{\varphi}_{k,i} = \mathbf{w}_{k,i-1} + \mu_k \sum_{l\in N_k} c_{lk}(1 - \hat{\gamma}_{lk,i})\mathbf{g}_{l,i}$ where $\mathbf{g}_{l,i}$ is given by (6).

  Adversary-detection:
- if $i \geq D$
- Determine $\hat{\gamma}_{lk,i}$ based on algorithm 4.

  Combination:
- $\hat{\mathbf{w}}_{k,i} = a_{kk}\boldsymbol{\psi}_{k,i} + \sum_{l\in N_k^-} a_{lk}\Big[(1 - \hat{\gamma}_{lk,i})\boldsymbol{\varphi}_{lk,i} + \hat{\gamma}_{lk,i}\mathbf{w}_{k,i-1}\Big]$ where $\boldsymbol{\varphi}_{lk,i}$ is given by (8)
- end for.

until A stopping criterion is reached

entries on their diagonal. Hence, we have $E\{\mathbf{H}_{l,i}\} = \frac{M}{L}\mathbf{I}_L$ and $E\{\mathbf{Q}_{l,i}\} = \frac{M_\Delta}{L}\mathbf{I}_L$. We shall now analyze the stochastic mean convergence of the AR-DC-DLMS algorithm. We follow the approach of [16] with the difference that we exclude the adversaries in the AR-DC-DLMS algorithm. We shall investigate the effect of this exclusion on the condition of the mean convergence. In this regard, we consider three assumptions, the first and second assumptions being

Algorithm 4 Adversary detection based on intermediate estimations

- Calculate $T_{k,l,i} = ||\boldsymbol{\varphi}_{l,i} - \hat{\mathbf{w}}_{k,i-1}||^2$.
- Calculate $\tilde{T}_{k,l,i} = \text{Sort}\{T_{k,l,i}\}$.
- Detect $n_{\max}$ of greatest $\tilde{T}_{k,l,i}$ (Farthest outliers).
- Detect maximum $N_{\max}$ non-repetitive Adversaries from $n_{\max}$ previous items (Farthest outliers belong to what nodes).
- Determine $\hat{\gamma}_{lk,i}$ based on previous items.

the same as in [16]. The assumptions on the regression data, selection matrices, and the false data injections are as follows:

Assumption 1: The regression vectors $\mathbf{u}_{k,i}$ are independent zero-mean white random processes.

Assumption 2: The matrices $\mathbf{H}_{k,i}$ and $\mathbf{Q}_{l,i}$ are spatially independent white random processes and are independent from each other as well as any other processes.

Assumption 3: The false data injection vectors $\mathbf{q}_{k,i}$ are independent from the regression vectors $\mathbf{u}_{k,i}$, selection matrices $\mathbf{H}_{k,i}$, $\mathbf{Q}_{k,i}$, and noise $n_{k,i}$.

The error vector is defined as $\tilde{\mathbf{w}}_{k,i} = \mathbf{w}_o - \mathbf{w}_{k,i}$. Also, we collect all error vectors across all nodes into the $\tilde{\mathbf{w}}_i = \text{col}\{\tilde{\mathbf{w}}_{1,i}, \tilde{\mathbf{w}}_{2,i}, ..., \tilde{\mathbf{w}}_{N,i}\}$. We show in the Appendix A that a sufficient condition of convergence of the mean of the proposed algorithm is

$$\mu_k < \frac{2}{\lambda_{\max,k}}, \tag{29}$$

where we have:

$$\lambda_{\max,k} = (\frac{MM_\Delta}{L^2} + p_a)\lambda_{\max}(\mathbf{R}_k) + (1 - \frac{M_\Delta}{L})(1 + p_a)\lambda_{\max}(\mathbf{R}_{u_k}) + \frac{M_\Delta}{L}(1 - \frac{M}{L})(1 + p_a)\max_{l \in N_k} c_{lk}\lambda_{\max}(\mathbf{R}_{u_l}), \tag{30}$$

where $\mathbf{R}_k$ and $\mathbf{R}_{u_k}$ are defined in the Appendix A and $\lambda_{\max}(.)$ denotes the maximum eigenvalue of the matrix argument. Obviously, the above maximum eigenvalue is greater than that obtained in [16] for the DC-DLMS algorithm. So, the sufficient convergence condition of μ_k of the AR-DC-DLMS is more stringent than that of the DC-DLMS algorithm. Increasing the percentage of adversaries p_a, the step size parameters μ_k should be selected smaller and the convergence condition is tighter.

6 Cramer-Rao bound for distributed estimation with adversaries

In this section, we calculate the CRB [52] for the problem of distributed estimation with adversaries. All the observations can be modeled as

$$x_{k,i} = \mathbf{u}_{k,i}^T(\mathbf{w} + \tilde{\mathbf{q}}_{k,i}) + n_{k,i} \tag{31}$$

where $\tilde{\mathbf{q}}_{k,i}$ is the false data injection and can be modeled as a Bernoulli-Gaussian variable as follows

$$\tilde{\mathbf{q}}_{k,i} = \begin{cases} \mathbf{q}_{k,i} & \text{with probability } p_a \\ 0 & \text{otherwise,} \end{cases} \tag{32}$$

where the $\mathbf{q}_{k,i}$ are assumed to be white zero-mean random processes[1] with variance σ_q^2. Then, the observation vector of the k'th node which is defined as $\mathbf{x}_k = [x_{k,1}, x_{k,2}, ..., x_{k,I}]^T$, can be written as:

$$\mathbf{x}_k = \mathbf{U}_k \mathbf{w} + \mathbf{z}_k \tag{33}$$

where $\mathbf{U}_k^T = [\mathbf{u}_{k,1}|\mathbf{u}_{k,2}|...|\mathbf{u}_{k,I}]$, and $\mathbf{z}_k = [z_{k,1}, z_{k,2}, ..., z_{k,I}]^T$ with $z_{k,i} = \mathbf{u}_{k,i}^T \tilde{\mathbf{q}}_{k,i} + n_{k,i}$. All the measurements are $\mathbf{X} = [\mathbf{x}_1|\mathbf{x}_2|..., |\mathbf{x}_N]$. The Fisher-Information Matrix (FIM) elements are defined as

$$F_{l,j} = -E\{\frac{\partial^2 \log p(\mathbf{X}|\mathbf{w})}{\partial w_l w_j}\} \tag{34}$$

The closed form formula of the FIM is derived in the Appendix B and is equal to:

$$\mathbf{F} = \sum_{k=1}^{N} \mathbf{U}_k^T \mathbf{P}_k^{-1} \mathbf{U}_k \tag{35}$$

where $\mathbf{P}_k = \text{diag}(\sigma_{z_{k,i}}^2)$ in which $\sigma_{z_{k,i}}^2 = p_a \sigma_q^2 ||\mathbf{u}_{k,i}||^2 + \sigma_{k,n}^2$ is defined in the Appendix B. Then, the CRB is as follows:

$$E\{(w_j - \hat{w}_j)^2\} \geq \text{CRB}_j = \left[\mathbf{F}^{-1}\right]_{jj} \tag{36}$$

To find a closed form formula for the CRB as well, we assume for simplicity that $\sigma_{k,n}^2 = \sigma_n^2$ and assume that L is large. We can deduce that $\sigma_{z_{k,i}}^2 \approx \sigma_z^2 = Lp_a\sigma_q^2\sigma_u^2 + \sigma_n^2$. Then, we have $\mathbf{F} \approx \frac{1}{\sigma_z^2} \sum_{k=1}^{N} \mathbf{U}_k^T \mathbf{U}_k$. The term $\mathbf{U}_k^T \mathbf{U}_k$ can be approximated by $I\sigma_u^2 \mathbf{I}_L$. Finally, the CRB is approximated as

$$E\{(w_j - \hat{w}_j)^2\} \geq \text{CRB}_j \approx \frac{\sigma_z^2}{NI\sigma_u^2} = \frac{Lp_a\sigma_q^2\sigma_u^2 + \sigma_n^2}{NI\sigma_u^2} \tag{37}$$

[1]The false data injection error $q_{k,i}$ for various adversary nodes and for various time indexes will be more covert (if it is biased in one direction, the adversary can be detected from this bias) if they are assumed to be positive and negative equiprobably. So, we assume such distribution for $q_{k,i}$. We could assume other distribution for $q_{k,i}$ and find the CRB under that assumption.

7 Simulation Results

In this section, the performance of the proposed AR-DC-DLMS algorithm for distributed estimation is evaluated with respect to state-of-the-are techniques. For the performance metric, there are mean square deviation (MSD), normalized MSD, or Signal to Noise Ratio (SNR). Because of the versatility of MSD, the performance of the proposed method is evaluated by calculating the MSD defined as

$$\mathrm{MSD}(dB) = 20\log(||\mathbf{w} - \mathbf{w}_o||_2).$$

The number of Monte Carlo simulations are selected as 50 and the results are averaged over these 50 random independent run of the experiment. The programming tool is MATLAB which is run on a laptop with corei7. The reconstruction quality performance of the proposed algorithm is compared with the three approaches mentioned earlier. In all the experiments, $L = 50$. The sensor network size is chosen with $N = 16$ sensors and is shown in Fig 1. The channels between the nodes are assumed to be AWGN with a background noise. Also, the standard deviation of the Gaussian background noise is $\sigma_n = 0.025$. The elements of the 50×1 parameter vector $\mathbf{w}_o$ are selected as Gaussian iid random variables with zero mean and variance equal to 1. Also, the 50×1 regression vectors $\mathbf{u}_{k,i}$ are exactly generated as the unknown parameter vector. At first, we investigate the effect of parameters of the proposed algorithm which are delay D, n_{max}, and N_{max}. Since the effect of step-sizes is well-known in the literature, we exclude to inspect the effect of step-sizes and they are selected as $\mu_{\mathrm{glob}} = 0.005$ and $\mu_k = \mu = 0.07$ in all experiments hereafter. The explanation for the adversaries and details of false data injection attacks are the same as the following first experiment that will be presented in sequel. Figure 2 shows the effect of the delay parameter D on the MSD curve versus iteration. Figure 3 demonstrates the effect of n_{max}, and Figure 4 depicts the effect of parameter N_{max}. The mentioned figures show that the sensitivity of the performance of the proposed algorithm with respect to D and n_{max} is low. So, we select the values of $D = 100$ and $n_{max} = 10$ in all experiments. Also, Figure 4 shows that the best value for parameter N_{max} is equal to 2. Hence, we choose $N_{max} = 2$ in the following experiments. After adjusting the value of parameters, three experiments are performed and discussed next.

In the first and second experiments, the performance of the proposed centralized AR-DLMS and decentralized AR-DC-DLMS algorithm are compared with the centralized DLMS without adversary detection, the centralized DLMS with ideally known adversaries, the AR-DC-DLMS with known adversaries, DC-DLMS algorithm and Cramer-Rao bound. In the first experiment, we use only the first type of attack for the adversaries. The elements of the false data $\mathbf{q}_{k,i}$ are assumed to be iid Gaussian random variables with zero mean and variance $\sigma_q^2 = 0.36$. The parameters of AR-DC-DLMS are selected as $D = 100$, $N_{\max} = 2$, and $n_{\max} = 10$. To fairly compare the proposed method to other communication reducing algorithms, we use the same communication reduction ratio at each node for all algorithms. We assume to have 4 attacked nodes among the 16 nodes in the network. The adversaries are assumed to be

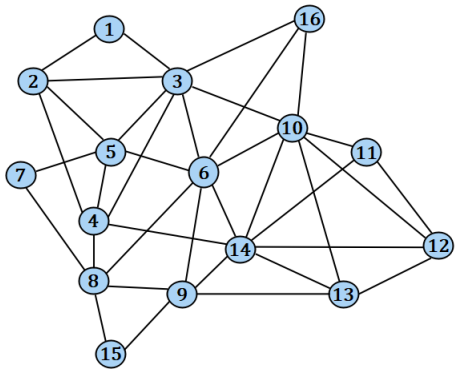

Fig. 1 The network used in the simulations.

fixed and are the 2nd, 5th, 10th, and 12th node in the network. The MSD is displayed in Fig. 5 versus the iteration number. Seven curves are depicted in the mentioned figure. The first is the classic DC-DLMS algorithm which have not any adversary detector. The second is the DC-DLMS with ideal adversary detector which means that we know the adversaries beforehand and exclude them from the algorithm. The third is the proposed AR-DC-DLMS algorithm. The fourth is the centralized algorithm without adversary detection. The fifth is the centralized with ideal adversary detection which means the adversaries are known in advance. The sixth is the centralized AR-DC-DLMS algorithm. The last curve is the CRB which is calculated in Section 6. It shows that the final MSD of the centralized AR-DLMS with adversary detection reaches the final MSD of the centralized AR-DLMS with ideal adversary detection but with a delay due to detection processing. Similarly, the final MSD of the decentralized AR-DC-DLMS reaches the final MSD of the AR-DC-DLMS with ideal adversary detection. Also, the proposed AR-DC-DLMS algorithm is 15dB better than the DC-DLMS algorithm.
In the second experiment, we use a similar experiment to the first experiment. But, we use both first and second attack models simultaneously. In this second experiment, the adversaries not only inject false data into the measurements but also injects false data into the intermediate estimations. For the elements of the false data vector $\mathbf{e}_{l,i}$, we use an iid Gaussian random variable with zero mean and variance $\sigma_e^2 = 0.36$.

For the second experiment, the MSD versus iteration number is depicted in Fig. 6. The curves are similar to the previous figure. It can be seen that the results are somehow similar to these of the first experiment. However, the convergence of the proposed AR-DC-DLMS is slower than in the first experiment and the final MSD of the DC-DLMS is higher than before. The second experiment shows that the proposed AR-DC-DLMS algorithm works well for both attack models when the attackers take control of communication link in addition to taking control of the measurements.

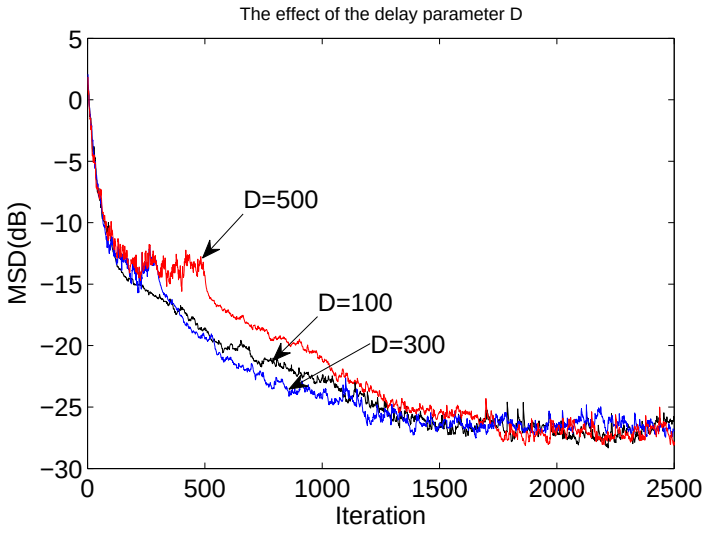

Fig. 2 MSD versus iteration number for the proposed algorithm for different values of delay parameter D. The other parameters are selected as $n_{max} = 10$ and $N_{max} = 2$.

In the next experiment, we use only the first attack model.
In the third experiment, we compare the AR-DC-DLMS algorithm with some state-of-the-art algorithms. The other algorithms are DLMS [1]-[4], Correction-based DLMS (CDLMS) [38], resilient distributed diffusion under F-local bounds (R-DLMSAW) [41], DC-DLMS [16] and Reduced Communication DLMS (RC-DLMS) [8]. The MSD is displayed in Fig. 7 versus the iterations' number. It shows that the proposed AR-DC-DLMS outperforms the other algorithms with respect to final MSD. However, the lower final MSD is obtained with a slower rate of convergence. Table 1 shows the final MSD of various algorithms. It demonstrates that the proposed AR-DC-DLMS has at least a 5dB lower final MSD with respect to others.

8 Conclusion and future works

In this paper, an adversary-resilient doubly-compressed diffusion LMS, called AR-DC-DLMS, is proposed for the distributed estimation problem. The resiliency to adversaries is obtained by detecting adversaries using a BHT. We proved that the test statistics is a distance metric which should be compared to a threshold which is obtained in closed form. Since its calculation requires the adversary statistics that are not available beforehand, a suboptimal procedure for calculating the threshold is proposed. Based on the attack model, the adversary detection is performed using measurements or using intermediate estimations. Direct measurements are used in the centralized version of the proposed algorithm, the AR-DLMS, while intermediate estimations are

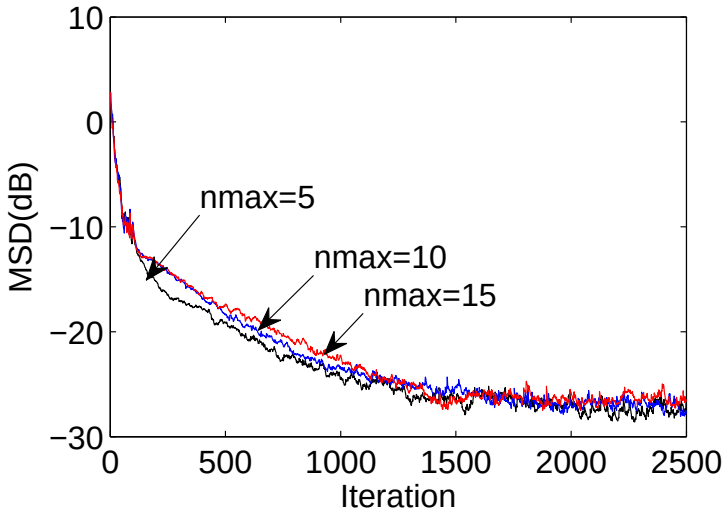

Fig. 3 MSD versus iteration number for the proposed algorithm for different values of parameter n_{max}. The other parameters are $D = 100$ and $N_{max} = 2$.

used in the decentralized version of the algorithm, the AR-DC-DLMS. The proposed algorithm is a three-step adaptation-detection-combination strategy in which the detected adversaries are excluded in the adaptation and combination steps. Moreover, the mean convergence analysis of the AR-DC-DLMS is provided in the paper. We also derived the CRB for the distributed estimation problem in presence of adversaries. The simulation results show the effectiveness of the proposed algorithms in comparison to the ideal case where the adversaries are known in advance and also in comparison to some state-of-the-art algorithms. The main limitation of the proposed method which limits the use of this algorithm in practical scenarios is the usage of simple false data injection models. For example, an adversary may inject false data for a short duration and then stop the false data injection. So, more sophisticated false data injection models (e.g. non-stationary) should be used in practical situations. It will be addressed in future works.

Appendix A The convergence of the mean condition

For calculating the sufficient condition of mean convergence of the weight vectors, we shall define some notations. We define $\tilde{\boldsymbol{\psi}}_{k,i} = \mathbf{w}_o - \boldsymbol{\psi}_{k,i}$. Then, we collect them in a vector as $\tilde{\boldsymbol{\psi}}_i = \text{col}\{\tilde{\boldsymbol{\psi}}_{1,i}, \tilde{\boldsymbol{\psi}}_{2,i}, ..., \tilde{\boldsymbol{\psi}}_{N,i}\}$. Let $\mathbf{R}_{\mathbf{u}_l,i} = \mathbf{u}_{l,i}\mathbf{u}_{l.i}^T$. The other notations are defined as follows:

$$\mathcal{M} = \text{diag}\{\mu_1\mathbf{I}_L, \mu_2\mathbf{I}_L, ..., \mu_N\mathbf{I}_L\} \tag{38}$$

$$\mathcal{R}_{u,i} = \text{diag}\{\mathcal{R}_{u_1,i}, ..., \mathcal{R}_{u_N,i}\} \tag{39}$$

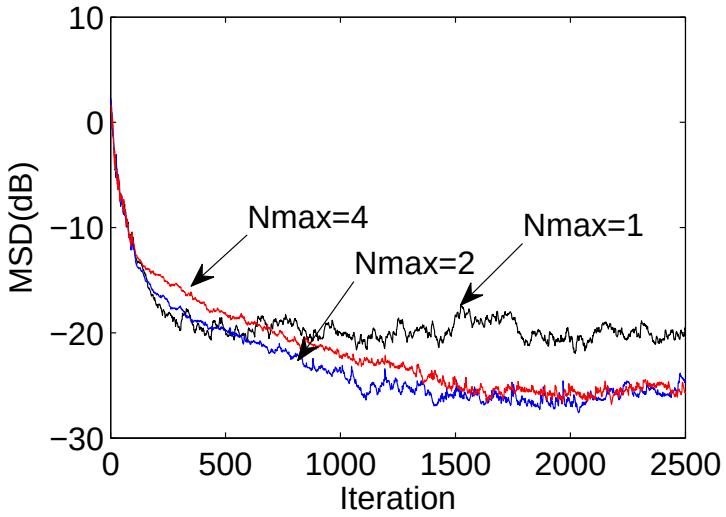

Fig. 4 MSD versus iteration number for the proposed algorithm for different values of N_{max}. The other parameters are $D = 100$ and $n_{max} = 10$

$$\mathcal{C} = \mathbf{C} \otimes \mathbf{I}_L \tag{40}$$

$$\mathcal{R}_{Q,i} = \mathrm{diag}\Big\{ \sum_{l\in\mathcal{N}_1} c_{l1}\mathbf{Q}_{l,i}\mathbf{R}_{\mathbf{u}_l,i}, ..., \sum_{l\in\mathcal{N}_N} c_{lN}\mathbf{Q}_{l,i}\mathbf{R}_{\mathbf{u}_l,i} \Big\} \tag{41}$$

$$\mathcal{R}_{\gamma Q,i} = \mathrm{diag}\Big\{ \sum_{l\in\mathcal{N}_1} c_{l1}\gamma_{l1}\mathbf{Q}_{l,i}\mathbf{R}_{\mathbf{u}_l,i}, ..., \sum_{l\in\mathcal{N}_N} c_{lN}\gamma_{lN}\mathbf{Q}_{l,i}\mathbf{R}_{\mathbf{u}_l,i} \Big\} \tag{42}$$

$$\mathcal{H}_i = \mathrm{diag}\{\mathbf{H}_{1,i}, \mathbf{H}_{2,i}, ..., \mathbf{H}_{N,i}\} \tag{43}$$

$$\mathcal{Q}_i^{'} = \mathrm{diag}\Big\{ \sum_{l\in\mathcal{N}_1} c_{l1}(\mathbf{I}_L - \mathbf{Q}_{l,i}), ..., \sum_{l\in\mathcal{N}_N} c_{lN}(\mathbf{I}_L - \mathbf{Q}_{l,i}) \Big\} \tag{44}$$

$$\mathcal{Q}_{\gamma,i}^{'} = \mathrm{diag}\Big\{ \sum_{l\in\mathcal{N}_1} c_{l1}\gamma_{l1}(\mathbf{I}_L - \mathbf{Q}_{l,i}), ..., \sum_{l\in\mathcal{N}_N} c_{lN}\gamma_{lN}(\mathbf{I}_L - \mathbf{Q}_{l,i}) \Big\} \tag{45}$$

$$\mathcal{R}_{u,i} = \mathrm{diag}\{\mathbf{R}_{\mathbf{u}_1,i}, \mathbf{R}_{\mathbf{u}_2,i}, ..., \mathbf{R}_{\mathbf{u}_N,i}\} \tag{46}$$

$$\mathcal{Q}_i = \mathrm{diag}\{\mathbf{Q}_{1,i}, \mathbf{Q}_{2,i}, ..., \mathbf{Q}_{N,i}\} \tag{47}$$

$$\mathcal{F} = \mathrm{diag}\Big\{ \sum_{l\in\mathcal{N}_1} a_{l1}\gamma_{l_1,i}(\mathbf{I}_L - \mathbf{H}_{l,i}), ..., \sum_{l\in\mathcal{N}_N} a_{lN}\gamma_{l_N,i}(\mathbf{I}_L - \mathbf{H}_{l,i}) \Big\} \tag{48}$$

$$\mathcal{F}^{'} = \mathrm{diag}\Big\{ \sum_{l\in\mathcal{N}_1} a_{l1}\gamma_{l_1,i}\mathbf{H}_{l,i}, ..., \sum_{l\in\mathcal{N}_N} a_{lN}\gamma_{l_N,i}\mathbf{H}_{l,i} \Big\} \tag{49}$$

$$\Big[\mathcal{R}_{Q(I-H),i}\Big]_{kl} = c_{lk}\mathbf{Q}_{l,i}\mathbf{R}_{u_l,i}(\mathbf{I}_L - \mathbf{H}_{k,i}) \tag{50}$$

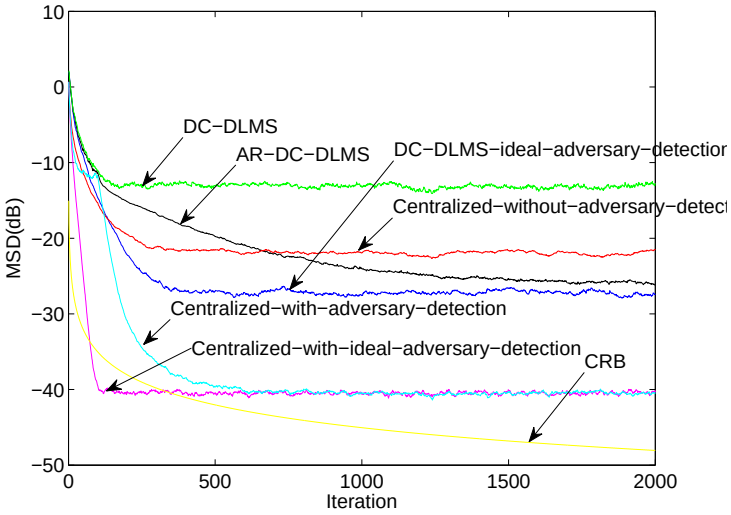

Fig. 5 MSD versus iteration number for centralized and decentralized algorithms with 4 adversaries. Only the first attack model is used.

$$\Big[\mathcal{R}_{\gamma Q(I-H),i}\Big]_{kl} = c_{lk}\gamma_{lk,i}\mathbf{Q}_{l,i}\mathbf{R}_{u_l,i}(\mathbf{I}_L - \mathbf{H}_{k,i}) \tag{51}$$

$$\mathcal{D} = \mathbf{D}_i \otimes \mathbf{I}_L, \quad d_{kl,i} = \sum_{l\in\mathcal{N}} a_{lk}(1-\gamma_{lk,i}) \tag{52}$$

$$\mathcal{R}_{qQ,i} = \mathrm{diag}\Big\{ \sum_{l\in\mathcal{N}_1} c_{l1}\mathbf{Q}_{l,i}\mathbf{R}_{\mathbf{u}_l,i}\mathbf{q}_{l,i}, ..., \sum_{l\in\mathcal{N}_N} c_{lN}\mathbf{Q}_{l,i}\mathbf{R}_{\mathbf{u}_l,i}\mathbf{q}_{l,i} \Big\} \tag{53}$$

$$\mathcal{R}_{\gamma qQ,i} = \mathrm{diag}\Big\{ \sum_{l\in\mathcal{N}_1} c_{l1}\gamma_{l1}\mathbf{Q}_{l,i}\mathbf{R}_{\mathbf{u}_l,i}\mathbf{q}_{l,i}, ..., \sum_{l\in\mathcal{N}_N} c_{lN}\gamma_{lN}\mathbf{Q}_{l,i}\mathbf{R}_{\mathbf{u}_l,i}\mathbf{q}_{l,i} \Big\} \tag{54}$$

$$\mathcal{A}_{q,i} = \mathrm{diag}\Big\{ \sum_{l\in\mathcal{N}_1} c_{l1}(\mathbf{I}_L - \mathbf{Q}_{l,i})\mathbf{R}_{\mathbf{u}_l,i}\mathbf{q}_{l,i}, ..., \sum_{l\in\mathcal{N}_N} c_{lN}(\mathbf{I}_L - \mathbf{Q}_{l,i})\mathbf{R}_{\mathbf{u}_l,i}\mathbf{q}_{l,i} \Big\} \tag{55}$$

$$\mathcal{A}_{\gamma q,i} = \mathrm{diag}\Big\{ \sum_{l\in\mathcal{N}_1} c_{l1}\gamma_{l1}(\mathbf{I}_L - \mathbf{Q}_{l,i})\mathbf{R}_{\mathbf{u}_l,i}\mathbf{q}_{l,i}, ..., \sum_{l\in\mathcal{N}_N} c_{lN}\gamma_{lN}(\mathbf{I}_L - \mathbf{Q}_{l,i})\mathbf{R}_{\mathbf{u}_l,i}\mathbf{q}_{l,i} \Big\} \tag{56}$$

$$\mathcal{S}_i = \mathrm{col}\{\mathbf{u}_{1,i}n_{1,i}, ..., \mathbf{u}_{N,i}n_{N,i}\} \tag{57}$$

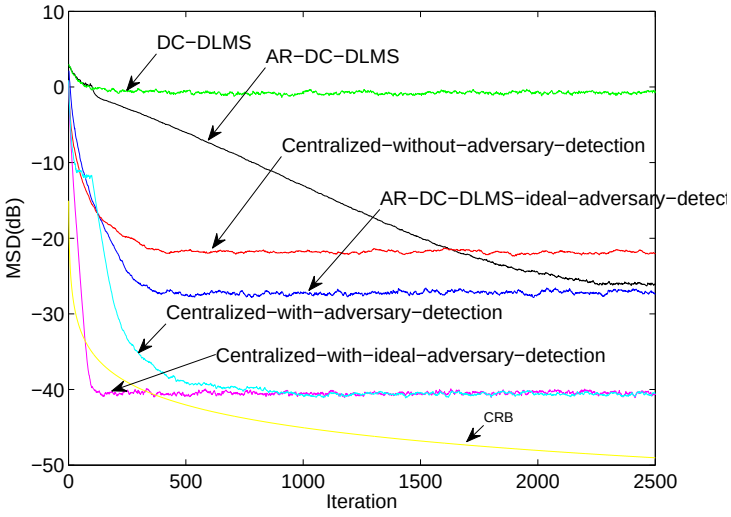

Fig. 6 MSD versus iteration number for centralized and decentralized algorithms. Both false data injection attack models are used.

After the above definitions, some manipulations show that we have:

$$\tilde{\mathbf{w}}_i = (\mathbf{I}_{NL} + \mathcal{F})\tilde{\psi}_i + \mathcal{F}^{'}\tilde{\mathbf{w}}_{i-1} + \mathcal{D}\tilde{\mathbf{w}}_{i-1} \tag{58}$$

For investigating the mean convergence, we take expectation from the above equation. So, we reach:

$$E\{\tilde{\mathbf{w}}_i\} = E\{(\mathbf{I}_{NL} + \mathcal{F})\}E\{\tilde{\psi}_i\} + (E\{\mathcal{F}^{'}\} + E\{\mathcal{D}\})E\{\tilde{\mathbf{w}}_{i-1}\} \tag{59}$$

In the above formula, the term of $E\{\tilde{\psi}_i\}$ is difficult to compute. It needs to calculate a recursion formula for $\tilde{\mathbf{w}}_i$. It is done and for brevity we omit the details of the derivation. We have:

$$\begin{aligned}\tilde{\psi}_i = \Big(&\mathbf{I}_{NL} - \mathcal{M}\mathcal{R}_{Q,i}\mathcal{H}_i - \mathcal{M}\mathcal{Q}_i^{'}\mathcal{R}_{u,i} - \mathcal{M}\mathcal{R}_{Q(I-H),i} \\ &-\mathcal{M}\mathcal{R}_{\gamma Q,i}\mathcal{H}_i - \mathcal{M}\mathcal{Q}_{\gamma,i}^{'}\mathcal{R}_{u,i} - \mathcal{M}\mathcal{R}_{\gamma Q(I-H),i}\Big)\tilde{\mathbf{w}}_{i-1} \\ &-\Big(\mathcal{M}\mathcal{C}^T\mathcal{Q}_i + \mathcal{M}\mathcal{Q}_i^{'} + \mathcal{M}\mathcal{C}^T\mathcal{Q}_{\gamma,i} + \mathcal{M}\mathcal{Q}_{\gamma,i}^{'}\Big)\mathcal{S}_i \\ &-\mathcal{M}\mathcal{R}_{qQ,i} - \mathcal{M}\mathcal{R}_{\gamma qQ,i} - \mathcal{M}\mathcal{A}_{q,i} - \mathcal{M}\mathcal{A}_{\gamma q,i}\end{aligned} \tag{60}$$

Since we assume that $E\{\mathbf{q}_{k,i}\} = 0$ and expectation of noise vectors are zero, the expectations of the above terms in the forth row of (60) are zero. Then, some calculations lead to the following formula:

$$E\{\tilde{\psi}_i\} = \Big(\mathbf{I}_{NL} - \frac{MM_\Delta}{L^2}\mathcal{M}\mathcal{R} - (1 - \frac{M_\Delta}{L})\mathcal{M}\mathcal{R}_u$$

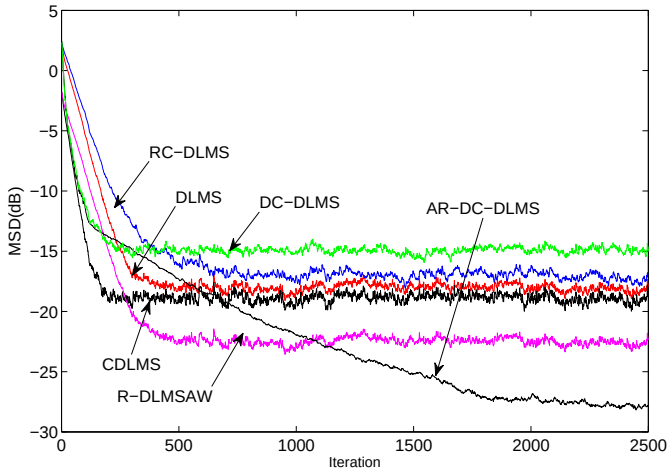

Fig. 7 MSD versus iteration number for decentralized algorithms of AR-DC-DLMS, DC-DLMS, R-DLMSAW, DLMS, CDLMS, and CR-DLMS.

Table 1 Final MSD of various diffusion algorithms

Algorithm	DLMS	CR-DLMS	R-DLMSAW	AR-DC-DLMS	DC-DLMS	CDLMS
MSD (dB)	-18	-17.3	-22.6	-27.7	-15	-19

$$-\frac{M_\Delta}{L}(1-\frac{M}{L})\mathcal{M}\mathcal{C}^T\mathcal{R}_u - p_a\frac{MM_\Delta}{L^2}\mathcal{M}\mathcal{R} - p_a(1-\frac{M_\Delta}{L})\mathcal{M}\mathcal{R}_u$$
$$-p_a\frac{M_\Delta}{L}(1-\frac{M}{L})\mathcal{M}\mathcal{C}^T\mathcal{R}_u\Big)E\{\tilde{\mathbf{w}}_{i-1}\} = \mathcal{B}E\{\tilde{\mathbf{w}}_{i-1}\}, \tag{61}$$

where

$$\mathcal{R}_u = E\{\mathcal{R}_{u,i}\} = \text{diag}\{\mathcal{R}_{u_1}, ..., \mathcal{R}_{u_N}\} \tag{62}$$

$$\mathcal{R} = \text{diag}\{\mathcal{R}_1, ..., \mathcal{R}_N\} \tag{63}$$

with

$$\mathcal{R}_k = \sum_{l\in\mathcal{N}_k} c_{lk}\mathcal{R}_{u_l}. \tag{64}$$

Then, replacing (61) into (59), and with some calculations, we reach to

$$E\{\tilde{\mathbf{w}}_i\} = \Big(1 + p_a(1-\frac{M}{L})\Big)\mathcal{B}E\{\tilde{\mathbf{w}}_{i-1}\}$$
$$+\Big(p_a\frac{M}{L} + (1-p_a)\Big)E\{\tilde{\mathbf{w}}_{i-1}\} = (\mathcal{B}^{'} + \mathcal{Y})E\{\tilde{\mathbf{w}}_{i-1}\}, \tag{65}$$

where $\mathcal{B}^{'} = (1 + p_a(1 - \frac{M}{L}))\mathcal{B}$ and $\mathcal{Y} = [p_a\frac{M}{L} + (1 - p_a)]\mathbf{I}_N$. Then, (65) can be written in the following recursive form

$$E\{\tilde{\mathbf{w}}_i\} = (\mathcal{B}^{'} + \mathcal{Y})E\{\tilde{\mathbf{w}}_{i-1}\} \tag{66}$$

Therefore, similar to [16], the proposed AR-DC-DLMS asymptotically converges in the mean toward $\mathbf{w}_o$ if, and only if $\rho(\mathcal{B}^{'} + \mathcal{Y}) < 1$ where $\rho(.)$ stands for the spectral radius of the matrix argument. From matrix algebra, we have $\rho(\mathbf{X}) \leq ||\mathbf{X}||$ for any induced norm. So, we have:

$$\rho(\mathcal{B}^{'} + \mathcal{Y}) \leq ||\mathcal{B}^{'} + \mathcal{Y}||_{b,\infty} \leq \max||\Big[\mathcal{B}^{'} + \mathcal{Y}\Big]_{kl}|| \tag{67}$$

where $||.||_{b,\infty}$ is the block maximum norm. Deducing from (67), we will have:

$$\begin{aligned}\rho(\mathcal{B}^{'} + \mathcal{Y}) \leq \max_{k,l}||\mathbf{I}_L - \mu_k\Big[\frac{MM_\Delta}{L^2} + (1 - \frac{M_\Delta}{L})\mathcal{R}_{u_k} \\ + \frac{M_\Delta}{L}(1 - \frac{M}{L})c_{lk}\mathcal{R}_{u_l} - p_a[\frac{MM_\Delta}{L^2}\mathcal{R}_k + (1 - \frac{M_\Delta}{L})\mathcal{R}_{u_k} \\ + \frac{M_\Delta}{L}(1 - \frac{M}{L})c_{lk}\mathcal{R}_{u_l}] + p_a\frac{M}{L} + (1 - p_a)\Big]|| < 1\end{aligned} \tag{68}$$

Similar to [16], as a linear combination with positive coefficients of positive definite matrices $\mathcal{R}_k$, $\mathcal{R}_{u_k}$, and $\mathcal{R}_{u_l}$, the matrix in square brackets on the RHS of (68) is positive definite. Then, the condition in right side of (68) holds if (29) is satisfied. Then, the $\lambda_{\max,k}$ is given by (30) and the proof is completed.

Appendix B Calculating the Fisher-information matrix

For calculating the FIM, the total likelihood can be computed as

$$p(\mathbf{X}|\mathbf{w}) = \prod_{k=1}^{N} p(\mathbf{x}_k|\mathbf{w}) = \prod_{k=1}^{N} p_{\mathbf{z}_k}(\mathbf{x}_k - \mathbf{U}_k\mathbf{w}). \tag{69}$$

Since $\mathbf{z}_k = [z_{k,1}, z_{k,2}, ..., z_{k,T}]^T$ with $z_{k,i} = \mathbf{u}_{k,i}^T\tilde{\mathbf{q}}_{k,i} + n_{k,i}$, the $z_{k,i}$ is Gaussian with zero mean and variance $\sigma^2_{z_{k,i}} = E(z^2_{k,i})$. Since $n_{k,i}$ and $\tilde{\mathbf{q}}_{k,i}$ are assumed to be independent and uncorrelated with mean zero, we have $\sigma^2_{z_{k,i}} = E\{(\mathbf{u}_{k,i}^T\tilde{\mathbf{q}}_{k,i})^2\} + E\{n^2_{k,i}\}$. This would be equal to $\sigma^2_{z_{k,i}} = E\{\mathbf{u}_{k,i}^T\tilde{\mathbf{q}}_{k,i}\tilde{\mathbf{q}}_{k,i}^T\mathbf{u}_{k,i}\} + \sigma^2_{k,n}$ where $\sigma^2_{k,n}$ is the variance of noise at node k. Some simple calculations lead to the following formula

$$\sigma^2_{z_{k,i}} = \mathbf{u}_{k,i}^T E\{\tilde{\mathbf{q}}_{k,i}\tilde{\mathbf{q}}_{k,i}^T\}\mathbf{u}_{k,i} + \sigma^2_{k,n} = p_a\sigma_q^2||\mathbf{u}_{k,i}||^2 + \sigma^2_{k,n}, \tag{70}$$

where it is assumed the elements of $\tilde{\mathbf{q}}_{k,i}$ are uncorrelated. Since the elements of $\mathbf{z}_k$ is independent of each other, the vector $\mathbf{z}_k$ is Gaussian with zero mean and diagonal covariance matrix equal to $\mathbf{P}_k = \text{diag}(\sigma^2_{z_{k,i}})$. So, from (69), we can write the log-likelihood as $\log p(\mathbf{X}|\mathbf{w}) = \sum_{k=1}^{N}\Big[-\frac{I}{2}\log(2\pi) -$

$\frac{1}{2}\sum_i \log(\sigma^2_{z_{k,i}}) - \frac{1}{2}\Big(\mathbf{x}_k - \mathbf{U}_k\mathbf{w}\Big)^T \mathbf{P}_k^{-1}\Big(\mathbf{x}_k - \mathbf{U}_k\mathbf{w}\Big)\Big]$. So, the partial derivative will be equal to $\frac{\partial \log p(\mathbf{X}|\mathbf{w})}{\partial w_j} = -\frac{1}{2}\sum_{k=1}^{N} \frac{\partial}{\partial w_j}\Big[\mathbf{e}_k^T \mathbf{P}_k^{-1}\mathbf{e}_k\Big]$ where $\mathbf{e}_k = \mathbf{x}_k - \mathbf{U}_k\mathbf{w}$. We can write $B = \mathbf{e}_k^T\mathbf{P}_k^{-1}\mathbf{e}_k = \sum_{i=1}^{I} P_{k,ii}^{-1} e_{k,i}^2$. So, the partial derivative is equal to $\frac{\partial B}{\partial w_j} = \sum_{i=1}^{I} 2P_{ii}^{-1} e_{k,i}\frac{\partial e_{k,i}}{\partial w_j}$. Also, we have $\frac{\partial e_{k,i}}{\partial w_j} = -U_{k,i,l}$. Taking the second partial derivative and doing some simple manipulations, we reach $F_{l,j} = -E\{\frac{\partial^2 \log p(\mathbf{X}|\mathbf{w})}{\partial w_l w_j}\} = -\sum_{k=1}^{N}\sum_{i=1}^{T} P_{k,ii}^{-1} U_{k,i,j} U_{k,i,l}$, where the this formula leads to (35).

Data Availability Statement

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Acknowledgements For this study, the work of I. Fijalkow has been partially supported by the ELIOT ANR-18-CE40-0030 and FAPESP 2018/12579-7 project.

References

1. A. H. Sayed, *Adaptation, Learning and Optimization over networks*, Foundations and Trends in Machine Learning, 2014.
2. S. Modalavalasa, U. K. Sahoo, A. K. Sahoo, and S. Baraha, "A review of robust distributed estimation strategies over wireless sensor networks," *Elsevier Signal Processing*, vol. 188, Nov 2021.
3. C. G. Lopes, and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. on Signal Proc.*, vol. 56, pp. 3122–3136, 2008.
4. F. S. Cattivelli, and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. on Signal Proc.*, vol. 58, pp. 1035–1048, 2010.
5. F. Hua, R. Nassif, C. Richard, H. Wang, and A. H. Sayed, "Diffusion LMS With Communication Delays: Stability and Performance Analysis," *IEEE Signal Processing Letters*, vol. 27, April 2020.
6. R. Arablouei, S. Werner, Y. Huang, and K. Dogancay, "Distributed Least Mean-Square Estimation With Partial Diffusion," *IEEE Tran. Signal Processing*, vol. 62, no. 2, pp. 472–484, 2014.
7. R. Arablouei, K. Dogancay, S. Werner, and Y. Huang, "Adaptive Distributed Estimation Based on Recursive Least-Squares and Partial Diffusion," *IEEE Tran. Signal Processing*, vol. 62, no. 14, pp. 3510–3522, 2014.
8. R. Arablouei, S. Werner, K. Dogancay, and Y. Huang, "Analysis of a reduced-communication diffusion LMS algorithm," *Elsevier Signal Processing*, vol. 117, pp. 355–361, 2015.
9. J. W. Lee, S. E. Kim, and W. J. Song, "Data-selective diffusion LMS for reducing communication overhead," *Elsevier Signal Processing*, vol. 113, pp. 211–217, 2015.
10. M. O. Sayin, and S. S. Kozat, "Compressive Diffusion Strategies Over Distributed Networks for Reduced Communication Load," *IEEE Tran. Signal Processing*, vol. 62, no. 20, pp. 5308–5323, 2014.
11. M. O. Sayin, and S. S. Kozat, "Single Bit and Reduced Dimension Diffusion Strategies Over Distributed Networks," *IEEE Signal Processing Letters*, vol. 20, no. 10, pp. 976–979, 2013.

12. F. Chen, and X. Shao, "Broken-motifs diffusion LMS algorithm for reducing communication load," *Elsevier Signal Processing*, vol. 133, pp. 213–218, 2017.
13. W. Huang, X. Yang, and G. Shen, "Communication-reducing diffusion LMS algorithm over multitask networks," *Information Sciences (Elsevier)*, vol. 382-383, pp. 115–134, 2017.
14. H. Shiri, M. A. Tinati, M. Coudreanu, and G. Azarnia, "Distributed sparse diffusion estimation with reduced communication cost," *IET Signal Processing*, vol. 12, no. 8, pp. 1043–1052, 2018.
15. J. W. Lee, J. T. Kong, W. J. Song, and S. E. Kim, "Data-Reserved Periodic Diffusion LMS With Low Communication Cost Over Networks," *IEEE Access*, vol. 6, pp. 54636–54650, 2018.
16. E. Harrane, R. Flamary, and C. Richard, "On reducing the communication cost of the diffusion LMS algorithm," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 1, pp. 100–112, March 2019.
17. F. Chen, S. Deng, Y. Hua, S. Duan, L. Wang, and J. Wu, "Communication-Reducing Algorithm of Distributed Least Mean Square Algorithm with Neighbor-Partial Diffusion," *Circuit, System, and Signal Processing*, vol. 39, pp. 4416–4435, 2020.
18. H. Zayyani, "Communication Reducing Diffusion LMS Robust to Impulsive Noise Using Smart Selection of Communication Nodes," *Circuit, System, and Signal Processing*, 2021.
19. J. Ni, J. Chen, and X. Chen, "Diffusion sign-error LMS algorithm: Formulation and stochastic behavior analysis," *Elsevier Signal Processing*, vol. 128, pp. 142–149, Nov 2016.
20. S. Ashkezari-Toussi, and H. Sadoghi-Yazdi, "Robust diffusion LMS over adaptive networks," *Elsevier Signal Processing*, vol. 158, pp. 201–209, May 2019.
21. W. Ma, B. Chen, J. Duan, and H. Zhao, "Diffusion maximum correntropy criterion algorithms for robust distributed estimation," *Digital Signal Processing.*, vol. 58, pp. 10-16, 2016.
22. J. T. Kong, J. W. Lee, S. E. Kim, S. Shin, and W. J. Song, "Diffusion LMS algorithms with multi combination for distributed estimation: Formulation and performance analysis," *Digital Signal Processing.*, vol. 71, pp. 117-130, Dec 2017.
23. A. Rastegarnia, "Reduced-Communication Diffusion RLS for Distributed Estimation Over Multi-Agent Networks," *IEEE Trans. Circuit and Systems-Part II: Express Briefs*, vol. 67, no. 1, pp. 177–181, 2020.
24. A. M. Wilson, T. Panigrahi, and A. Dubey, "Robust distributed Lorentzian adaptive filter with diffusion strategy in impulsive noise environment," *Digital Signal Processing.*, vol. 96, Jan 2020.
25. H. Zayyani, "Robust Minimum Disturbance Diffusion LMS for Distributed Estimation," *IEEE Trans. Circuit and Systems-Part II: Express Briefs*, vol. 68, no. 1, pp. 521–525, Jan 2021.
26. H. Chang, and W. Li, "Correction-Based Diffusion LMS Algorithms for Distributed Estimation," *Circuit, System, and Signal Processing*, vol. 39, pp. 4136–4154, 2020.
27. L. Hu, F. Chen, S. Duan, L. Wang, and J. Wu, "An Improved Diffusion Affine Projection Estimation Algorithm for Wireless Sensor Networks," *Circuit, System, and Signal Processing*, vol. 39, pp. 3173–3188, 2020.
28. H. Zayyani, A. Javaheri, "A Robust Generalized Proportionate Diffusion LMS Algorithm for Distributed Estimation," *IEEE Trans. Circuit and Systems-Part II: Express Briefs*, Early Access, Oct 2020.
29. F. Chen, L. Hu, P. Liu, and M. Feng, "A Robust Diffusion Estimation Algorithm for Asynchronous Networks in IoT," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 9103–9115, Sep 2020.
30. M. Korki, and H. Zayyani, "Weighted diffusion continuous mixed p-norm algorithm for distributed estimation in non-uniform noise environment," *Elsevier Signal Processing.*, vol. 164, pp. 225–233, Nov 2019.
31. X. Li, M. Feng, F. Chen, Q. Shi, and J. Kurths, "Robust distributed estimation based on a generalized correntropy logarithmic difference algorithm over wireless sensor networks," *Elsevier Signal Processing*, vol. 77, Dec 2020.
32. N. J. Bershad, E. Eweda, and J. C. M. Bermudez, "Stochastic analysis of the diffusion LMS algorithm for cyclostationary white Gaussian inputs," *Elsevier Signal Processing*, vol. 185, August 2021.

33. M. Shirazi, and A. Vosoughi, "On Distributed Estimation in Hierarchical Power Constrained Wireless Sensor Networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 442–459, May 2020.
34. F. S. Abkenar, and A. Jamalipour, "Energy Optimization in Association-Free Fog-IoT Networks," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 2, pp. 404–412, June 2020.
35. M. Amarlingam, K. V. V. Durga Prasad, P. Rajalakshmi, S. S. Channappayya, and C. S. Sastry, "A Novel Low-Complexity Compressed Data Aggregation Method for Energy-Constrained IoT Networks," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 3, pp. 717–730, Sep 2020.
36. Z. Yang, A. Gang, and W. U. Bajwa, "Adversary-Resilient Distributed and Decentralized Statistical Inference and Machine Learning," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 146–159, May 2020.
37. Y. Chen, S. Kar, and J. M. F. Moura, "Resilient Distributed Estimation Through Adversary Detection," *IEEE Tran. Signal Processing*, vol. 66, no. 9, pp. 2455–2469, May 2018.
38. H. Chang, and W. Li, "Correction-based diffusion LMS algorithms for secure distributed estimation under attacks," *Digital Signal Processing*, vol. 102, July 2020.
39. H. Chang, and W. Li, "Correction-Based Diffusion LMS Algorithms for Distributed Estimation," *Circuit, System, and Signal Processing*, vol. 39, pp. 4136–4154, 2020.
40. Y. Hua, F. Chen, S. Deng, S. Duan, and L. Wang, "Secure distributed estimation against false data injection attack," *Information Sciences (Elsevier)*, vol. 515, pp. 248–262, April 2020.
41. J. Li, W. Abbas, and X. Koutsoukos, "Resilient Distributed Diffusion in Networks With Adversaries," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 1–17, 2020.
42. Q. Shi, M. Feng, X. Li, S. Wang, and F. Chen, "A Secure Distributed Information Sharing Algorithm Based on Attack Detection in Multi-Task Networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 5125–5138, Dec 2020.
43. Y. Liu, and C. Li, "Secure Distributed Estimation over Wireless Sensor Networks under Attacks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1815–1831, Aug 2018.
44. K. Ntemos, J. Plata-Chaves, N. Kolokotronis, N. Kalouptsidis, and M. Moonen, "Secure Information Sharing in Adversarial Adaptive Diffusion Networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 111–124, March 2018.
45. P. Cheng, et al, "Asynchronous Fault Detection Observer for 2-D Markov Jump Systems," *IEEE Transactions on Cybernetics*, Early access, 2021.
46. V. Flipovic, N. Nedic, and V. Stojanovic, "Robust identification of pneumatic servo actuators in the real situations," *Forschung im Ingenieurwesen*, vol. 75, pp. 183–196, 2011.
47. V. Stojanovic, and N. Nedic, "Robust identification of OE model with constrained output using optimal input design," *Journal of the Franklin Institute*, vol. 353, no. 2, pp. 576–593, Jan 2016.
48. H. Fang, et al, "Adaptive optimization algorithm for nonlinear Markov jump systems with partial unknown dynamics," *International Journal of Robust and Nonlinear Control*, 2021.
49. H. Zayyani, M. Korki, and F. Marvasti, "A Distributed 1-bit Compressed Sensing Algorithm Robust to Impulsive Noise," *IEEE Communications Letters*, vol. 20, no. 6, pp. 1132–1135, June 2016.
50. G. Nunez, C. Borges, and A. Chorti, "Understanding the Performance of Software Defined Wireless Sensor Networks Under Denial of Service Attack," *Open Journal of Internet of Things*, vol. 5, no. 1, 2019.
51. J. G. Proakis, *Digital Communications*, Mc-GrawHill, 2001.
52. S. M. Kay, *Fundamentals of statistical signal processing: Estimation theory*, Prentice Hall, 1993.