

Robust multi-task distributed estimation based on generalized maximum correntropy criterion

Qian Hu(胡倩), Feng Chen(陈枫), and Ming Ye(叶明)

Citation: Chin. Phys. B, 2023, 32 (**6**): 068902. DOI: 10.1088/1674-1056/acb9fc
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Oscillatory Shannon entropy in the process of equilibration of nonequilibrium crystalline systems

A. Giri, Nilangshu K. Das, P. Barat
Chin. Phys. B, 2015, 24 (**8**): 088902. DOI: 10.1088/1674-1056/24/8/088902

Optimal network structure to induce the maximal small-world effect

Zhang Zheng-Zhen (张争珍), Xu Wen-Jun (许文俊), Zeng Shang-You (曾上游), Lin Jia-Ru (林家儒)
Chin. Phys. B, 2014, 23 (**2**): 028902. DOI: 10.1088/1674-1056/23/2/028902

Measuring causality by directional symbolic mutual information approach

Chen Gui (陈贵), Xie Lei (谢磊), Chu Jian (褚健)
Chin. Phys. B, 2013, 22 (**3**): 038902. DOI: 10.1088/1674-1056/22/3/038902

Robust multi-task distributed estimation based on generalized maximum correntropy criterion

Qian Hu(胡倩), Feng Chen(陈枫), and Ming Ye(叶明)†

College of Artificial Intelligence, Southwest University, Chongqing 400715, *China*

(Received 12 August 2022; revised manuscript received 9 January 2023; accepted manuscript online 8 February 2023)

False data injection (FDI) attacks are common in the distributed estimation of multi-task network environments, so an attack detection strategy is designed by combining the generalized maximum correntropy criterion. Based on this, we propose a diffusion least-mean-square algorithm based on the generalized maximum correntropy criterion (GMCC-DLMS) for multi-task networks. The algorithm achieves gratifying estimation results. Even more, compared to the related work, it has better robustness when the number of attacked nodes increases. Moreover, the assumption about the number of attacked nodes is relaxed, which is applicable to multi-task environments. In addition, the performance of the proposed GMCC-DLMS algorithm is analyzed in the mean and mean-square senses. Finally, simulation experiments confirm the performance and effectiveness against FDI attacks of the algorithm.

Keywords: distributed estimation, generalized correntropy, multi-task networks, adaptive filtering

PACS: 89.70.Cf, 84.40.Ua, 89.70.–a, 43.60.Jn **DOI:** 10.1088/1674-1056/acb9fc

1. Introduction

Distributed estimation originated from research in the field of computing, and it has been extended to wireless sensor networks (WSNs) by modern wireless communication technologies.[1–4] With the widespread use of WSNs in environmental monitoring, military, medical care, and other applications, distributed algorithms have been further investigated.[5,6] In general, WSNs consist of many sensor nodes deployed in an area. Sensors collect data and communicate with each other, and they work in a cooperative manner to solve some common tasks such as temperature, humidity, motion, and sound.[7,8] It is called distributed single-task estimation when all nodes are interested in the same task. Correspondingly, the tasks are different between clusters when the nodes can be divided into clusters, which is called distributed multi-task estimation.[9–11]

One of the characteristics of WSNs is that they are not quite secure. Attacks may exist in practical applications such as power systems,[12] Internet of things (IoT),[13] aviation,[14] unmanned aerial vehicle (UAV) networks,[15] and other environments, so it is essential to study attacks on distributed networks. So far, researchers have studied a variety of attacks in distributed networks, including Byzantine attacks,[16] denial of service attacks (DoS),[17] random attacks,[18] and false data injection (FDI) attacks.[12] An *et al.* proposed a distributed algorithm based on a local min-switching decision (LMSD) for Byzantine links and nodes.[19] For nonlinear network systems with resource constraints, Liu *et al.* investigated secure distributed state estimation under DoS attacks.[20] For random attacks, Xie *et al.* designed an attack-resistant distributed interval observer by taking advantage of interval attack estimation and attack frequency.[21]

In recent years, distributed algorithms that are effective against FDI attacks have mushroomed.[22] A distributed adaptive algorithm based on the Kullback–Leibler divergence is proposed to detect FDI attacks by Hua *et al.*[23] Liu *et al.* proposed a secure diffusion least-mean square (S-dLMS) algorithm to handle FDI attacks by combining two subsystems.[24] For target tracking with FDI attacks, a consensus-based distributed state estimation method is proposed by Shui *et al.*[25] However, the above-mentioned papers considered single-task estimation rather than multi-task estimation. Given that distributed multi-task estimation also has wide applications in practice, like pattern classification and multi-target tracking, it is realistic to consider how to defend against FDI attacks in multi-task environments.[26] Additionally, it is not reasonable to migrate the approach of defending against attacks from single-task estimation to multi-task estimation directly, so researchers investigate resisting attacks in multi-task environments in these years. Li *et al.* developed a resilient diffusion least mean-square (RDLMS) algorithm motivated by a mean sub-sequence reduced (MSR) approach over multi-task networks in the presence of any data falsification attack.[27] They also presented the trade-off between the resilience of the distributed strategy and its performance of the networked mean square deviation (MSD) in qualitative analysis. Further, in scenarios of multi-task networks with impulsive interferences and Byzantine attacks, Yu *et al.* developed a robust resilient diffusion least mean Geman–McClure-estimation (RDLMG) algorithm based on the cost function used by the Geman–McClure estimator.[28] As mentioned by Yu *et al.*, the information from normal neighbors may be discarded, because nodes always discard extremes from their neighbors that are different from their own state, even if the network is secure.

Based on the above-mentioned literature, this paper con-

†Corresponding author. E-mail: zmxym@swu.edu.cn

© 2023 Chinese Physical Society and IOP Publishing Ltd

siders a distributed estimation algorithm to resist attacks for multi-task network environments. Chen *et al.* studied entropy and correntropy, and proposed generalized maximum correntropy criterion (GMCC), the kernel minimum error entropy (KMEE) algorithm, and other achievements.[29–33] Researchers found that, compared with minimum mean square error (MMSE), the GMCC can obtain a more robust performance when there are outliers.[34,35] In addition, it can be proved that the theoretical prediction error is bounded. Thus, the GMCC is incorporated for detecting attacked nodes by taking advantage of the boundedness of prediction error.

The main contributions of this paper are outlined as follows:

(i) For the FDI attacks, an attack detection strategy based on the GMCC is proposed. The assumption about the number of attacked nodes in multi-task network environments is relaxed.

(ii) For multi-task network environments, a distributed estimation algorithm is proposed to defend against attacks. The algorithm is analyzed theoretically, and mean and mean-square performance analysis is presented.

(iii) The effectiveness of the proposed algorithm is confirmed by simulations. Compared to related work, the performance of the algorithm degrades slowly with the increasing number of attacked nodes, and the algorithm has better robustness.

The rest of the paper is structured as follows. In Section 2, the multi-task network model, the multi-task diffusion least-mean-square (DLMS) algorithm, and the attack model are introduced. The proposed diffusion DLMS algorithm based on the generalized maximum correntropy criterion (GMCC-DLMS) for multi-task networks is described carefully in Section 3. In Section 4, the mean and mean-square performance of the proposed algorithm is analyzed. Simulations are given in Section 5. Lastly, the full text is summarized in Section 6. What's more, a summary of the notations adopted in this work is presented in Table 1.

Table 1. Summary of symbols.

x	normal font denotes realizations of random variables.
$\boldsymbol{x}$	boldface small letters indicate random vectors.
$\mathrm{col}\{s,t\}$	column vector with entries s and t.
$\mathrm{vec}(A)$	vectorizes matrix A and stacks its columns on top of each other.
$\boldsymbol{R}$	boldface capital letters indicate matrices.
$\boldsymbol{I}_N$	identity matrix of $N\times N$.
$\mathrm{diag}\{s,t\}$	the index set of nodes for node k, which includes neighbors and itself.
C_s	the set of nodes in the s-th cluster.
$C_C(k)$	the cluster where node k resides.
$\|x\|$	Euclidean norm of its vector argument.
$\|x\|_\Sigma^2$	weighted square value $x^*\Sigma x$, where $*$ denotes complex conjugation for scalars.

2. Problem formulation and preliminaries

In this part, the network model and the multi-task DLMS algorithm without attacks will be introduced. Then, a brief introduction to FDI attacks will be shown.

2.1. Network model

For communications, a network with N nodes is considered. These nodes are distributed in a region with a certain topology. N_k denotes the set of nodes adjacent to node k, including the node k itself, and $|N_k|$ is the degree. At each iteration n, node k will jointly estimate the parameter w^{o} by the data $\{d_k(n),x_k(n)\}$. Scalar $d_k(n)$ and regression vector $x_k(n)$ of L-dimensional satisfy a linear relationship as follows:

$$d_k(n)=x_k^{\mathrm{T}}(n)w_k^{\mathrm{o}}+z_k(n), \tag{1}$$

where w_k^{o} is unknown L-dimensional column vector and $z_k(n)$ is the measurement noise.

In multi-task networks, all nodes are divided into M clusters depending on the tasks, and the same parameter is jointly estimated within each cluster. C_s and C_t denote the sets of nodes belonging to the s-th and t-th clusters respectively, and $s,t\in\{1,2,\ldots,M\}$ denote the index of the cluster. The relationship of the tasks between nodes can be expressed as

$$\begin{aligned} &w_k^{\mathrm{o}}=w_{C_s}^{\mathrm{o}},\ \ \forall k\in C_s,\\ &w_{C_s}^{\mathrm{o}}\sim w_{C_t}^{\mathrm{o}},\ \ \text{if } C_s,C_t \text{ are connected.}\end{aligned} \tag{2}$$

Reasonable assumptions are made in the following.

Assumption 1 The regression vectors $x_k(n)$ have a covariance matrix $\boldsymbol{R}_{x,k}$, where $\boldsymbol{R}_{x,k}=E\{x_k(n)x_k^{\mathrm{T}}(n)\}>0$; $x_k(n)$ is generated by $\boldsymbol{x}$ that is temporal stationary and white stochastic as well as spatial independent.

Assumption 2 $z_k(n)$ is a zero-mean Gaussian noise with covariance $\boldsymbol{R}_{z,k}$, which is identically independent distributed (i.i.d.) in time and independent in space.

2.2. Distributed multi-task DLMS without FDI attacks

For the distributed DLMS strategy, the Adaptive-Then-Combine (ATC) DLMS algorithm is employed because of the better performance of the ATC fusion rule. In multi-task environments, the unknown parameter estimation problem is solved by minimizing the following cost function:

$$\begin{aligned}&\min_{w_{C_1},\ldots,w_{C_M}} J^{\mathrm{glob}}(w_{C_1},\ldots,w_{C_M})\\ &=\sum_{k=1}^{N}E\left\{|d_k(n)-x_k^{\mathrm{T}}(n)w_{C_C(k)}|^2\right\},\end{aligned} \tag{3}$$

where $w_{C_C(k)} \in \{w_{C_1}, \ldots, w_{C_M}\}$ is the estimated parameter vector of the cluster where node k resides. Thus, the multi-task ATC DLMS algorithm is obtained as

$$\begin{cases} \varphi_k(n+1) = w_k(n) + \mu\Bigg(\sum\limits_{l\in N_k\cap C_C(k)} c_{lk}[d_l(n) \\ \qquad\qquad - x_l^{\mathrm{T}}(n)w_k(n)]x_l(n) \\ \qquad\qquad + \eta \sum\limits_{l\in N_k\setminus C_C(k)} \rho_{kl}(w_l(n) - w_k(n))\Bigg), \\ w_k(n+1) = \sum\limits_{l\in N_k\cap C_C(k)} a_{lk}\varphi_l(n+1), \end{cases} \tag{4}$$

where μ is the step size, η is the strength parameter, and the non-negative coefficients ρ_{kl} aim at adjusting the regularization strength. The first formula in Eq. (4) describes that node k uses measurements of neighbors in $N_k \cap C_C(k)$ to obtain an intermediate estimate, and the second formula in Eq. (4) describes the fusion of node k using the intermediate estimates of neighbors in $N_k \cap C_C(k)$. Both c_{lk} and a_{lk} are non-negative coefficients, and they are (l,k)-th terms in the left random matrix $\boldsymbol{A}$ and the right random matrix $\boldsymbol{C}$:

$$\boldsymbol{A}^{\mathrm{T}} I_N = I_N, \quad \boldsymbol{C} I_N = I_N. \tag{5}$$

The coefficients satisfy the following formulas:

$$\begin{aligned} &a_{lk} \geq 0, \quad \sum_{l\in N_k} a_{lk} = 1, \quad a_{lk} = 0 \ \text{ if } \ l \notin N_k, \\ &c_{lk} \geq 0, \quad \sum_{k\in N_l} c_{lk} = 1, \quad c_{lk} = 0 \ \text{ if } \ k \notin N_l. \end{aligned} \tag{6}$$

The diffusion ATC algorithm has different effects with different fusion matrices. When $\boldsymbol{C} = \boldsymbol{I}$, the node transmits only the intermediate estimate $\varphi_k(n+1)$ to the neighbor nodes. Correspondingly, when $\boldsymbol{C} \neq \boldsymbol{I}$, the scalar $d_k(n+1)$ and the regression vector $x_k(n+1)$ are transmitted to neighbor nodes in addition to the intermediate estimate $\varphi_k(n+1)$. In this study, $\boldsymbol{C} = \boldsymbol{I}$ is chosen to exclude the possible interference of the attack on the scalar $d_k(n+1)$ and the regression vector $x_k(n+1)$.

2.3. Attack model

The FDI attack is mainly considered in this work. An attacker can inject arbitrary false data to modify node data or control structure. The attack model for an arbitrary node k is expressed as

$$w_k^*(n) = w_k^{\mathrm{o}} + \omega_k, \tag{7}$$

where ω_k is the L-dimensional interference vector. Thus, the linear measurement model can be modified to

$$d_k(n) = x_k^{\mathrm{T}}(n)w_k^*(n) + z_k(n). \tag{8}$$

This study aims to propose an attack detection algorithm for multi-task environments in presence of attacks. In multi-task networks, if node k needs to utilize information from its neighbors to detect attacked nodes, and only the intermediate estimates of their neighbors at the same cluster are employed in the fusion step, it is inappropriate to assume that the number of attacked neighbor nodes of any node k is no more than $\lceil |N_k|/2 \rceil$.[24] In addition, for networks satisfying the F-local model, it is always difficult to determine F without priori information.[27] Therefore, the following relaxed and more reasonable assumption is made for the multi-task network environments:

Assumption 3 The network is initially secure. For any node k, all neighbors of the same cluster cannot be attacked at the same time.

3. Secure distributed estimation algorithm

The GMCC is introduced for constructing an attack detection strategy, leading to the proposed GMCC-DLMS applicable to the attacked multi-task environments. The GMCC-DLMS has five main parts: the transmission of estimates, incremental step, the transmission of intermediate estimates, attack detection, and diffusion step. Among them, the incremental step, attack detection, and diffusion step of the algorithm are described in detail in the following. The flowchart of the multi-task GMCC-DLMS is obtained in Fig. 1.

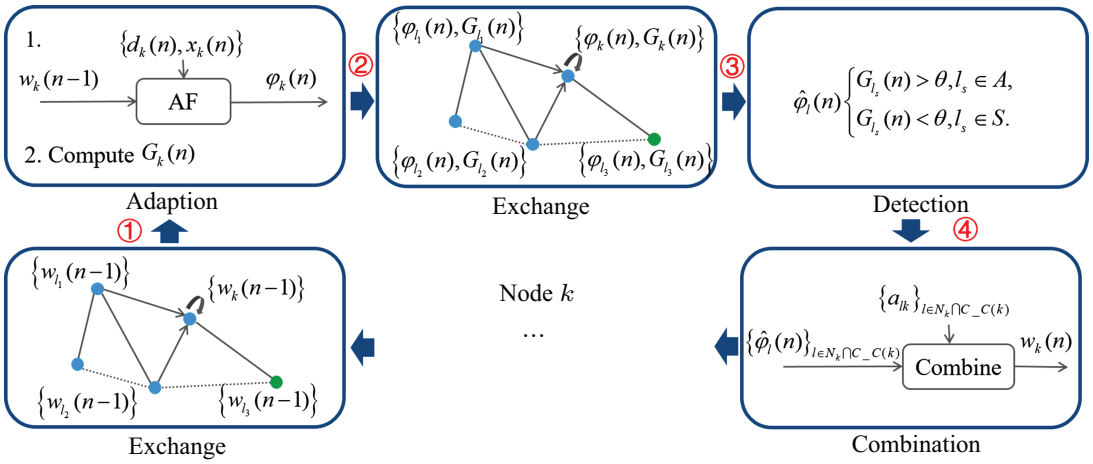


Fig. 1. The flowchart of the multi-task GMCC-DLMS.

3.1. Incremental step

In the first step, the estimated parameter of node k at the iteration n is transmitted to the neighbors at the same cluster. At the same time, node k will receive estimates $w_l(n)$ from its neighbor nodes l. These estimate parameters will be used for the incremental step and attack detection.

As the presence of attacks in the networks, the data $\{d_l(n), x_l(n)\}$ of neighbor node l may be attacked. Therefore, preventing node k unknowingly uses the attacked information, a non-cooperative form is taken in the adaptive estimation. That is, the estimate $w_k(n+1)$ is updated by $\{d_k(n+1), x_k(n+1)\}$, $w_k(n)$, and $w_l(n)$. The expression is as follows:

$$\varphi_k(n+1) = w_k(n) + \mu\big([d_k(n+1) - x_k^{\mathrm{T}}(n+1)w_k(n)]x_k(n+1) + \eta \sum_{l\in N_k\setminus C_C(k)} \rho_{kl}(w_l(n) - w_k(n))\big). \tag{9}$$

3.2. Attack detection

When the adaptive estimation is completed, node k transmits the intermediate estimate $\varphi_k(n+1)$ and calculated value $G_k(\mathrm{n}+1)$ to neighbors in $N_k \cap C_C(k)$. At the same time, node k will receive $\varphi_l(n+1)$ and $G_l(\mathrm{n}+1)$ from those neighbors. The rest of this section describes how to calculate $G_k(n+1)$. However, it is difficult to determine whether node k is attacked or normal based on intermediate estimates alone. Therefore, an attack detection step needs to be designed.

In distributed multi-task diffusion LMS, a node updates the estimated parameter by employing weighted average estimated parameters of all nodes in $N_k \cap C_C(k)$. However, the information of those attacked nodes cannot be adopted directly, or the network will be affected. We use information from the lowest-cost neighbors to replace information of some higher-cost neighbors. Correntropy is a robust nonlinear similarity measure in kernel space. In Ref. [29], Chen *et al.* proposed to use the generalized Gaussian density (GGD) function as a kernel function in correntropy, and the new correntropy was called the generalized correntropy, i.e., GMCC. In data analysis such as regression and classification, a measure called the generalized correntropic loss (GC-loss), instead of the generalized correntropy, can be used. Minimizing the GC-loss will be equivalent to maximizing the generalized correntropy. Compared with the MMSE, GMCC can achieve more robust performance in the presence of outliers, so the proposed algorithm combines GMCC to determine whether the received information is attacked. The GC-loss function is available:[29]

$$G_{\mathrm{GC-loss}} = \gamma_{\alpha,\beta}\left\{1 - E\left[\exp(-\lambda|e(n)|^{\alpha})\right]\right\}, \tag{10}$$

where $\gamma_{\alpha,\beta} = \alpha/(2\beta\Gamma(1/\alpha))$ is the normalization constant, $\alpha > 0$ is the shape parameter, $\beta > 0$ is the scale parameter, $\Gamma(\cdot)$ is the gamma function, and $\lambda = 1/\beta^{\alpha}$ is the kernel parameter. In addition, $e(n) = d(n) - x^{\mathrm{T}}(n)w(n)$ denotes the error at iteration n. Equation (10) is the global loss function, from which the loss function of the node k at iteration $n+1$ can be obtained as

$$G_k(n+1) = \gamma_{\alpha,\beta}\left\{1 - \exp(-\lambda|e_k(n+1)|^{\alpha})\right\}. \tag{11}$$

Since $\gamma_{\alpha,\beta}$ is a normalization constant, Eq. (11) can be rewritten as

$$G_k(n+1) = 1 - \exp(-\lambda|e_k(n+1)|^{\alpha}). \tag{12}$$

The prediction error at iteration $n+1$ is $e_k(n+1) = d_k(n+1) - x_k^{\mathrm{T}}(n+1)w_k(n)$ using the estimated parameter $w_k(n)$. Thus, the function $G_k(n+1)$ measures the error in the parameter estimation. There will be an abnormal increase in the prediction error $e_k(n+1)$ if the node k is attacked.

At iteration $n+1$, each node computes the $G_k(n+1)$ using its estimated parameter $w_k(n)$. Then, node k and its neighbors in $N_k \cap C_C(k)$ exchange the calculated value $G_k(n+1)$ and $G_l(n+1)$. After finishing the process, node k sorts all values in ascending order:

$$G_{l_1}(n+1) < \cdots < G_{l_m}(n+1) < \cdots < G_{l_{|N_k\cap C_C(k)|}}(n+1), \tag{13}$$

where $m \in \{1, 2, \ldots, |N_k \cap C_C(k)|\}$.

Compared to other nodes in $N_k \cap C_C(k)$, the node l_1 has the smallest value $G_{l_1}(n+1)$ with the smallest prediction error. As a consequence, the intermediate estimated $\varphi_{l_1}(n+1)$ is considered to be optimal among them. The attack makes a large difference between the values of prediction error for the abnormal node and the normal node. That is the value of prediction error because the attacked node will be an outlier. In the proposed attack detection strategy, the threshold is set for discrimination since the $G_{l_m}(n+1)$ of the attacked node will be abnormally large,

$$\begin{cases} G_{l_m}(n+1) > \theta,\ l_m \in A, \\ G_{l_m}(n+1) < \theta,\ l_m \in S, \end{cases} \tag{14}$$

where $l_m \in A$ and $l_m \in S$ indicate that node l_m is under attack and is secure, respectively.

The GC-loss function used for the above attack detection is based on the GGD function. At zero mean it is

$$\begin{aligned} G_{\alpha,\beta}(e) &= \frac{\alpha}{2\beta\Gamma(1/\alpha)} \exp\left(-\left|\frac{e}{\beta}\right|^{\alpha}\right) \\ &= \gamma_{\alpha,\beta}\exp(-\lambda|e|^{\alpha}). \end{aligned} \tag{15}$$

For the parametric family of symmetric distributions, the Gaussian and Laplace distributions are included and they are particular cases when $\alpha = 2$ and $\alpha = 1$, respectively. When the two distributions are used for attack detection similar to Eq. (12), the two expressions are

$$K_k(n+1) = 1 - \exp\left(-\frac{e_k^2(n+1)}{2\sigma^2}\right), \tag{16}$$

$$L_k(n+1) = 1 - \exp\left(-\frac{|e_k(n+1)|}{b}\right), \tag{17}$$

where σ indicates the kernel parameter and b denotes the scale parameter.

Algorithmic 1 GMCC-DLMS algorithm

Input: $\{d_k, x_k\} \in \mathbb{R} \times \mathbb{R}^L (k = 1, 2, \ldots, N)$

Initialization: $w_k(0) = 0,\ \mu > 0,\ a_{lk} > 0,\ \alpha > 0,\ \lambda > 0,\ \theta \in (0,1)$

1: **for** $n = 2 \to T$ **do**

2: **for** $k = 1 \to N$ **do**

3: Adaption:

$$\varphi_k(n+1) = w_k(n) + \mu\left([d_k(n+1) - x_k^{\mathrm{T}}(n+1)w_k(n)]x_k(n+1) + \eta \sum_{l \in N_k \backslash C_C(k)} \rho_{kl}(w_l(n) - w_k(n))\right)$$

5: **Detection:**

6: **for** $l = 1 \to |N_k \cap C_C(k)|$ **do**

7: Computation: $G_l(n+1) = 1 - \exp(-\lambda |e_l(n+1)|^{\alpha})$

8: Sort: $G_{l_1}(n+1) < \ldots < G_{l_m}(n+1) < \ldots < G_{l_{|N_k \cap C_C(k)|}}(n+1)$

9: Judgement: $\begin{cases} G_{l_m}(n+1) > \theta,\ l_m \in A \\ G_{l_m}(n+1) < \theta,\ l_m \in S \end{cases}$

10: **end for**

11: **Combination:**

12: $\hat{\varphi}_{l_m}(n+1) = \begin{cases} \varphi_{l_m}(n+1) = \varphi_{l_1}(n+1),\ l_m \in A \\ \varphi_{l_m}(n+1) = \varphi_{l_m}(n+1),\ l_m \in S \end{cases}$

13: $w_k(n+1) = \sum_{l \in N_k \cap C_C(k)} a_{lk} \hat{\varphi}_l(n+1)$

14: **end for**

15: **end for**

3.3. Diffusion step

After identifying the attacked nodes, appropriate measures need to be taken before fusion. When $G_{l_m}(n+1) > \theta$, the node l_m is considered to be under attack and its intermediate estimate is replaced with $\varphi_{l_1}(n+1)$. Otherwise, the node is safe and does nothing. Expression is as follows:

$$\hat{\varphi}_{l_m}(n+1) = \begin{cases} \varphi_{l_m}(n+1) = \varphi_{l_1}(n+1), & l_m \in A, \\ \varphi_{l_m}(n+1) = \varphi_{l_m}(n+1), & l_m \in S. \end{cases} \tag{18}$$

Thus the fusion step of Eq. (4) can be rewritten as

$$w_k(n+1) = \sum_{l \in N_k \cap C_C(k)} a_{lk} \hat{\varphi}_l(n+1). \tag{19}$$

For coefficient a_{lk}, there are several combination rules such as Laplacian, Metropolis, and maximum-degree.[36] The Metropolis criterion is used in this study:

$$a_{lk} = \begin{cases} 1/\max(n_k, n_l), & \text{if } l \in N_k \cap C_C(k)/\{k\}, \\ 1 - \sum_{l \in N_k \cap C_C(k) \backslash \{k\}} a_{lk}, & \text{if } l \in k, \\ 0, & \text{otherwise}, \end{cases} \tag{20}$$

where n_k is the degree of node k.

The pseudocode of the proposed algorithm is raised in algorithm 1.

4. Performance analysis

A more comprehensive performance analysis of the multi-task GMCC-DLMS will be presented in this section. For the node k at iteration n, the weight error vector is defined as

$$v_k(n) = w_k(n) - w_k^{\mathrm{o}}. \tag{21}$$

To facilitate the calculation, the information in the network is represented as block vectors and matrices. Expressions $v(n)$, $w(n)$, w^{o}, and $\varphi(n)$ denote the block weight error vector, the block weight estimate vector, the block optimal weight vector, and the block intermediate weight estimate vector with size $LN \times 1$, respectively,

$$v(n) = \mathrm{col}\{v_k(n)\}_{k=1}^N, \tag{22}$$

$$w(n) = \mathrm{col}\{w_k(n)\}_{k=1}^N, \tag{23}$$

$$w^{\mathrm{o}} = \mathrm{col}\{w_k^{\mathrm{o}}\}_{k=1}^N, \tag{24}$$

$$\varphi(n) = \mathrm{col}\{\varphi_k(n)\}_{k=1}^N. \tag{25}$$

4.1. Performance of ideal detection

The estimation error generated by the first equation in Eq. (4) is

$$d_l(n) - x_l^{\mathrm{T}}(n)w_k(n) = z_l(n) - x_l^{\mathrm{T}}(n)v_k(n) + x_l^{\mathrm{T}}(n)\omega_l, \tag{26}$$

where ω_l is a random interference vector. Noticeably, $w_l^{\mathrm{o}} = w_k^{\mathrm{o}}$ when $l \in N_k \cap C_C(k)$.

Using w_k^{o} from each side of the first expression in Eq. (4) and employing Eq. (26), we can rewrite the first expression in Eq. (4) as

$$\begin{aligned} \varphi(n+1) - w^{\mathrm{o}} &= v(n) - \mu \boldsymbol{X}_x(n)v(n) + \mu \boldsymbol{X}_x(n)\omega(n) \\ &\quad + \mu p_{zx}(n) - \mu\eta Q(v(n) + w^{\mathrm{o}}), \end{aligned} \tag{27}$$

$$\boldsymbol{Q} = \boldsymbol{I}_{LN} - \boldsymbol{P} \otimes \boldsymbol{I}_L, \tag{28}$$

where $\otimes$ denotes the Kronecker product. $\boldsymbol{P}$ is $N \times N$ matrix and has (k,l)-th entry ρ_{kl}, and $\boldsymbol{P}_{kk} = 1$ if $N_k \backslash C_C(k)$ is empty. Moreover, $\boldsymbol{X}_x(n)$ is a block diagonal matrix of $LN \times LN$:

$$\boldsymbol{X}_x(n) = \mathrm{diag}\left\{\sum_{l \in N_k \cap C_C(k)} c_{lk} x_l(n) x_l^{\mathrm{T}}(n)\right\}_{k=1}^N, \tag{29}$$

and $p_{zx}(n)$ is the following $LN \times 1$ vector:

$$p_{zx}(n) = \mathrm{col}\left\{\sum_{l \in N_k \cap C_C(k)} c_{lk} x_l^{\mathrm{T}}(n) z_l(n)\right\}_{k=1}^N. \tag{30}$$

Let $\boldsymbol{A} = \boldsymbol{A} \otimes \boldsymbol{I}_L$, the second equation in Eq. (4) can be rewritten as

$$w(n+1) = \boldsymbol{A}^{\mathrm{T}} \boldsymbol{\varphi}(n+1). \tag{31}$$

Then, w^{o} is subtracted from each side of Eq. (31) to obtain

$$v(n+1) = \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{\varphi}(n+1) - w^{\mathrm{o}}) + (\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{I}_{LN}) w^{\mathrm{o}}. \tag{32}$$

The iterative formula for the error vector is obtained by substituting Eq. (27) into the above equation:

$$\begin{aligned} v(n+1) \;=\; & \boldsymbol{A}^{\mathrm{T}}[\boldsymbol{I}_{LN} - \mu(\boldsymbol{X}_x(n) + \eta \boldsymbol{Q})] v(n) \\ & + [\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{I}_{LN} - \mu \eta \boldsymbol{A}^{\mathrm{T}} \boldsymbol{Q}] w^{\mathrm{o}} \\ & + \mu \boldsymbol{A}^{\mathrm{T}} \boldsymbol{X}_x(n) \boldsymbol{\omega}(n) + \mu \boldsymbol{A}^{\mathrm{T}} p_{zx}(n). \end{aligned} \tag{33}$$

For simplicity of analysis, we take the following substitution:

$$\boldsymbol{D}(n) = \boldsymbol{A}^{\mathrm{T}}[\boldsymbol{I}_{LN} - \mu(\boldsymbol{X}_x(n) + \eta \boldsymbol{Q})], \tag{34}$$

$$g(n) = \underbrace{[\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{I}_{LN} - \mu \eta \boldsymbol{A}^{\mathrm{T}} \boldsymbol{Q}] w^{\mathrm{o}}}_{g_q(n)} + \underbrace{\mu \boldsymbol{A}^{\mathrm{T}} \boldsymbol{X}_x(n) \boldsymbol{\omega}(n)}_{g_\omega(n)}, \tag{35}$$

$$\boldsymbol{X}(n) = \mu \boldsymbol{A}^{\mathrm{T}} p_{zx}(n). \tag{36}$$

Then, Eq. (33) is simplified to

$$v(n+1) = \boldsymbol{D}(n) v(n) + \boldsymbol{X}(n) + g(n). \tag{37}$$

(i) Mean error

The mean error measures performance from the perspective of the parameter.

Taking the expectation for each side of Eq. (37) yields

$$E\{v(n+1)\} = E\{\boldsymbol{D}(n) v(n) + \boldsymbol{X}(n) + g(n)\}. \tag{38}$$

To make the expressions more concise, the following substitution is made for the expected value:

$$\begin{aligned} \boldsymbol{X}_x \;=\; & E\{\boldsymbol{X}_x(n)\} \\ =\; & E\left\{ \mathrm{diag}\left\{ \sum_{l \in N_k \cap C_C(k)} c_{lk} \boldsymbol{R}_{x,l} \right\}_{k=1}^{N} \right\}, \end{aligned} \tag{39}$$

$$p_{zx} \;=\; E\{p_{zx}(n)\} = 0. \tag{40}$$

Based on the covariance matrix of neighbor nodes in the same cluster, $\boldsymbol{R}_k = \sum_{l \in N_k \cap C_C(k)} c_{lk} \boldsymbol{R}_{x,l}$ is obtained.

The expectations for $\boldsymbol{D}(n)$ and $g(n)$ in Eq. (38) are defined as

$$\boldsymbol{D} = E\{\boldsymbol{D}(n)\} = \boldsymbol{A}^{\mathrm{T}}[\boldsymbol{I}_{LN} - \mu(\boldsymbol{X}_x(n) + \eta \boldsymbol{Q})], \tag{41}$$

$$g_q = E\{g_q(n)\} = (\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{I}_{LN} - \mu \eta \boldsymbol{A}^{\mathrm{T}} \boldsymbol{Q}) w^{\mathrm{o}}, \tag{42}$$

$$g_\omega = E\{g_\omega(n)\} = \mu \boldsymbol{A}^{\mathrm{T}} \boldsymbol{X}_x(n) \boldsymbol{\omega}. \tag{43}$$

Taking the expectation for Eq. (38) yields the expectation of $v(n)$:

$$E\{v(n+1)\} = \boldsymbol{D} E\{v(n)\} + g_q + g_\omega. \tag{44}$$

The communication of the nodes is not affected when attacks can be accurately detected, so we can consider $g_\omega = 0$. Equation (44) is modified to

$$E\{v(n+1)\} = \boldsymbol{D} E\{v(n)\} + g_q. \tag{45}$$

Theorem 1 (i.e., mean stability) Mean stability indicates whether and to what value the mean error converges as the number of iterations increases. According to the hypotheses and model given above, the distributed multi-task GMCC-DLMS algorithm converges asymptotically to the mean when the step size meets the condition as follows:[10]

$$0 < \mu < \frac{2}{\max_k \{\lambda_{\max}(\boldsymbol{R}_k)\} + 2\eta}, \tag{46}$$

where $\lambda_{\max}$ indicates the maximum eigenvalue. In accordance with Eq. (45), the asymptotic mean deviation is deduced as

$$\lim_{n \to \infty} E\{v(n+1)\} = (\boldsymbol{I}_{LN} - \boldsymbol{D})^{-1} g_q. \tag{47}$$

(ii) Mean-square error

Mean-square error measures performance from a scalar perspective. Since it is possible that the error vectors $v(n+1)$ converge to zero on average, but they may have large fluctuations around the zero value, it makes sense to find the mean-square error and mean-square stability.

Based on Eqs. (37) and (40), the mean-square error can be obtained by a positive semi-definite matrix Σ satisfying

$$\begin{aligned} & E\{\|v(n+1)\|_\Sigma^2\} \\ =\; & E\{\|v(n)\|_{\Sigma'}^2\} + \mathrm{Tr}\{\Sigma E\{\boldsymbol{X}(n) \boldsymbol{X}^{\mathrm{T}}(n)\}\} \\ & + E\{\|g_q(n)\|_\Sigma^2\} + 2E\{g_q^{\mathrm{T}}(n) \Sigma \boldsymbol{D}(n) v(n)\}, \end{aligned} \tag{48}$$

where $\|x\|_\Sigma^2 = x^{\mathrm{T}} \Sigma x$ and $\Sigma' = E\{\boldsymbol{D}(n)^{\mathrm{T}} \Sigma \boldsymbol{D}(n)\}$. To simplify analysis, let

$$\boldsymbol{C} = \boldsymbol{C} \otimes \boldsymbol{I}_L, \tag{49}$$

$$\Gamma(g_q(n), \Sigma, v(n)) = \|g_q(n)\|_\Sigma^2 + 2 g_q^{\mathrm{T}}(n) \Sigma \boldsymbol{D}(n) v(n), \tag{50}$$

$$\begin{aligned} \boldsymbol{X} &= E\{\boldsymbol{X}(n) \boldsymbol{X}^{\mathrm{T}}(n)\} \\ &= \boldsymbol{A}^{\mathrm{T}} \boldsymbol{C}^{\mathrm{T}} \mathrm{diag}\{\sigma_{z,1}^2 \boldsymbol{R}_{x,1}, \ldots, \sigma_{z,N}^2 \boldsymbol{R}_{x,N}\} \boldsymbol{C} \boldsymbol{A}. \end{aligned} \tag{51}$$

According to the above simplified formulas, Eq. (48) can be rewritten as

$$\begin{aligned} E\{\|v(n+1)\|_\Sigma^2\} \;=\; & E\{\|v(n)\|_{\Sigma'}^2\} + \mathrm{Tr}\Big\{\Sigma \boldsymbol{X}\Big\} \\ & + E\Big\{\Gamma(g_q(n), \Sigma, v(n))\Big\}. \end{aligned} \tag{52}$$

To verify the mean-square stability, it is necessary to prove that the mean-square error converges at sufficiently small step sizes. Vectorize the matrices Σ and Σ' into $\sigma = \mathrm{vec}(\Sigma)$ and $\sigma' = \mathrm{vec}(\Sigma')$. A linear relationship existing between them is expressed as $\sigma' = \varepsilon \sigma$, where ε is an $(LN)^2 \times (LN)^2$ matrix represented by $\varepsilon = E\{\boldsymbol{D}(n)^{\mathrm{T}} \otimes \boldsymbol{D}(n)^{\mathrm{T}}\}$.[14] Here, ε can also be expressed as $\boldsymbol{D}^{\mathrm{T}} \otimes \boldsymbol{D}^{\mathrm{T}} + \mathcal{O}(\mu_{\max}^2)$. The step size satisfying the Eq. (46) is considered in this paper, so

terms involving higher power steps can be ignored. We can obtain the following expression:

$$\varepsilon \approx \boldsymbol{D}^{\mathrm{T}} \otimes \boldsymbol{D}^{\mathrm{T}}. \tag{53}$$

For the last term $E\{\Gamma(g_q(n),\Sigma,v(n))\}$ to the right of the equal sign of Eq. (52), the first term is $E\{\|g_q(n)\|_{\Sigma}^2\}$ by definition (50):

$$E\{\|g_q(n)\|_{\Sigma}^2\} = E\{g_q^{\mathrm{T}}(n)\sum g_q(n)\}. \tag{54}$$

Because the random component in $g_q(n)$ is related to the square of the step size, the expression is

$$E\{\|g_q(n)\|_{\Sigma}^2\} = \|g_q(n)\|_{\Sigma}^2 + \mathcal{O}(\mu_{\max}^2). \tag{55}$$

Similar to the above operation, we can obtain

$$\begin{aligned} &E\{g_q^{\mathrm{T}}(n)\Sigma\boldsymbol{D}(n)v(n)\} \\ &= g_q^{\mathrm{T}}(n)\Sigma\boldsymbol{D}(n)E\{v(n)\} + \mathcal{O}(\mu_{\max}^2). \end{aligned} \tag{56}$$

Neglecting sufficiently small higher-order terms of the step size yields

$$E\{\Gamma(g_q(n),\Sigma,v(n))\} = \Gamma(g_q(n),\Sigma,E\{v(n)\}). \tag{57}$$

Therefore, Eq. (52) can be approximated as

$$\begin{aligned} E\{\|v(n+1)\|_{\sigma}^2\} &= E\{\|v(n)\|_{\varepsilon\sigma}^2\} + \mathrm{vec}(\boldsymbol{X}^{\mathrm{T}})^{\mathrm{T}}\sigma \\ &\quad + \Gamma(g_q(n),\Sigma,E\{v(n)\}). \end{aligned} \tag{58}$$

The notations $\|\cdot\|_{\Sigma}^2$ and $\|\cdot\|_{\sigma}^2$ denote the same square weighted norm.

Theorem 2 (i.e., mean-square stability) Mean-square stability indicates whether and to what value the mean-square error converges as the number of iterations increases. First of all, it is assumed that the step size satisfies Eq. (46), and further assumed that the approximation terms (53) and (57) are adjusted by neglecting the higher-order terms of the step size. In addition, Eq. (58) is the updating process of mean square error with the number of iterations. It is available that the distributed multi-task diffusion algorithm can be mean-square stable when the matrix ε is stable. As the approximation (53) is obtained, the stability of matrix ε is ensured under the condition of sufficiently small step size.

Proof of Theorem 2 Equation (58) is started from $n=0$. It is found that

$$\begin{aligned} E\{\|v(n+1)\|_{\sigma}^2\} &= E\{\|v(0)\|_{\varepsilon^{n+1}\sigma}^2\} + \mathrm{vec}(X^{\mathrm{T}})^{\mathrm{T}}\sum_{i=0}^{n}\varepsilon^i\sigma \\ &\quad + \sum_{i=0}^{n}\Gamma(g_q(n-i),\varepsilon^i\sigma,E\{v(n-i)\}), \end{aligned} \tag{59}$$

where $v(0) = w(0) - w^{\mathrm{o}}$. When matrix ε is stable, the terms to the right of the equal sign in Eq. (59) converge to zero and bounded values. Thus, the mean-square stable is proved.

Theorem 3 (i.e., transient MSD) L-2 norm is used by MSD to measure performance from the parameter perspective. Let $\Sigma = \frac{1}{N}\boldsymbol{I}_{LN}$. The MSD learning curve is defined as $\gamma(n) = \frac{1}{N}E\{\|v(n)\|\}^2$. Its evolution is in terms of the following recursions:

$$\begin{aligned} \gamma(n+1) &= \gamma(n) + \frac{1}{N}\Big(\mathrm{vec}(\boldsymbol{X}^{\mathrm{T}})^{\mathrm{T}}\varepsilon^n\mathrm{vec}(\boldsymbol{I}_{LN}) \\ &\quad - E\{\|v(0)\|_{(\boldsymbol{I}_{(LN)^2}-\varepsilon)\varepsilon^n\mathrm{vec}(\boldsymbol{I}_{LN})}^2\} \\ &\quad + \|g_q(n)\|_{\varepsilon^n\mathrm{vec}(\boldsymbol{I}_{LN})}^2 - 2((\boldsymbol{D}E\{v(n)\} \\ &\quad \otimes g_q(n))^{\mathrm{T}} + \rho(n))\mathrm{vec}(\boldsymbol{I}_{LN})\Big), \end{aligned} \tag{60}$$

$$\rho(n+1) = \rho(n)\varepsilon + (\boldsymbol{D}E\{v(n)\}\otimes g_q(n))^{\mathrm{T}}(\varepsilon - \boldsymbol{I}_{(LN)^2}), \tag{61}$$

with $\gamma(0) = \frac{1}{N}\|v(0)\|^2$ and $\rho(0) = 0_{1\times LN}$.

Proof of Theorem 3 The expressions of Eq. (59) for iteration $n+1$ and n are written and then compared, items $E\{\|v(n+1)\|_{\sigma}^2\}$ and $E\{\|v(n)\|_{\sigma}^2\}$ are related by

$$\begin{aligned} &E\{\|v(n+1)\|_{\sigma}^2\} \\ &= E\{\|v(n)\|_{\sigma}^2\} + \mathrm{vec}(\boldsymbol{X}^{\mathrm{T}})^{\mathrm{T}}\varepsilon^n\sigma - E\{\|v(0)\|_{(\boldsymbol{I}_{(LN)^2}-\varepsilon)\varepsilon^n\sigma}^2\} \\ &\quad + \sum_{i=0}^{n}\Gamma(g_q(n-i),\varepsilon^i\sigma,E\{v(n-i)\}) \\ &\quad - \sum_{i=0}^{n-1}\Gamma(g_q(n-1-i),\varepsilon^i\sigma,E\{v(n-1-i)\}). \end{aligned} \tag{62}$$

In addition, the last two terms to the right of the equal sign of Eq. (62) are replaced with

$$\begin{aligned} &\sum_{i=0}^{n}\Gamma(g_q(n-i),\varepsilon^i\sigma,E\{v(n-i)\}) \\ &\quad - \sum_{i=0}^{n-1}\Gamma(g_q(n-1-i),\varepsilon^i\sigma,E\{v(n-1-i)\}) \\ &= \|g_q(n)\|_{\varepsilon^n\sigma}^2 - 2((\boldsymbol{D}E\{v(n)\}\otimes g_q(n))^{\mathrm{T}} + \rho(n))\sigma, \end{aligned} \tag{63}$$

where

$$\begin{aligned} \rho(n) &= \sum_{i=1}^{n}(\boldsymbol{D}E\{v(n-i)\}\otimes g_q(n-i))^{\mathrm{T}}\varepsilon^i \\ &\quad + \sum_{i=0}^{n-1}(\boldsymbol{D}E\{v(n-i-1)\}\otimes g_q(n-i-1))^{\mathrm{T}}\varepsilon^i. \end{aligned} \tag{64}$$

Thus Eq. (62) is rephrased as

$$\begin{aligned} &E\{\|v(n+1)\|_{\sigma}^2\} \\ &= E\{\|v(n)\|_{\sigma}^2\} + \mathrm{vec}(X^{\mathrm{T}})^{\mathrm{T}}\varepsilon^n\sigma - E\{\|v(0)\|_{(\boldsymbol{I}_{(LN)^2}-\varepsilon)\varepsilon^n\sigma}^2\} \\ &\quad + \|g_q(n)\|_{\varepsilon^n\sigma}^2 - 2((\boldsymbol{D}E\{v(n)\}\otimes g_q(n))^{\mathrm{T}} + \rho(n))\sigma, \end{aligned} \tag{65}$$

$$\rho(n+1) = \rho(n)\varepsilon + (\boldsymbol{D}E\{v(n)\}\otimes g_q(n))^{\mathrm{T}}(\varepsilon - \boldsymbol{I}_{(LN)^2}), \tag{66}$$

with $\rho(0) = 0_{1\times LN}$.

Theorem 4 (i.e., steady-state MSD) Steady-state MSD indicates whether and to what value the transient MSD converges as the number of iterations increases. The steady-state MSD is obtained when the step size is small enough for theories 1 and 2 to hold:

$$\gamma^* = \mathrm{vec}(\boldsymbol{X}^{\mathrm{T}})^{\mathrm{T}}\frac{1}{N}(\boldsymbol{I}_{(LN)^2} - \varepsilon)^{-1}\mathrm{vec}(\boldsymbol{I}_{LN})$$

$$+\Gamma(g_q(\infty),\frac{1}{N}(\boldsymbol{I}_{(LN)^2}-\varepsilon)^{-1}\mathrm{vec}(\boldsymbol{I}_{LN}),E\{v(\infty)\}), \quad (67)$$

where $E\{v(\infty)\}$ depends on Eq. (47).

Proof of Theorem 4 Steady-state MSD is a limiting value:

$$\gamma^* = \lim_{n\to\infty}\frac{1}{N}E\{\|v(n)\|\}^2. \quad (68)$$

As $n\to\infty$, we obtain the expression from the recursive Eq. (58) as follows:

$$\begin{aligned}&\lim_{n\to\infty}E\{\|v(n)\|^2_{(\boldsymbol{I}_{(LN)^2}-\varepsilon)\sigma}\}\\ &=\mathrm{vec}(\boldsymbol{X}^{\mathrm{T}})^{\mathrm{T}}\sigma+\Gamma\big(g_q(\infty),\Sigma,E\{v(\infty)\}\big).\end{aligned} \quad (69)$$

From Eqs. (68) and (69), it is clear that, to obtain the MSD, the σ can be chosen as follows:

$$(\boldsymbol{I}_{(LN)^2}-\varepsilon)\sigma=\frac{1}{N}\mathrm{vec}(\boldsymbol{I}_{LN}). \quad (70)$$

This yields the expression (67).

4.2. Performance of attack detection

It is determined that attack detection affects the estimation performance of the algorithm, and the performance of GMCC-DLMS is analyzed in this part when there are false detections in the detection strategy. Generally speaking, false detections are classified as false alarms and missed detections. Noteworthily, the performance of the algorithm in the case of false detections cannot be precisely obtained due to the lack of a priori attack information. For node k, the set of neighbors in the same cluster $W_k(n)=N_k\cap C_C(k)$ is divided into two subsets:

$$W_k(n)=W_k^+(n)\cup W_k^-(n). \quad (71)$$

$W_k^+(n)$ indicates that normal nodes in $W_k(n)$ are detected as secure, and attacked nodes in $W_k(n)$ detected exactly. $W_k^-(n)$ denotes that nodes in $W_k(n)$ have false alarms and missed detections. Here, it is assumed that when there are false detections, the intermediate estimate is expressed as

$$\tilde{\varphi}_l(n)=\varphi_l(n)+\tau_l(n). \quad (72)$$

In iteration n, the intermediate estimate of node l satisfies $\tilde{\varphi}_l(n)=\varphi_l(n)$ when node $l\in W_k^+(n)$. When node $l\in W_k^-(n)$, the intermediate estimate of node l satisfies Eq. (72). Here $\tau_l(n)$ is an L-dimensional vector that represents the difference between $\tilde{\varphi}_l(n)$ and $\varphi_l(n)$.

To describe more clearly the effect of false detections on the information fusion process of local estimation, the fusion coefficient a_{lk} is defined as

$$a_{lk}=\begin{cases}a_{lk}^+, \text{ if } l\in W_k^+(n),\\ a_{lk}^-, \text{ if } l\in W_k^-(n).\end{cases} \quad (73)$$

The information fusion step is modified to

$$\begin{aligned}\tilde{w}_k(n) &= \sum_{l\in W_k^+(n)}a_{lk}^+\varphi_l(n)+\sum_{l\in W_k^-(n)}a_{lk}^-\tilde{\varphi}_l(n)\\ &= \sum_{l\in X_k(n)}a_{lk}\varphi_l(n)+\sum_{l\in W_k^-(n)}a_{lk}^-\tau_l(n)\\ &= w_k(n)+\sum_{l\in W_k^-(n)}a_{lk}^-\tau_l(n).\end{aligned} \quad (74)$$

In the non-ideal detection, the weight error vector is defined as

$$\Delta v_k(n)=\tilde{w}_k(n)-w_k^{\mathrm{o}}. \quad (75)$$

The block weight error vector is $\Delta v(n)=\mathrm{col}\{\Delta v_k(n)\}_{k=1}^N$. Subtracting w_k^{o} from two sides of Eq. (74) yields

$$\Delta v(n+1)=v(n)+\mathcal{A}_-^{\mathrm{T}}\tau(n), \quad (76)$$

where $\mathcal{A}_+=\boldsymbol{A}_+\otimes\boldsymbol{I}_L$, $\mathcal{A}_-=\boldsymbol{A}_-\otimes\boldsymbol{I}_L$, $\tau(n)=\mathrm{diag}\Big\{\sum_{l\in W_k^-(n)}\tau_l(n)\Big\}_{k=1}^N$, and $\tau=E\{\mathcal{A}_-^{\mathrm{T}}\tau(n)\}$. The expectation of the global weight error recursion is obtained as follows:

$$E\{\Delta v(n+1)\}=E\{v(n)\}+\tau=DE\{v(n)\}+g_q+\tau. \quad (77)$$

Theorem 5 (i.e., mean stability for non-ideal detection) With the network model in Section 2 and the assumptions in the non-ideal detection case, the proposed algorithm is convergent asymptotically. When the step size satisfies Eq. (46), the asymptotic mean deviation is obtained as follows:

$$\lim_{n\to\infty}E\{\Delta v(n+1)\}=(\boldsymbol{I}-\boldsymbol{D})^{-1}(g_q+\tau). \quad (78)$$

From Eqs. (47) and (78), it is clear that bias is influenced by the detection probability of false detections. If there are missed detections, the term $(g_q+\tau)$ in Eq. (78) will not be zero when $n\to\infty$, which will lead to biased estimation. However, if there are false alarms, the mean stability of the estimate is guaranteed.

5. Simulations

In this part, the network topology and the simulation data needed for the experiment are shown first. Then, simulations are executed to prove the effectiveness of the multi-task GMCC-DLMS.

5.1. Illustrative model

As shown in Fig. 2(a), the multi-task network considered in this paper consists of 20 nodes, and they are classified into four clusters: $C_1=\{1,2,3,4,5\}$, $C_2=\{6,7,8,9,10,11\}$, $C_3=\{12,13,14\}$, $C_4=\{15,16,17,18,19,20\}$. The regression input $x_k(n)$ are four-dimensional zero-mean Gaussian random variables, and their covariance matrices are $R_{x,k}=\sigma_{x,k}^2 I_L$. The noises $z_k(n)$ are zero-mean Gaussian random variables, with covariance matrices $R_{z,k}=\sigma_{z,k}^2 I_L$, and are uncorrelated with other signals. As shown in Fig. 2(b), the top graph represents the variance $\sigma_{x,k}^2$, and the bottom graph represents the variance $\sigma_{z,k}^2$.

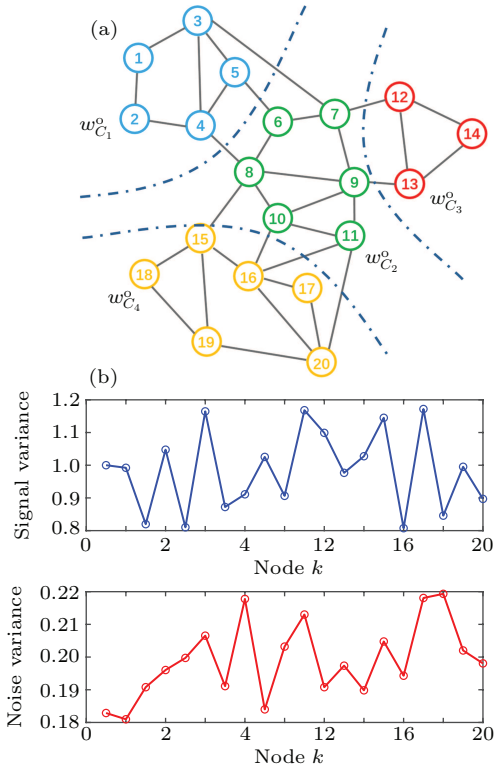


Fig. 2. (a) Network topology; (b) input signal variances (top) and noise variances (bottom).

5.2. The MSD performance with attack

Simulations are executed to describe the performance of the GMCC-DLMS algorithm when the multi-task network is under FDI attacks. These simulations include comparing the proposed algorithm with the work of other researchers and studying the effects of the number of attacked nodes on the proposed algorithm's performance. The simulation results run over 500 independent Monte Carlo simulations. It is assumed that attacks are generated by a Gaussian distribution $N(\mu_\omega, \sigma_\omega^2)$, where $\mu_\omega = 0$. Weak attack and strong attack are considered and their variances are set as $\sigma_\omega^2 = 1.5$ and $\sigma_\omega^2 = 10$, respectively. For the GMCC-DLMS algorithm in multi-task networks, the step size and the regularization parameter are $\mu = 0.02$ and $\eta = 0.1$, respectively. The performance of the algorithm is better as the shape parameter α increases, and it becomes insignificant when $\alpha > 6$ with the computational burden increasing. Therefore, $\alpha = 6$ is chosen in this work. A too small kernel parameter λ will lead to reduced performance and a too large λ will lead to slower convergence, so $\lambda = 0.04$ is chosen. A small threshold θ causes more nodes to be considered under attack, which leads to a slower convergence, so $\theta = 0.9$ is set.

(i) MSD performance comparison with other algorithms

Figures 3 and 4 show the performance of the algorithms when two random nodes are attacked by FDI attacks. The kernel parameter of the Gaussian distribution ($\alpha = 2$) is $\sigma = 1$, and the scale parameter of the Laplace distribution ($\alpha = 1$) is $b = 0.7$. It can be found that both the proposed algorithms and the S-dLMS algorithm can effectively resist the FDI attacks. They are both close to the unattacked multi-task DLMS algorithm.

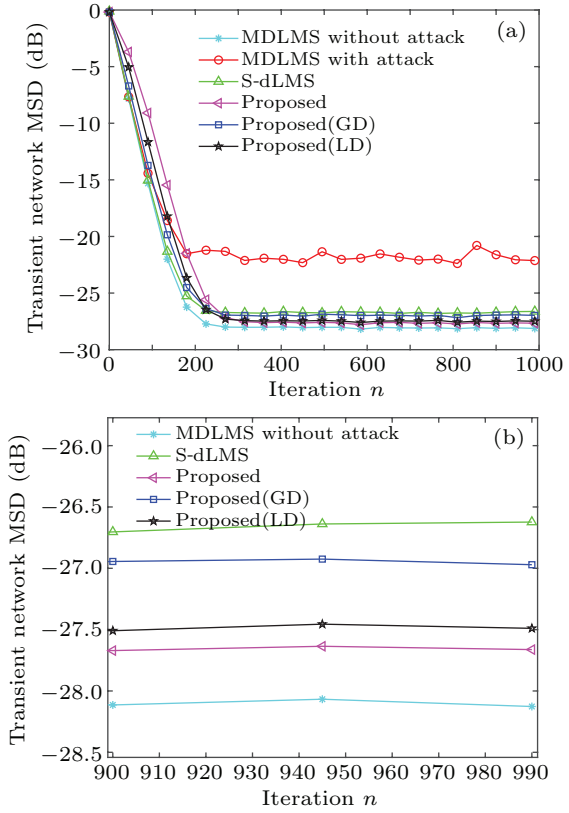


Fig. 3. Transient network MSD of different algorithms attacked by weak FDI ($\sigma_\omega^2 = 1.5$). (a) Comparison between different algorithms; (b) enlarged view of (a) after 900 iterations.

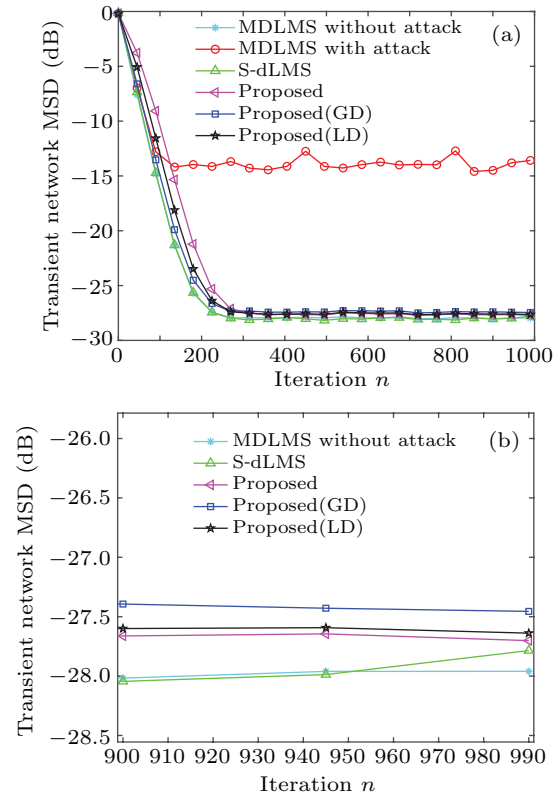


Fig. 4. Transient network MSD of different algorithms attacked by strong FDI ($\sigma_\omega^2 = 10$). (a) Comparison between different algorithms; (b) enlarged view of (a) after 900 iterations.

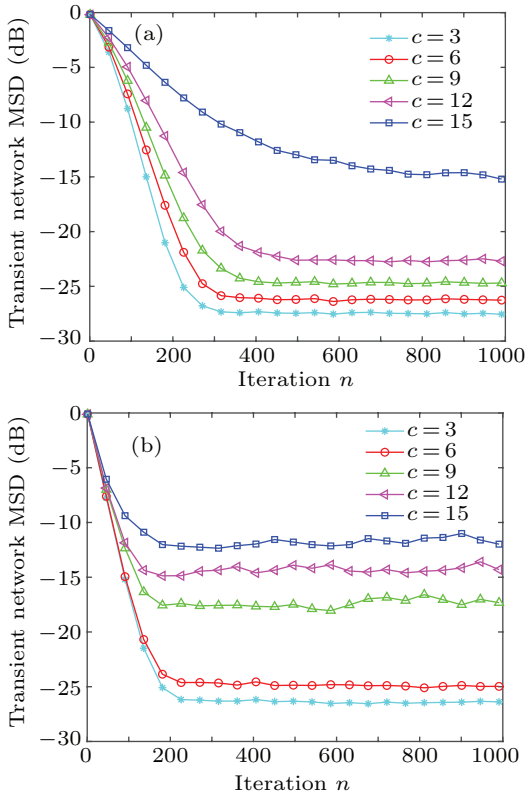


Fig. 5. Transient network MSD when the number of nodes under weak FDI attacks ($\sigma_\omega^2 = 1.5$) increases. (a) Proposed GMCC-DLMS algorithm; (b) S-dLMS algorithm.

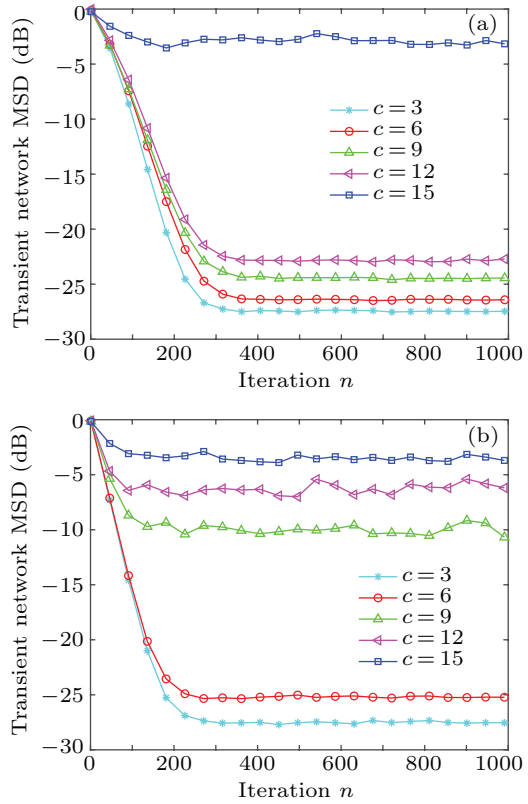


Fig. 6. Transient network MSD when the number of nodes under strong FDI attacks ($\sigma_\omega^2 = 10$) increases. (a) Proposed GMCC-DLMS algorithm; (b) S-dLMS algorithm.

(ii) MSD performance when the number of attacked nodes increases

The sign of c denotes the number of attacked nodes.[37] Figures 5(a), 6(a) and Figures 5(b), 6(b) show the performance of the proposed algorithm and the S-dLMS algorithm, respectively, and each algorithm considers the impact of the different number of attacked nodes. It is obvious that the performance of the proposed algorithm behaves well under the different number of attacked nodes. Even more, compared with other related works when c increases all the way up to 12, the proposed algorithm can still achieve satisfactory performance.

5.3. Complexity analysis

In this section, the computational complexity and the communication complexity of the proposed GMCC-DLMS are analyzed. In Table 2, H denotes the average degree of the nodes in the network. Y denotes the average number of the nodes that are neighbors at the same cluster of each node, i.e., $Y = (1/N)\sum_{k=1}^{N} |N_k \cap C_C(k)|$. According to Table 2, the estimated computational costs of the proposed algorithm GMCC-DLMS are smaller than those of the S-dLMS. For each node in each iteration, the GMCC-DLMS needs to transmit three quantities $w_k(n-1)$, $\varphi_k(n)$, and $G_k(n)$, while the S-dLMS needs to transmit two quantities $\breve{w}_k(n)$ and $\varphi_k(n)$. Here $\breve{w}_k(n)$ denotes the instantaneous estimates at node k and time n of the noncooperative LMS.

Table 2. Estimated computational costs of S-dLMS and GMCC-DLMS per iteration per node for real-valued data in terms of the number of real multiplication ($\times$), real addition ($+$), real division ($/$), and real comparison ($\prec$).

Algorithm	$\times$	$+$	$/$	$\prec$
S-dLMS	$7L+4$	$9L+2$	$2L$	$L(\log_2 L^2 H^H)+2$
GMCC-DLMS	$4L+\alpha+3$	$4L+H$	0	$Y+H\log_2 H$

6. Conclusion

For multi-task network environments with FDI attacks, a distributed security algorithm is proposed in this paper. The attack detection strategy detects outliers of prediction error based on the GC-loss function to determine which nodes are attacked. Moreover, we theoretically verify the stability and convergence of the proposed algorithm by analyzing the estimation performance in the mean and mean-square senses. In addition, simulations for the presence of attacks are executed. The experimental outcomes demonstrate that the proposed algorithm can effectively defend against FDI attacks, and the algorithm is robust as the number of attacked nodes increases compared with other related works.

Dynamic networks and time-varying tasks are widespread in actual scenarios, like wireless sensor networks, Industrial Internet of Things and vehicular networks. Therefore, these two points are worthy of subsequent consideration and research.

References

[1] Kassal P, Steinberg M D and Steinberg I M 2018 *Sensor. Actuat. B-Chem.* **266** 228

[2] Deng Y, Mo Z and Lu H 2022 *Chin. Phys. B* **31** 020503

[3] Modalavalasa S, Sahoo U K, Sahoo A K and Baraha S 2021 *Signal Process.* **188** 108150

[4] Park Y G, Lee S and Park J U 2019 *Sensors* **19** 4353

[5] Cattivelli F S and Sayed A H 2009 *IEEE Trans. Signal Process.* **58** 1035

[6] Hassani A, Plata-Chaves J, Bahari M H, Moonen M and Bertrand A 2017 *IEEE J. Sel. Top. Signal Process.* **11** 518

[7] Sun S, Lin H, Ma J and Li X 2017 *Inform. Fusion* **38** 122

[8] He S, Shin H S, Xu S Y and Tsourdos A 2020 *Inform. Fusion* **54** 21

[9] Chen F, Li X, Duan S, Wang L and Wu J 2018 *Digit. Signal Process.* **81** 16

[10] Chen J, Richard C and Sayed A H 2014 *IEEE Trans. Signal Process.* **62** 4129

[11] Huang W, Yang X and Shen G 2017 *Inf. Sci.* **382** 115

[12] Li B, Xiao G, Lu R, Deng R and Bao H 2020 *IEEE Trans. Ind. Inform.* **16** 854

[13] Wu H, Zhou B and Zhang C 2022 *IEEE Trans. Autom. Sci. Eng.* **19** 2552

[14] Song X, Zhang Q, Wang M and Song S 2021 *Nonlinear Dynam.* **106** 3183

[15] Borup K T, Fossen T I and Johansen T A 2020 *IEEE Trans. Aerosp. Electron. Syst.* **56** 2157

[16] Luo J and Cao Z 2015 *Int. J. Distrib. Sens. Netw.* **11** 381642

[17] Chen B, Ho D W C, Zhang W and Yu L 2019 *IEEE Trans. Syst. Man Cybern. -Syst.* **49** 455

[18] Yang Y, Xu H and Yue D 2019 *IEEE Trans. Circuits Syst. I:-Regul. Pap.* **66** 3089

[19] An L and Yang G 2021 *Automatica* **129** 109664

[20] Liu J, Suo W, Zha L, Tian E and Xie X 2020 *Int. J. Robust. Nonlin.* **30** 1156

[21] Xie J, Zhu S and Zhang D 2022 *ISA Trans.* **127** 22

[22] Zhou J, Chen B and Yu L 2020 *IEEE Trans. Circuits Syst. II:-Express Briefs* **67** 2762

[23] Hua Y, Chen F, Deng S, Duan S and Wang L 2020 *Inf. Sci.* **515** 248

[24] Liu Y and Li C 2018 *IEEE Trans. Aerosp. Electron. Syst.* **54** 1815

[25] Shui Y, Wang Y, Li Y, Shan Y, Cui N and Pang B 2021 *Appl. Sci.* **11** 4564

[26] Shi Q, Feng M, Li X, Wang S and Chen F 2020 *IEEE Trans. Circuits Syst. I:-Regul. Pap.* **67** 5125

[27] Li J, Abbas W and Koutsoukos X 2019 *IEEE Trans. Signal Inf. Process. Netw.* **6** 1

[28] Yu T, De Lamare R C and Yu Y 2022 *IEEE Trans. Signal Process.* **70** 2826

[29] Chen B, Xing L, Zhao H, Zheng N and Principe J C 2016 *IEEE Trans. Signal Process.* **64** 3376

[30] Chen B, Yuan Z, Zheng N and Principe J C 2013 *Neurocomputing* **121** 160

[31] Chen B, Xing L, Xu B, Zhao H and Principe J C 2016 *IEEE Trans. Neural Netw. Learn. Syst.* **29** 731

[32] Chen B, Dang L, Gu Y, Zheng N and Principe J C 2019 *IEEE Trans. Syst. Man Cybern.: -Syst.* **51** 5819

[33] Chen B, Xing L, Zheng N and Principe J C 2018 *IEEE Trans. Neural Netw. Learn. Syst.* **30** 1370

[34] Chen B, Wang X, Li Y and Principe J C 2019 *IEEE Signal Process. Lett.* **26** 1212

[35] Dang L, Chen B, Wang S, Gu Y and Principe J C 2019 *IEEE Trans. Circuits Syst. I:-Regul. Pap.* **66** 4265

[36] Sayed A H 2014 *Academic Press Library in Signal Processing* (Amerstadam: Elsevier) pp. 323–453

[37] Yu F, Dutta R G, Zhang T, Hu Y and Jin Y 2020 *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **39** 3555