Learning-Based Cooperative False Data Injection Attack and Its Mitigation Techniques in Consensus-Based Distributed Estimation

QIAOMU JIANG¹, HUIFANG CHEN¹,2,3, (Member, IEEE), LEI XIE¹,2, (Member, IEEE), AND KUANG WANG¹,2

¹College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China
²Zhejiang Provincial Key Laboratory of Information Processing, Communication and Networking, Hangzhou 310027, China
³Zhoushan Ocean Research Center, Zhoushan 316021, China

Corresponding author: Huifang Chen (chenhf@zju.edu.cn)

This work was supported in part by the Fund of National Natural Science Foundation of China under Grant 61671410, in part by the Science and Technology Department of Zhejiang Province under Grant LGG18F010005 and Grant 2018R52046, and in part by the Zhejiang Provincial Science and Technology under Grant LY14F010014.

ABSTRACT For the false data injection attack (FDIA) in the consensus-based distributed estimation, the information of weight matrix is critical for both attackers and defenders. In this paper, we study the impact of the weight matrix on the FDIA and the attack mitigation techniques. We first propose a learning-based cooperative FDIA strategy, where malicious nodes acquire the information of weight matrix cooperatively and launch the attack covertly. Two types of FDIA, the sudden FDIA and the dynamic FDIA, are considered to tamper the consensus result of the network to a pre-designed false value. Moreover, using the obtained information of weight matrix, a real-time surveillance and response mechanism is constructed to enhance the covertness of FDIA strategy. Since the surveillance and response mechanism can bypass existing FDIA detection methods, we further investigate the attack mitigation techniques against the learning-based cooperative FDIA. A real-time FDIA detection method and a reassessment mechanism with a punishment scheme are presented to resist the surveillance and response mechanism in the learning-based cooperative FDIA. Comprehensive simulation results verify that the attacker can obtain the information of weight matrix, and tamper the consensus result to a false value by launching the learning-based cooperative FDIA. And the real-time FDIA detection method can detect the malicious nodes efficiently and promptly.

INDEX TERMS Consensus-based distributed estimation, weight matrix, cooperative false data injection attack (FDIA), surveillance and response mechanism, real-time detection, reassessment mechanism.

I. INTRODUCTION

In the consensus-based distributed estimation, a collection of nodes with processing and communication functions are linked together through a connected topology. The nodes cooperate with each other through local interactions to estimate an unknown scalar or vector parameter in real-time [1], [2]. It is shown in [3] that the consensus update process in the consensus-based distributed estimation can be modeled as a linear transformation with weight matrix.

Owing to the open structure of the network, the consensus-based distributed estimation is vulnerable to various attacks, where false data injection attack (FDIA) is one of the most typical attacks [4], [5]. In FDIA, attackers access the network or compromise some nodes in the network in order to make them send some false data to their neighbors during the consensus update process. The goal of attackers is to tamper the consensus result to a false value or make the consensus unable to be achieved.

The defense mechanisms against FDIA in the consensus-based distributed estimation can be classified into two categories, resilient consensus algorithms [6]–[12] and FDIA detection mechanisms [13]–[17].

To tolerate a certain amount of faulty or malicious nodes in the network, some resilient consensus algorithms have been developed to make sure that the consensus result...
approximately converges to the true value \([6]–[12]\). In \([6]\), authors proposed a resilient consensus update rule, where each node updates its estimation as the average of maximum and minimum estimations among its neighbors, and some mobile nodes are deployed to identify the nodes violating the update rule. An approximate Byzantine consensus algorithm, named as weighted mean subsequence reduced (W-MSR) algorithm, is proposed in \([7]\). In the W-MSR algorithm, each node discards a certain amount of largest and smallest estimations received from its neighbors, and updates its estimation as the linear combination of the remainder estimations. The necessary and sufficient conditions for the W-MSR algorithm reaching an approximate consensus under various attack models are provided \([7]–[9]\). In \([10]\), a flag raising distributed estimation (FRDE) algorithm is proposed to perform resilient asymptotic consensus estimation and detect malicious nodes simultaneously. It is also proven in \([10]\) that if normal nodes performing the FRDE algorithm are connected and globally observable, their estimations are almost surely consistent. An iterative, consensus + innovations algorithm for the distributed estimation, named as constant weight saturated innovation update (CSIU) algorithm, is proposed in \([11]\). It is shown in \([11]\) that the estimations of normal nodes performing the CSIU algorithm will converge to the true value at an exponential rate if less than 30 percent of nodes are under attack. As the dynamics of the estimated parameters can be modeled as a linear discrete time-varying function, a resilient distributed filtering method based on the Krein space is proposed in \([12]\). In this method, an innovation analysis method and a projection technique in Krein space are utilized to calculate a resilient estimate, and the residual is used to detect the FDIA.

In \([13]–[17]\), efficient FDIA detection mechanisms have been proposed to detect malicious nodes in the network. In the stubborn FDIA model considered in \([13]\), malicious nodes do not receive any message from their neighbors, and update their estimations as the sum of an artificial noise and a constant false value. The temporal difference characteristic and the spatial difference characteristic are used to identify stubborn malicious nodes in \([13]\) and \([14]\), respectively. In \([15]–[17]\), a general FDIA model is considered, and two corresponding attack detection mechanisms are proposed. It is shown in \([15]\) that if normal nodes have full knowledge of weight matrix and the network topology satisfies some specific conditions, each normal node can estimate the real-time state vector using local observations, and the residual of estimation can be utilized to identify malicious nodes. In \([16]\) and \([17]\), a distributed filter is designed based on the theory of unknown input observability and the information of weight matrix. With the distributed filter, normal nodes can estimate the real-time state vector and detect malicious nodes among their neighbors.

To detect malicious nodes, an assumption that normal nodes have full knowledge of weight matrix is used in those FDIA detection methods \([15]–[17]\). However, the assumption seems unreasonable in a real-world large-scale distributed network. The reasons are two-fold. First, due to the dynamic nature of network, the dimension and value of weight matrix may change frequently. Second, normal nodes may not be willing to share their weight vectors with other nodes for security consideration. From the perspective of a normal node, it is unreasonable to use the real-time weight matrix to identify malicious nodes. However, from the perspective of a malicious node, obtaining the real-time information of weight matrix is benefit to the improvement of the covertness of its attack behavior. Although normal nodes with defense strategy can dynamically adjust their weight vectors when some suspicious situations occur, malicious nodes can learn about the defense strategies of normal nodes according to the real-time information of weight matrix, and try to bypass the defense mechanisms. Therefore, the information of weight matrix is significant for both attackers and defenders in the network.

Several algorithms are addressed to acquire the information of weight matrix \([18]\), \([19]\). In \([18]\), a novel algorithm is proposed to calculate the minimal polynomial of weight matrix in the network with time-invariant topology. A distributed consensus algorithm is proposed to calculate the Chebyshev polynomials of weight matrix using the second order difference equation \([19]\). However, the polynomials of weight matrix obtained in \([18]\) and \([19]\) cannot be utilized to construct a covert FDIA strategy or an effective FDIA detection mechanism. Therefore, in this paper, we investigate the method for obtaining the weight matrix, and the effect of weight matrix on the FDIA and the FDIA mitigation techniques.

Our main contributions are summarized as follows.

First, we propose a learning-based cooperative FDIA strategy against the consensus-based distributed estimation. In the proposed attack strategy, malicious nodes cooperate with each other to acquire the information of weight matrix. Two kinds of covert FDIA, the sudden FDIA and the dynamic FDIA, are launched using the obtained weight matrix. Under the presented FDIA, the consensus result of the network is tampered to a pre-designed false value.

Second, a surveillance and response mechanism is presented for malicious nodes to enhance the covertness of FDIA. In the proposed mechanism, malicious nodes keep surveilling the consensus update process of normal nodes using the obtained weight matrix. By switching between launching the FDIA and performing normal consensus update dynamically, the attacker with the proposed surveillance and response mechanism can bypass existing FDIA detection methods.

Third, in order to ensure the security of consensus-based distributed estimation, a real-time attack detection method against the learning-based cooperative FDIA is proposed. Moreover, to mitigate the effect of false-alarm of the FDIA detection system on the performance of consensus-based distributed estimation, a reassessment mechanism is constructed to reassess malicious nodes. With the embedded punishment scheme, the reassessment mechanism can resist the
surveillance and response mechanism in the learning-based cooperative FDIA strategy effectively.

The rest of the paper is organized as follows. Section II introduces the system model and the problem to be resolved in this work. In Section III, we propose a learning-based cooperative FDIA strategy against the consensus-based distributed estimation. The attack mitigation techniques against the FDIA are presented in Section IV. Simulation results and discussions are given in Section V. Finally, we conclude the paper in Section VI.

For clarity, we explain some notations used in this paper. Boldface lower and upper case letters denote column vector and matrix, respectively. For matrix \( A \) and column vector \( a \), the element at the \( i \)-th row and the \( j \)-th column in \( A \) is denoted as \( a_{ij} \), and the \( i \)-th element in \( a \) is denoted as \( a_i \). Superscript \((\cdot)^T\) represents the transpose operation. \( I \) is an identity matrix with appropriate dimension, \( 1 \) is a vector of 1s with appropriate dimension, and \( 0 \) is a null matrix or null vector with appropriate dimension. \( \text{diag}(a) \) denotes a diagonal matrix, where \( a_i \) is the \( i \)-th diagonal element. A normal distribution with mean \( \mu \) and variance \( \sigma^2 \) is denoted as \( \mathcal{N}(\mu, \sigma^2) \). The set of real numbers is denoted as \( \mathbb{R} \), and the set of positive integers is denoted as \( \mathbb{Z}^+ \).

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we introduce the network model for the consensus-based distributed estimation, the model of FDIA and some typical attack detection mechanisms. Then, the problem aiming to be resolved is formulated.

A. NETWORK MODEL

We consider a network consisting of \( N \) nodes with the same configuration. The topology of the network is modeled as a connected and undirected graph \( G = (\mathcal{V}, \mathcal{E}) \), where \( \mathcal{V} = \{1, 2, \ldots, N\} \) is the set of nodes, and \( \mathcal{E} \) is the set of communication links. If \((i,j) \in \mathcal{E}\), node \( i \) and node \( j \) are adjacent, and node \( i \) can receive information from node \( j \). The neighboring set of node \( i \) is defined as \( \mathcal{N}_i \triangleq \{j | j \in \mathcal{V}, (i,j) \in \mathcal{E}\} \).

In the consensus-based distributed estimation, an unknown scalar or vector parameter is first estimated by the nodes in the network separately. Then each node iteratively updates its estimation as a linear combination of estimations received from its neighbors and its own one. After multiple iterations, all the nodes in the network will come to a consensus [2].

It is assumed that a total \( K \) instances of consensus-based distributed estimation is implemented over network \( G \). In instance \( k \), \( 1 \leq k \leq K \), the initial estimation at node \( i \) is denoted as \( x_i^k(0) \). Then, each node exchanges the estimation with its neighbors, and updates its estimation using the estimations received from its neighbors. At iteration \( t, t \geq 1 \), the consensus update at node \( i \) can be expressed as

\[
x_i^k(t) = w_i x_i^k(t - 1) + \sum_{j \in \mathcal{N}_i} w_{ij} x_j^k(t - 1),
\]

where \( w_{ij} \) and \( w_i \) are the weights for the consensus update process, \( w_{ij} > 0, w_{ij} \geq 0, j \in \mathcal{N}_i \), and \( w_i + \sum_{j \in \mathcal{N}_i} w_{ij} = 1 \).

According to (1), the consensus update at node \( i \) can be rewritten as

\[
x_i^k(t) = w_i^T x_i^k(t - 1),
\]

and the consensus update in the network can be represented as

\[
x^k(t) = W x^k(t - 1),
\]

where \( x^k(t) \) is the \( N \times 1 \) state vector of the network at iteration \( t \) of instance \( k \), and \( x^k(t) = [x_{11}^k(t), x_{12}^k(t), \ldots, x_{1N}^k(t)]^T \); \( w_i^T \) is the \( 1 \times N \) weight vector at node \( i \). \( w_i^T = [w_{i1}, w_{i2}, \ldots, w_{iN}] \), \( w_{ij} \) is the element at the \( i \)-th row and the \( j \)-th column of \( W \), and \( W \) is the \( N \times N \) weight matrix. Since the row-sum of \( W \) equals to \( 1 \), \( W \) is a right stochastic matrix.

We assume that the total number of iterations implemented in each instance of consensus-based distributed estimation is \( T \).

B. ATTACK MODEL

It is assumed that some attackers exist in the network. In order to degrade the performance of consensus-based distributed estimation, attackers capture some unprotected nodes and make them be malicious. The malicious nodes will launch the FDIA; that is, malicious nodes send some pre-designed false data to their neighbors during the consensus update process. The goal of attackers is to tamper the consensus result to a false value or make the consensus unable to be achieved.

The FDIA can be classified into two types, the stubborn FDIA and the general FDIA. Since the stubborn FDIA problem and its mitigation techniques have been well studied in [13] and [14], we focus on the general FDIA problem in this paper.

In the model of general FDIA, malicious nodes inject some pre-designed false data into their states during the consensus update process. Let the set of malicious nodes be \( \mathcal{A} \triangleq \{a_1, a_2, \ldots, a_f\} \), where \( f \) denotes the number of malicious nodes in the network, and \( \mathcal{A} \subseteq \mathcal{V} \). Under the general FDIA, the consensus update in the network can be expressed as

\[
x^k(t) = W x^k(t - 1) + \sum_{i=1}^{f} u_{ai}^k(t) e_i,
\]

\[
= W x^k(t - 1) + B_A u_A^k(t),
\]

where \( u_{ai}^k(t) \) denotes the data injected by malicious node \( a_i \) at iteration \( t \) of instance \( k \); \( e_i \) denotes an \( N \times 1 \) unit vector, the \( i \)-th element in \( e_i \) is 1 and the other elements are 0s; \( B_A \triangleq [e_{a_1}, e_{a_2}, \ldots, e_{a_f}] \), and \( u_A^k(t) \triangleq [u_{a_1}^k(t), u_{a_2}^k(t), \ldots, u_{a_f}^k(t)]^T \).

C. MALICIOUS NODE DETECTION MECHANISM

In [15]–[17], two malicious node detection mechanisms against the general FDIA are addressed. It is proven in [15] that, if the network topology satisfies some specific conditions, normal nodes can estimate the real-time state vector...
\( \hat{x}^k (t) \) using local observations. In [16] and [17], a distributed filter is designed for normal nodes to estimate the real-time state vector \( x^k (t) \).

The residual of estimated real-time state vector can be expressed as

\[
\mathbf{r}^k (t) = \hat{x}^k (t) - Wx^k (t - 1), \tag{5}
\]

where \( \hat{x}^k (t) \) is the estimate of state vector \( x^k (t) \), and \( \mathbf{r}^k (t) \) is the \( N \times 1 \) residual vector.

Using \( \mathbf{r}^k (t) \), malicious nodes can be identified as follows. If element \( r^k_i (t) \), \( 1 \leq i \leq N \), satisfies \( |r^k_i (t)| > \eta \), node \( i \) is identified as malicious node, where \( \eta \) is a pre-defined decision threshold for the FDIA detection mechanisms.

### D. PROBLEM FORMULATION

For malicious node detection mechanisms proposed in [15]–[17], normal nodes should obtain full knowledge of weight matrix \( \mathbf{W} \). However, in a real large-scale network, this assumption is unreasonable. The reasons are described in Section I.

From the perspective of an attacker, it is significant to acquire the real-time information of weight matrix \( \mathbf{W} \) and utilize the acquired information to enhance the covertness of FDIA. Normal nodes with defense strategies can dynamically adjust the weight vectors if some suspicious situations occur. For example, while detecting some abnormal behaviors of neighbor node \( j \), node \( i \) can set \( w_{ij} \) to be 0 and neglect the message from node \( j \). By acquiring the real-time information of weight matrix \( \mathbf{W} \), malicious nodes can learn about the defense strategies of normal nodes, and try to bypass the defense mechanisms.

Therefore, it is significant for both attackers and defenders to acquire the information of weight matrix. In this work, we first investigate the method for acquiring weight matrix, and then study the effect of the information of weight matrix on the FDIA and the FDIA mitigation techniques.

### III. LEARNING-BASED COOPERATIVE FDIA

In this section, a learning-based cooperative FDIA strategy against the consensus-based distributed estimation is presented.

We assume that malicious nodes in the network consist of an injector and several listeners. The proposed learning-based cooperative FDIA strategy consists of two stages. In the first stage, all the malicious nodes perform the normal consensus update without launching attack. Meanwhile, the listeners send their states and the states received from their neighbors to the injector. With the information of all the malicious nodes, the injector can acquire the information of weight matrix \( \mathbf{W} \). In the second stage, the injector utilizes the obtained information of weight matrix to launch two kinds of covert FDIA, the sudden FDIA and the dynamic FDIA. In addition, a real-time surveillance and response mechanism is constructed for malicious nodes to enhance the covertness of attack.

### A. STAGE ONE: LISTENING AND LEARNING

In order to simplify the description, we first consider an ideal scenario, and then extend it to a general scenario.

According to the distribution of malicious nodes in network \( \mathcal{G} \), we define two concepts as follows.

**Definition 1:** The set of observable nodes to \( \mathcal{A} \), \( \mathcal{V}_A \), is defined as \( \mathcal{V}_A \overset{\Delta}{=} \left( \bigcup_{i \in A} \mathcal{N}_i \right) \cup \mathcal{A} \).

**Definition 2:** If and only if \( \mathcal{V}_A = \mathcal{V} \), network \( \mathcal{G} \) is fully observable to \( \mathcal{A} \); otherwise, \( \mathcal{G} \) is partially observable to \( \mathcal{A} \).

1) **SCENARIO 1: NETWORK \( \mathcal{G} \) IS FULLY OBSERVABLE TO \( \mathcal{A} \)**

In this ideal scenario, the injector can obtain the real-time state vector \( \hat{x}^k(t) \) since all the listeners send their states and the states received from their neighbors to the injector. Using the state vectors from iteration 0 to \( N \), the injector can construct the following equation,

\[
\mathbf{X}^k (1 : N) = \mathbf{Wx}^k (0 : N - 1), \tag{6}
\]

where \( \mathbf{X}^k (0 : N - 1) \overset{\Delta}{=} [\hat{x}^k (0), \hat{x}^k (1), \ldots, \hat{x}^k (N - 1)] \) and \( \mathbf{X}^k (1 : N) \overset{\Delta}{=} [\hat{x}^k (1), \hat{x}^k (2), \ldots, \hat{x}^k (N)] \) are \( N \times N \) matrices.

According to [15]–[17], [20] and [21], weight matrix \( \mathbf{W} \) is a “structured matrix”. Hence, \( \mathbf{X}^k (0 : N - 1) \) and \( \mathbf{X}^k (1 : N) \) are nonsingular matrices for almost any initial state vector \( \hat{x}^k (0) \). That is, the set of initial state vector making \( \mathbf{X}^k (0 : N - 1) \) and \( \mathbf{X}^k (1 : N) \) be singular lies on an algebraic hyper-surface of parameters space \( \mathbb{R}^{N \times 1} \). Hence, at iteration \( N \), the weight matrix can be estimated by the injector as

\[
\hat{\mathbf{W}} = \mathbf{X}^k (1 : N) \left[ \mathbf{X}^k (0 : N - 1) \right]^{-1}, \tag{7}
\]

In real applications, in order to improve the robustness of estimation to background noise and transmission error, some redundant observations should be added. The state vectors from iteration 0 to \( N' \) should be adopted for the estimation, where \( N' = N + \Delta N \), \( \Delta N \) is the redundancy of observations, and \( \Delta N \geq 0 \). The improved estimation process can be expressed as

\[
\hat{\mathbf{W}} = \mathbf{X}^k (1 : N') \left[ \mathbf{X}^k (0 : N' - 1) \right]^\dagger, \tag{8}
\]

where \( (\cdot)^\dagger \) denotes the pseudo-inverse of a matrix.

2) **SCENARIO 2: NETWORK \( \mathcal{G} \) IS PARTIALLY OBSERVABLE TO \( \mathcal{A} \)**

Since network \( \mathcal{G} \) is partially observable to \( \mathcal{A} \), only the states of observable nodes within \( \mathcal{V}_A \) are available to the injector. In order to conveniently describe the listening and learning process in scenario 2, we define the concept of the vulnerable node set for \( \mathcal{A} \) using the observable node set \( \mathcal{V}_A \).

**Definition 3:** The vulnerable node set for \( \mathcal{A} \) is defined as \( \mathcal{V}_V \overset{\Delta}{=} \{ i | i \in \mathcal{V}_A \backslash \mathcal{A}, \mathcal{N}_i \subset \mathcal{V}_A \} \).

For each vulnerable node \( v, v \in \mathcal{V}_V \), the states of node \( v \) and its neighbors are available to the injector. The injector
can estimate the weight vector of vulnerable node \( v \) using (8). However, since the injector does not have full information about the network topology, the vulnerable node set is still unknown. Hence, the injector should first search for the vulnerable nodes within set \( \mathcal{V}_A \). 

For each vulnerable node \( v \), \( v \in \mathcal{V}_v \), the states of node \( v \) at each iteration \( t \), \( x^k_v(t) \), are the linear combination of states of nodes within \( \mathcal{V}_A \) at iteration \( t - 1 \). Therefore, the state vector of vulnerable node \( v \), \( [x^k_v(1 : N)]^T = [x^k_v(1), x^k_v(2), \ldots, x^k_v(N)]^T \), completely lies in \( \mathcal{R}([x^k_{\mathcal{V}_A}(0 : N - 1)]^T) \), where \( \mathcal{R}([x^k_{\mathcal{V}_A}(0 : N - 1)]^T) \) denotes the column space of matrix \( [x^k_{\mathcal{V}_A}(0 : N - 1)]^T \).

\[
[x^k_{\mathcal{V}_A}(0 : N - 1)]^T = \begin{bmatrix}
{x^k_{v1}(0 : N - 1)}^T, & {x^k_{v2}(0 : N - 1)}^T, & \ldots, & {x^k_{v|\mathcal{V}_A|}(0 : N - 1)}^T
\end{bmatrix}^T, \quad v_i \in \mathcal{V}_A.
\]

**Proposition 1:** Let \( A \) be a full column rank matrix. The orthogonal projectors onto \( \mathcal{R}(A) \) and \( \mathcal{R}(A)^\perp \) are \( P_{\mathcal{R}(A)} = A(A^T A)^{-1}A^T \) and \( P_{\mathcal{R}(A)^\perp} = I - A(A^T A)^{-1}A^T \), respectively, where \( \mathcal{R}(A) \) denotes the column space of \( A \), and \( \mathcal{R}(A)^\perp \) denotes the orthogonal complement of \( \mathcal{R}(A) \).

**Proof:** See [22, p. 430] or [23, p. 175].

Since \( X^f(0 : N - 1) \) is a nonsingular matrix, the state vector of invulnerable node \( j, j \notin \mathcal{V}_v \), does not lie in \( \mathcal{R}([x^k_{\mathcal{V}_A}(0 : N - 1)]^T) \). Therefore, according to Proposition 1, the set of vulnerable nodes can be identified as \( \mathcal{V}_V = \{ v \in \mathcal{V}_A \setminus \mathcal{A}, P_{\mathcal{R}(A)}[x^k_{\mathcal{V}_A}(0 : N - 1)]^T = 0 \} \), where \( P_{\mathcal{R}(A)}[x^k_{\mathcal{V}_A}(0 : N - 1)]^T \) is the orthogonal projector onto \( \mathcal{R}([x^k_{\mathcal{V}_A}(0 : N - 1)]^T)^\perp \), which can be expressed as

\[
P_{\mathcal{R}(A)}[x^k_{\mathcal{V}_A}(0 : N - 1)]^T = I - [x^k_{\mathcal{V}_A}(0 : N - 1)]^T \begin{bmatrix}
[x^k_{\mathcal{V}_A}(0 : N - 1)]^T \cdot [x^k_{\mathcal{V}_A}(0 : N - 1)]^T
\end{bmatrix}^{-1} [x^k_{\mathcal{V}_A}(0 : N - 1)]^T.
\]

When the injector has no information of the number of nodes in the network, an upper bound of the number of nodes, \( N \), can be used as the dimension of the state vector. If the redundancy of observations \( \Delta N, \Delta N = N - N \), is large enough, the vulnerable nodes can be identified correctly.

Moreover, due to the limited calculation precision, the orthogonal projector of state vector of a vulnerable node \( v \) on \( \mathcal{R}([x^k_{\mathcal{V}_A}(0 : N - 1)]^T)^\perp \), \( P_{\mathcal{R}(\mathcal{X}^f_{\mathcal{V}_A}(0 : N - 1))}^\perp \), may not exactly equal to 0. Therefore, the set of vulnerable nodes can be identified as

\[
\mathcal{V}_V = \left\{ v \mid \left| P_{\mathcal{R}(\mathcal{X}^f_{\mathcal{V}_A}(0 : N - 1))} \right| \left| [x^k_v(1 : N)]^T \right| < \eta_N, \quad v \in \mathcal{V}_A \setminus \mathcal{A} \right\},
\]

where \( | \cdot | \) denotes the 1-norm of a vector; \( \eta_N \) is the decision threshold for identifying vulnerable nodes, and

\[
\eta_N = \frac{\rho}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \left| P_{\mathcal{R}(\mathcal{X}^f_{\mathcal{V}_A}(0 : N - 1))} \right| \left| [x^k_v(1 : N)]^T \right|, \quad \rho > 1.
\]

After obtaining the set of vulnerable nodes, the injector can estimate the weight vector of vulnerable node \( v, v \in \mathcal{V}_V \), as

\[
\hat{\mathbf{w}}_v, _{\mathcal{V}_A} = \left( \left[ \mathbf{x}^f_{\mathcal{V}_A}(0 : N - 1) \right]^T \right) \left( \left[ \mathbf{x}_v^k (1 : N) \right]^T \right),
\]

where \( \hat{\mathbf{w}}_v, _{\mathcal{V}_A} \) is the \( |\mathcal{V}_A| \times 1 \) estimated weight vector of node \( v \).

From \( \hat{\mathbf{w}}_v, _{\mathcal{V}_A} \), the injector obtains not only the information of the neighboring set of node \( v \) (i.e., nodes correspond to non-zero elements in \( \hat{\mathbf{w}}_v, _{\mathcal{V}_A} \)), but also the detailed information of consensus update process of node \( v \).

**B. STAGE TWO: FALSE DATA INJECTION**

With the weight matrix or weight vectors acquired in stage one, the injector can launch a covert FDIA. The goal of the injector is tampering the final consensus result of the network to a false value \( G^k \), as well as remaining unsuspicous to normal nodes.

Here, we assume that the weight matrix remains unchanged if no attack is launched in the network. However, if some malicious nodes launch the FDIA, normal nodes with defense strategies can adjust their weight vectors to resist the attack. With these assumptions, the injector can launch two kinds of covert FDIA, the sudden FDIA and the dynamic FDIA.

1) **SUDDEN FDIA STRATEGY**

The sudden FDIA is launched in scenario 1, where network \( G \) is fully observable to \( A \).

According to Subsection III-A1, the injector can obtain the estimate of weight matrix, \( \hat{\mathbf{W}} \). Let row vector \( \mathbf{v}^T \) be the left eigenvector of \( \hat{\mathbf{W}} \) associated with eigenvalue 1, and \( \mathbf{v}^T \hat{\mathbf{W}} = \mathbf{v}^T \). The elements of \( \mathbf{v}^T \) also indicate the contribution of the nodes’ states to the consensus result, and \( \lim_{t \to \infty} \mathbf{x}^k(t) = \mathbf{v}^T \mathbf{x}^k(N') + 1 \) \( \mathbf{1} \). The injector, denoted as \( a_A \), can tamper its own state at iteration \( N' + 1 \) as

\[
\mathbf{x}^k_{a_A}(N' + 1) = [\hat{G}^k - \hat{\mathbf{W}}_{a_A} \mathbf{v}^T \mathbf{x}^k(N')] / \nu_{a_A},
\]

where \( \nu_{a_A} \) is the \( a_A \)-th element of \( \mathbf{v}^T ; \hat{\mathbf{W}}_{a_A} \) is a \( 1 \times N \) vector, the \( a_A \)-th element of \( \hat{\mathbf{W}}_{a_A} \) is 0 and other elements are the same as those in \( \mathbf{v}^T \).

After iteration \( N' + 1 \), malicious nodes keep performing the normal consensus update without launching attack. If the
weight matrix remains unchanged after iteration $N' + 1$, the final consensus result of the network can be expressed as
\[
\lim_{t \to \infty} x^k(t) = v^T x^k(N' + 1) 1
\]
\[
= v_{\alpha A} x_{\alpha A}^k (N' + 1) + \hat{V}_{\alpha A}^{t} \hat{W} x^k(N')
\]
\[
= G^k 1,
\]
which means the goal of the injector is achieved.

In a real network environment, due to the equipment malfunction and transmission error, a normal node may occasionally send some abnormal data to its neighbors. Since the injector only inject false data at iteration $N' + 1$, it imitates the behavior of a normal node suffered from burst equipment malfunction or transmission error. Hence, the veracity of the proposed sudden FDIA strategy is ensured.

However, there are two shortcomings for the sudden FDIA strategy. First, the sudden FDIA can only be launched in scenario 1, which is ideal for an injector with limited resource. Second, if some normal nodes dynamically adjust the weight vectors to resist the attack, the consensus result will deviate from the goal of the injector, $G^k$. To overcome these deficiencies, we further develop a dynamic FDIA strategy.

2) DYNAMIC FDIA STRATEGY
In the dynamic FDIA, the injector utilizes the obtained weight matrix or weight vectors to inject a pre-designed false data $u^k_{\alpha A} (t)$ to its state at each iteration. The consensus update at the injector can be expressed as
\[
x^k_{\alpha A} (t) = w^T_{\alpha A} x^k (t - 1) + u^k_{\alpha A} (t),
\]
where
\[
u^k_{\alpha A} (t) = \begin{cases} 0, & 0 \leq t \leq N', \\ \mu A \hat{w}_{\alpha V, \alpha A} \left[ G^k - \hat{x}^k_{\alpha A} (t - 1) \right], & N' + 1 \leq t \leq T, \end{cases}
\]
\[
\hat{x}^k_{\alpha A} (t - 1) = \hat{v}^T x^k (t - 1) \text{ in scenario 1, and } \hat{x}^k_{\alpha A} (t - 1) = \sum_{i \in \alpha V, \alpha A} x^k_i (t - 1) / |\alpha V, \alpha A| \text{ in scenario 2.}
\]
\[
\hat{w}_{\alpha V, \alpha A} = v_{\alpha A} \text{ in scenario 1, and } \hat{w}_{\alpha V, \alpha A} = \sum_{v \in \alpha V, \alpha A} w_{v, \alpha A} / |\alpha V, \alpha A| \text{ in scenario 2, where } \alpha V, \alpha A \triangleq \alpha V \cap N_{\alpha A}. \text{ If } \alpha V, \alpha A = \emptyset, the injector should be transformed into a listener. Meanwhile, if there exists a listener adjacent to at least one node within $\alpha V$, it is assigned as a new injector. }\mu A \text{ is an attack parameter determined by the injector, and } \mu A > 0. \text{ The larger the value of } \mu A, \text{ the more quickly all of the nodes converge to } G^k. \text{ However, if the value of } \mu A \text{ is too large, the injector can be detected by normal nodes with a high probability.}

In the proposed dynamic FDIA strategy, the reason to make the injected data $u^k_{\alpha A} (t)$ be proportional to $\hat{w}_{\alpha V, \alpha A}$ is that, if the value of $\hat{w}_{\alpha V, \alpha A}$ gets smaller, the nodes within $\alpha V, \alpha A$ begin to distrust the injector, and the injector should scale down the injected data $u^k_{\alpha A} (t)$.

C. REAL-TIME SURVEILLANCE AND RESPONSE MECHANISM
Since normal nodes with defense strategies can resist the FDIA by dynamically adjusting their weight vectors during the consensus update process, malicious nodes should surveil the behavior of normal nodes. Once malicious nodes observe some normal nodes adjusting their weight vectors, some corresponding response mechanisms can be implemented to enhance the veracity of attack.

After launching the sudden FDIA, malicious nodes keep surveilling the behavior of normal nodes. The surveillance of normal node $v$ utilizes the consistency of states among observable nodes at iteration $t$ and the state of node $v$ at iteration $t + 1$. That is, the adjustment of the weight vector of normal node $v$ can be declared if the magnitude of $x^k_v (t + 1) - \hat{w}^T_v x^k (t)$ exceeds a pre-defined threshold, where $\hat{w}^T_v$ is the $v$-th row of $\hat{W}$. Furthermore, if the weights $w_{v, \alpha A}$ and $w_{v, v}$ of normal node $v$ are adjusted to $w_{v, \alpha A}$ and $w_{v, v}$ at iteration $t + 1$, and $w_{v, \alpha A} + w_{v, v} = w_{v, \alpha A} + w_{\alpha V, v}$, we have $E \left[ x^k_v (t + 1) - \hat{w}^T_v x^k (t) \right] = \Delta w_{v, \alpha A} x_{\alpha A}^k (t), \Delta w_{v, \alpha A} = w_{v, \alpha A} - w_{\alpha A}, \alpha A \text{ and } x_{\alpha A}^k (t) = x^k_v (t) - \bar{x}^k_v (t)$. In order to evaluate the effectiveness of FDIA, the injector should determine whether $\Delta w_{v, \alpha A} \leq \text{less than a threshold } \phi_S \text{ or not, and } \bar{w}_{v, \alpha A} \leq \phi_S \leq 0$. Therefore, the surveillance of normal node $v, v \in \alpha V, \alpha A$, can be expressed as
\[
\text{sgn} \left[ \hat{d}^k_{\alpha A, v} (t) \right] \cdot \left[ x^k_v (t + 1) - \hat{w}^T_v x^k (t) \right] \leq D_0^+ \leq D_0^-, \phi_S, \hat{d}^k_{\alpha A, v} (t),
\]
where $\text{sgn} (x)$ is a modified sign function, and $\text{sgn} (x) = \begin{cases} 1, & x \geq 0; \\ -1, & x < 0; \end{cases}$ $D_0^+$ denotes the hypothesis that node $v$ is adjusting its weight vector; $D_0^-$ denotes the hypothesis that node $v$ keeps its weight vector unchanged. Parameter $\phi_S$ is chosen by the injector. The larger the value of parameter $\phi_S$, the more sensitive the surveillance mechanism will become. In an extreme case of $\phi_S = 0$, $D_1^-$ is declared once the weight $w_{v, \alpha A}$ decreases. On the other hand, $\phi_S = \bar{w}_{v, \alpha A}, D_1^-$ is declared only if the weight $w_{v, \alpha A}$ is set to be 0.

The goal of the injector, $G^k$, can be achieved only if the weight matrix remains unchanged after the sudden FDIA. Therefore, once some normal nodes are found adjusting their weight vectors, the injector will return to stage one to acquire the updated information of weight matrix, and then launch the dynamic FDIA.

After launching the dynamic FDIA, malicious nodes keep surveilling the behavior of vulnerable node $v, v \in \alpha V$, as
\[
\text{sgn} \left[ \hat{d}^k_{\alpha A, v} (t) \right] \cdot \left[ x^k_v (t + 1) - \hat{w}^T_v x^k_{\alpha A} (t) \right] \leq D_0^+ \leq D_0^- \leq \phi_S, \hat{d}^k_{\alpha A, v} (t),
\]
where $\hat{w}_v, \alpha A$ is the estimated weight vector of node $v$.

Next, the response mechanism is constructed according to the result of the surveillance process in (15). The injector can adopt two kinds of response mechanisms, the
conservative response mechanism and the radical response mechanism.

In the conservative response mechanism, once observing at least one vulnerable node within \( \mathcal{V}_v \) adjusting its weight vector, the injector stops launching the dynamic FDIA immediately, and returns to stage one to acquire the updated information of weight matrix. Only if the updated information of weight matrix shows that \( \hat{w}_{v,a} \neq 0 \) for all \( v \in \mathcal{V}_v,A \), the injector continues to launch the dynamic FDIA. Furthermore, parameter \( \psi_S \) can be increased to improve the sensitivity of the real-time surveillance mechanism.

In the radical response mechanism, the injector stops launching the dynamic FDIA and returns to stage one if the surveillance results are \( D_1^k \) for all \( v \in \mathcal{V}_v,A \). If the updated information of weight matrix shows that at least one node \( v \) within \( \mathcal{V}_v,A \) satisfies \( \hat{w}_{v,a} \neq 0 \), the injector keeps on launching the dynamic FDIA. Furthermore, parameter \( \psi_S \) can be decreased to reduce the sensitivity of the real-time surveillance mechanism.

### IV. ATTACK MITIGATION TECHNIQUES

In order to enhance the security of consensus-based distributed estimation, we further study the attack mitigation techniques. It is assumed that normal nodes do not know whether the other nodes are normal or not. In this case, each normal node resists the FDIA with local information separately. To defend against the learning-based cooperative FDIA, a real-time attack detection method using local observations is proposed for normal nodes. Moreover, to mitigate the effect of false-alarm of the real-time FDIA detection system on the estimation performance, we address a reassessment mechanism to reassess malicious nodes.

#### A. REAL-TIME FDIA DETECTION METHOD

Let \( \Psi(t) \) be the one-shot statistic at time \( t \). For a traditional real-time event detection system [24], [25], if the event happens at time \( t \), the expectation of \( \Psi(t) \) satisfies \( E[\Psi(t)] > 0 \); otherwise, \( E[\Psi(t)] \leq 0 \).

The real-time statistic at time \( t \), \( S(t) \), is updated as \( S(t) = \max(0, S(t-1) + \Psi(t)) \). If the event does not happen, \( S(t) \) keeps hovering around 0. If the event happens after time \( \tau \), \( S(t) \) rises rapidly after time \( \tau \).

For a real-time detection method against the learning-based cooperative FDIA, it is important to construct an applicable one-shot statistic satisfying the following conditions. For normal node \( i \) and its neighbor node \( j \), the expectation of one-shot statistic, \( \Psi_{ij}^k(t) \), should satisfy two conditions, namely \( E[\Psi_{ij}^k(t)|\mathcal{H}_0^k] \leq 0 \) and \( E[\Psi_{ij}^k(t)|\mathcal{H}_1^k] > 0 \), where \( \mathcal{H}_0^k \) denotes the hypothesis that node \( j \) does not launch the FDIA, and \( \mathcal{H}_1^k \) denotes the hypothesis that node \( j \) launches the FDIA. Based on the one-shot statistic, the real-time statistic can be constructed as

$$ S_{ij}^k(t) = \max \left\{ 0, S_{ij}^k(t-1) + \Psi_{ij}^k(t) \right\}. \quad (16) $$

The quickest stopping time for node \( i \) declaring that node \( j \) is launching the FDIA can be expressed as

$$ T_{ij}(h) = \inf \left\{ t | S_{ij}^k(t) > h \right\}. \quad (17) $$

where \( \inf \{ \cdot \} \) denotes the infimum operation, and \( h \) is a predefined decision threshold for the real-time detection method.

Once the occurrence of FDIA is declared for node \( j \), node \( i \) sets \( w_{ij} \) to be 0 and neglects the message from node \( j \) after \( T_{ij}(h) \).

In order to construct an applicable one-shot statistic, we define the spatial-difference-based statistic for node \( i \) and its neighbor node \( j \) at each iteration as \( \lambda_{ij}^k(t) = \lambda_i^k(t) - \lambda_j^k(t) \). Using \( \lambda_{ij}^k(t) \), the one-shot statistic, \( \psi_{ij}^k(t) \), can be defined as

$$ \psi_{ij}^k(t) = |s_{ij}^k(t)| - \Lambda_{ij}^k(t), \quad (18) $$

where \( \Lambda_{ij}^k(t) \) is an upper bound of the magnitude of spatial-difference-based statistic when node \( j \) does not launch the FDIA.

If the malicious nodes do not launch the FDIA, we have \( x^k(t) = Wx^k(t-1) = [W]x^k(t-2) = \cdots = [W]^kx^k(0) \), the spatial-difference-based statistic can be rewritten as

$$ s_{ij}^k(t) = (e_j - e_i)^T x^k(t) = (e_j - e_i)^T [W]^kx^k(0). \quad (19) $$

The eigenvalue decomposition of \([W]^k\), \( t \geq 1 \), can be expressed as

$$ [W]^k = \Sigma \lambda^k \Sigma^T = \sum_{n=1}^{N} \lambda_n^k v_n u_n^T, \quad (20) $$

where \( \lambda_n \) denotes the \( n \)-th eigenvalue of \( W \), \( 1 \leq n \leq N \); \( \Sigma \) denotes an \( N \times N \) diagonal matrix, and \( \lambda_n \) is the \( n \)-th diagonal element of \( \Sigma \); \( v_n \) and \( u_n \) denote the \( n \)-th column of matrices \( V \) and \( U \), respectively. \( v_n \) is the right eigenvector of \( W \) associated with eigenvalue \( \lambda_n \), and \( U^T = \Sigma^{-1} \).

Without loss of generality, we assume that \( |\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_N| \). According to the Gerschgorin circle theorem [26], since the elements of matrix \( W \) are not less than 0 and \( W1 = 1 \), we have \( \lambda_1 = 1, v_1 = 1/\sqrt{N} \) and \( |\lambda_n| \leq 1, 2 \leq n \leq N \). Substituting (20) into (19), the spatial-difference-based statistic can be rewritten as

$$ s_{ij}^k(t) = e_j^T \left( \sum_{n=1}^{N} \lambda_n^k v_n u_n^T \right) x^k(0) $$

$$ = \sum_{n=1}^{N} \lambda_n^k \left[ e_j^T v_n u_n^T x^k(0) \right] = \sum_{n=1}^{N} \lambda_n^k c_n, \quad (21) $$

where \( e_j \triangleq e_j - e_i \) and \( c_n \triangleq e_j^T v_n u_n^T x^k(0) \).

Since \( e_j^T 1 = 0 \) and \( v_1 = 1/\sqrt{N} \), we have \( c_1 = 0 \). Hence, \( s_{ij}^k(t) = \sum_{n=2}^{N} \lambda_n^k c_n \).

Let \( \left[ s_{ij}^k(1: N-1) \right]^T \triangleq \left[ s_{ij}^k(1), s_{ij}^k(2), \ldots, s_{ij}^k(N-1) \right]^T \), we have

$$ \left[ s_{ij}^k(1: N-1) \right]^T = V \alpha \delta^e, \quad (22) $$
where \( c' = [c_2, c_3, \ldots, c_N]^T \), Vad(\( \lambda' \)) is an \((N-1) \times (N-1)\) Vandermonde matrix, the first row of Vad(\( \lambda' \)) is \( \lambda'^T = [\lambda_2, \lambda_3, \ldots, \lambda_N] \), the \( i \)-th row of Vad(\( \lambda' \)) is \( \lambda'^T[diag(\lambda')]^{i-1} \), and diag(\( \lambda' \)) is an \((N-1) \times (N-1)\) diagonal matrix with diagonal elements \( \{\lambda_n\}_{n=2}^{N} \).

It is assumed that the values in \( \lambda' \) are non-zero and different from each other. It is proven in [26] that matrix Vad(\( \lambda' \)) is nonsingular if the values in \( \lambda' \) are non-zero and different from each other. Therefore, row vector \( [\lambda_2^N, \lambda_3^N, \ldots, \lambda_N^N] \) can be expressed as a unique linear combination of the rows of Vad(\( \lambda' \)), and a unique \((N-1) \times 1\) vector \( d \) exists and satisfies

\[
\begin{bmatrix} \lambda_2^N, \lambda_3^N, \ldots, \lambda_N^N \end{bmatrix} = d\text{Vad}(\lambda').
\] (23)

The left side of (23) can be rewritten as

\[
[\lambda_2^N, \lambda_3^N, \ldots, \lambda_N^N] = \lambda'^T\text{diag}(\lambda')^{N-1}
\]

and the right side of (23) can be rewritten as

\[
d^T\text{Vad}(\lambda') = \sum_{n=1}^{N-1} d_n\lambda'^T\text{diag}(\lambda')^{n-1} = \lambda'^T\sum_{n=1}^{N-1} d_n[\text{diag}(\lambda')]^{n-1}.
\]

Since matrix \( [\text{diag}(\lambda')]^i \), \( i \geq 0 \), is nonsingular, the annihilating polynomial of \( \text{diag}(\lambda') \) can be expressed as

\[
[\text{diag}(\lambda')]^{N-1} - \sum_{n=1}^{N-1} d_n[\text{diag}(\lambda')]^{n-1} = 0.
\]

According to the Cayley-Hamilton theorem [27], the characteristic polynomial of \( \text{diag}(\lambda') \), \( p(x) = \text{det}[\text{diag}(\lambda') - xI] = \prod_{n=2}^{N} (\lambda_n - x) \), is also the annihilating polynomial of \( \text{diag}(\lambda') \). Hence, \( p[\text{diag}(\lambda')] = \prod_{n=2}^{N} [\lambda_n I - \text{diag}(\lambda')] = 0 \). The elements of \( d \) in (23), \( d_n \), can be iteratively calculated using the LeVerrier-Faddeev algorithm [28] as

\[
M_n = \text{diag}(\lambda') M_{n-1} + d_{N-n-1}I,
\]

\[
d_{N-n} = -\frac{1}{n}\text{tr}[\text{diag}(\lambda') M_n], \quad 1 \leq n \leq N - 1.
\] (24)

where \( M_0 \) is an \((N-1) \times (N-1)\) null matrix, \( d_N = 1 \), and \( \text{tr}(\cdot) \) is the trace of a square matrix.

According to (19)-(23), we have

\[
[\lambda_2^{N+\tau}, \lambda_3^{N+\tau}, \ldots, \lambda_N^{N+\tau}] = [\lambda_2^N, \lambda_3^N, \ldots, \lambda_N^N][\text{diag}(\lambda')]^\tau = d^T\text{Vad}(\lambda')[\text{diag}(\lambda')]^\tau, \quad \tau \geq 0.
\] (25)

Since \( s_j^k(t) = [\lambda_2^j, \lambda_3^j, \ldots, \lambda_N^j] c' \), \( t \geq 1 \), we have

\[
s_j^k(N + \tau) = d^T\text{Vad}(\lambda)[\text{diag}(\lambda')]^\tau c', \quad \tau \geq 0.
\]

(26)

**Remark 1:** From (26), one finds that if the malicious nodes do not launch the FDIA, the spatial-difference-based statistic at iteration \( N + \tau \) can be represented as a linear combination of spatial-difference-based statistics from iteration \( \tau + 1 \) to \( N + \tau - 1 \). Taking advantage of this property, the upper bound of the magnitude of spatial-difference-based statistic in (18) can be constructed to satisfy \( E[\Psi_j^k(t) | H_0^t] \leq 0 \).

Since eigenvalues \( \{\lambda_n\}_{n=2}^{N} \) are unknown to normal nodes, the elements of \( d \) still cannot be directly calculated with (24). However, since the eigenvalues in \( \lambda' \) are different from each other, some eigenvalues with large magnitude can be asymptotically estimated by normal nodes with local observations. Here, two variables, \( s_j^k(t) \) and \( s_j^\cdot(t) \), are defined as \( s_j^k(t) = \lambda^k_j(t) s_j^k(t - 2) - [s_j^k(t - 1)]^2 \) and \( s_j^\cdot(t) = \lambda^\cdot_j(t) s_j^\cdot(t - 2) - [s_j^\cdot(t - 1)]^2 \). The estimation of eigenvalues can be summarized in Theorem 1.

**Theorem 1:** The estimates of eigenvalues \( \lambda_2, \lambda_3, \text{ and } \lambda_4 \) can be sequentially calculated by (27a)-(27c), whereas other eigenvalues \( \lambda_i \), \( i \geq 5 \), cannot be uniquely determined.

\[
\hat{\lambda}_2 = \frac{s_j^k(T)}{s_j^k(T - 1)}, \quad s_j^\cdot(T - 1), \quad \hat{\lambda}_3 = \frac{s_j^\cdot(T)}{\hat{\lambda}_2 s_j^\cdot(T - 1)}, \quad \hat{\lambda}_4 = \frac{1}{\hat{\lambda}_2^2 + \hat{\lambda}_3^2} \frac{s_j^\cdot(T)}{\hat{\lambda}_2 s_j^\cdot(T - 1)},
\] (27a)

(27b)

(27c)

where \( T \) is the total number of iterations implemented in each instance of consensus-based distributed estimation.

The proof of Theorem 1 is given in Appendix.

As shown in (27a), (27b), and (27c), the calculation of \( \hat{\lambda}_2, \hat{\lambda}_3, \text{ and } \hat{\lambda}_4 \) is completely decentralized, which means that each normal node can obtain the estimations of \( \hat{\lambda}_2, \hat{\lambda}_3, \text{ and } \hat{\lambda}_4 \) with local information separately. According to Theorem 1, the rest of the eigenvalues, \( \{\lambda_n\}_{n=5}^{N} \), cannot be estimated. Hence, the spatial-difference-based statistic cannot be predicted with (26). To deal with this issue, we construct an optimization problem to obtain an upper bound of the magnitude of spatial-difference-based statistic satisfying \( E[\Psi_j^k(t) | H_0^t] \leq 0 \). According to (24)-(27), the real-time upper bound of the magnitude of spatial-difference-based statistic can be constructed by solving the optimization problem as

\[
\Lambda_j^k(t) = \max_{\lambda_j^k(5 \leq n \leq N)} \left[ \sum_{n=1}^{N-1} d_n s_j^k(t - n) \right], \quad t \geq N - 1, \quad
\]

s. t. \( \lambda^k_2 = \hat{\lambda}_2, \lambda^\cdot_3 = \hat{\lambda}_3, \lambda^\cdot_4 = \hat{\lambda}_4, \)

\[
|\lambda^\cdot_5| < |\lambda^\cdot_4|, \quad |\lambda^\cdot_n| < |\lambda^\cdot_{n-1}|, \quad 6 \leq n \leq N,
\]

(P.1)

where variables \( \{\lambda^\cdot_n\}_{n=3}^{N} \) are used to calculate \( [d_n]_{n=1}^{N-1} \) with the LeVerrier-Faddeev algorithm in (24). Since the eigenvalues in \( \{\lambda_n\}_{n=5}^{N} \) should satisfy the inequality constraints \( |\lambda_n| < |\lambda_{n-1}| (5 \leq n \leq N) \), the optimal \( \Lambda_j^k(t) \) obtained in problem (P.1) must be larger than \( |s_j^k(t)| \), which satisfies

\[
E[\Psi_j^k(t) | H_0^t] \leq 0.
\]
In order to simplify the calculation for the upper bound of the magnitude of spatial-difference-based statistic, the inequality constrain conditions in (P.1) can be relaxed as \( |\lambda'_k| \leq |\hat{\lambda}_4|, \ k \geq 3 \). And the problem in (P.1) can be transformed into

\[
\Lambda^k_{ij}(t) = \max_{\lambda'_{n}(2 \leq n \leq N)} \left| \sum_{n=1}^{N-1} d_n s_{ij}^k(t - n) \right|, \ t \geq N - 1, \ \\
\text{s. t. } \lambda'_2 = \hat{\lambda}_2, \lambda'_3 = \hat{\lambda}_3, \lambda'_4 = \hat{\lambda}_4, \ \\
\left| \lambda'_{n} \right| \leq |\hat{\lambda}_4|, \ 5 \leq n \leq N. \quad (P.2)
\]

Since the inequality constrain conditions in (P.1) are tighter than those in (P.2), the upper bound of the magnitude of spatial-difference-based statistic calculated in (P.2) is larger than that in (P.1). Hence, the upper bound of the magnitude of spatial-difference-based statistic calculated in (P.2) is feasible for the real-time FDIA detection.

The optimization problem formulated in (P.2) can be regarded as a geometric programming problem, because the values of \( \{s_{ij}^k(t - n)\}_{n=1}^{N-1} \) are known to node \( i \). Since all the variables in problem (P.2) are mutually independent, the maximization can be achieved at the extreme point of each variable. That is, the inequality constraints in problem (P.2), \( |\lambda'_{n}| \leq |\hat{\lambda}_4|, \ 5 \leq n \leq N \), can be transformed into the constraints, \( \lambda'_{n} \in [\hat{\lambda}_4, -\hat{\lambda}_4], \ 5 \leq n \leq N \). Hence, the problem formulated in (P.2) can be transformed into

\[
\Lambda^k_{ij}(t) = \max_{\lambda'_{n}(2 \leq n \leq N)} \left| \sum_{n=1}^{N-1} d_n s_{ij}^k(t - n) \right|, \ t \geq N - 1, \ \\
\text{s. t. } \lambda'_2 = \hat{\lambda}_2, \lambda'_3 = \hat{\lambda}_3, \lambda'_4 = \hat{\lambda}_4, \ \\
\lambda'_{n} \in [\hat{\lambda}_4, -\hat{\lambda}_4], \ 5 \leq n \leq N. \quad (P.3)
\]

In problem (P.3), since each variable in \( \{\lambda'_{n}\}_{n=1}^{N} \) equals to either \( \hat{\lambda}_4 \) or \(-\hat{\lambda}_4\), there are \( 2^{N-4} \) possible choices for the values of \( \{\lambda'_{n}\}_{n=5}^{N} \). Therefore, the problem formulated in (P.3) can be solved with the enumeration method. That is, \( 2^{N-4} \) possible choices for the values of \( \{\lambda'_{n}\}_{n=5}^{N} \) can be considered simultaneously to find the optimal combination choice that results in a maximum \( \Lambda^k_{ij}(t) \).

If the injector \( d_{A} \) launches the FDIA at iteration \( t_{A} - 1 \), a false data \( u'_{ij}(t_{A} - 1) \) is injected to its state \( x'_{A} \) \( (t_{A} - 1) \). Without loss of generality, it is assumed that the magnitude of injected false data is much larger than the magnitude of spatial-difference-based statistic before the FDIA is launched. Under this assumption, the upper bound of the magnitude of spatial-difference-based statistic calculated in problem (P.3) can be asymptotically approximated as \( \Lambda^k_{ij}(t) \approx \sum_{n=1}^{N-1} d_n^* u'_{ij}(t - n), \ t \geq t_{A}, \) where \( \{d_n^*\}_{n=1}^{N-1} \) is the variables obtained by solving problem (P.3). Since the magnitude of \( u'_{ij}(t_{A} - 1) \) is much larger than the magnitude of spatial-difference-based statistic before the FDIA is launched, we have \( \sum_{n=1}^{N-1} d_n^* u'_{ij}(t - n) >> \sum_{n=1}^{N-1} d_n^* s^k_{ij}(t - n), \ t \geq t_{A} \). Therefore, the upper bound of the magnitude of spatial-difference-based statistic will experience a sharp increase after iteration \( t_{A} - 1 \).

In order to make the upper bound of the magnitude of spatial-difference-based statistic change smoothly and the condition \( E[s_{ij}^k(t) | H_{1}^t] > 0 \) be satisfied, we modify the problem formulated in (P.3) by replacing the statistic \( s_{ij}^k(t) \) with \( \text{sgn}(s_{ij}^k(t - 1)) \Lambda^k_{ij}(t) \) if \( |\hat{s}^k_{ij}(t)| > \Lambda^k_{ij}(t) \); otherwise, \( s_{ij}^k(t) = \hat{s}^k_{ij}(t) \). Similar to problem (P.3), the problem formulated in (P.4) can be solved with the enumeration method. The real-time FDIA detection method can be summarized as Algorithm 1. With the real-time FDIA detection method, each normal node \( i \) can obtain the set of malicious nodes in the neighborhood as \( A_i \).

Remark 2: Due to the limited memory and computational capability of nodes, the computational complexity of the solving process of problem (P.4) may be too high. In order to reduce the computational burden for each node, the solving process of problem (P.4) can be simplified as \( \Lambda^k_{ij}(t) = \max_{\lambda'_{n}(2 \leq n \leq N)} \left| \sum_{n=1}^{N-1} d_n s_{ij}^k(t - n) \right|_{\lambda'_n = \pm \hat{\lambda}_4, \ 5 \leq n \leq N}, \ \\
\left| \sum_{n=1}^{N-1} d_n s'_{ij}^k(t - n) \right|_{\lambda'_n = -\hat{\lambda}_4, \ 5 \leq n \leq N}. \) However, the obtained \( \Lambda^k_{ij}(t) \) cannot be guaranteed to be an upper bound of the magnitude of spatial-difference-based statistic, and the false-alarm probability of the real-time FDIA detection method is likely to increase.

Remark 3: For a real-time FDIA detection mechanism, the most important performance indicator is the detection delay after the FDIA occurs. When encountering the sudden FDIA or the dynamic FDIA presented in Section III, the proposed real-time FDIA detection method may show different detection performance. If the sudden FDIA is launched, the magnitude of spatial-difference-based statistic of the injector experiences a steep increase, and the real-time statistic of the injector also rises quickly. In this circumstance, the injector can be identified by normal nodes in a short time. On the other hand, if the injector launches the dynamic FDIA and sets the attack parameter \( \mu_A \) to be a relatively small value, the magnitude of spatial-difference-based statistic of the injector is likely to be lower than the upper bound obtained in problem (P.4), which results in a low detection probability and a long detection delay. Therefore, it is more difficult to resist the dynamic FDIA than the sudden FDIA.
Algorithm 1 Real-Time FDIA Detection Method (implemented by normal node $i$, $i \in \mathcal{V} \setminus \mathcal{A}$)

**Input and Parameter Initialization**

1. Input $h$, \( \{s^k_{ij}(t)\}_{1 \leq k \leq K, 1 \leq t \leq T, j \in \mathcal{N}_i} \)
2. Calculate the upper bound of the magnitude of spatial-difference-based statistic for each neighbor node $j$ by solving the optimization problem in (P.4).

**Real-time FDIA Detection Procedure**

(2). for $k = 2$ to $K$, $k++$

(3). Estimate eigenvalues $\lambda_2$, $\lambda_3$, and $\lambda_4$ with (27) and the statistics in the former instance, $s^{k-1}_{ij}(t)$, $\dot{s}^{k-1}_{ij}(t)$ and $\ddot{s}^{k-1}_{ij}(t)$.

(4). for $t = N - 1$ to $T$, $t++$

(5). Calculate the one-shot statistic $\Psi_{ij}^k(t)$ for each neighbor node $j$ with (18).

(6). Calculate the real-time statistic $S_{ij}^k(t)$ for each neighbor node $j$ with (16).

(7). Determine whether the FDIA is launched by neighbor node $j$ or not at iteration $t$ with (17).

(8). Calculate statistics $\dot{s}_{ij}^k(t)$ and $\ddot{s}_{ij}^k(t)$ for each neighbor node $j$ with the spatial-difference-based statistics $s_{ij}^k(t)$.

(9). end for

(10). end for

**Output**

(12). Output the set of malicious nodes in the neighborhood: $\mathcal{A}_i = \{j | T_{ij}(h) \leq T\}$.

B. Resilient Weight Vector Adjustment

Once node $j$ in set $\mathcal{N}_i$ is identified as a malicious node, normal node $i$ sets the weight $w_{ij}$ to be 0 and stops receiving any information from node $j$. In order to guarantee the consensus among the normal nodes in the network, the weight vector of each node should be adjusted appropriately after the malicious nodes being isolated.

According to [3], as long as the weight matrix $W$ is doubly-stochastic, and the spectral radius of matrix $(W - 11^T/N)$ is strictly less than 1, the average consensus of the network can be achieved. In order to satisfy the above two conditions, the weight vectors of normal nodes can be adjusted based on the maximum-degree protocol or the Metropolis protocol [29].

In the maximum-degree protocol, the weights for each normal node $i$ are set to be

\[
w_{ij} = \begin{cases} 
\frac{1}{N} \left( \frac{1}{N} \right), & \text{if } j \in \mathcal{N}_i \setminus \mathcal{A}_i, \{0.5pc\} \\
1 - \sum_{k \in \mathcal{N}_i \setminus \mathcal{A}_i} w_{ik}, & \text{if } j = i, \\
0, & \text{otherwise.}
\end{cases}
\]

(28)

In the Metropolis protocol, the weights for each normal node $i$ are set to be

\[
w_{ij} = \begin{cases} 
1, & \text{if } j \in \mathcal{N}_i \setminus \mathcal{A}_i, \\
1 + \max \{D_i, D_j\}, & \text{if } j = i, \\
1 - \sum_{k \in \mathcal{N}_i \setminus \mathcal{A}_i} w_{ik}, & \text{otherwise.}
\end{cases}
\]

(29)

where $D_i$ denotes the degree of node $i$ after the malicious nodes are removed from $\mathcal{N}_i$, and $D_j = |\mathcal{N}_j| - |\mathcal{A}_j|$. In the Metropolis protocol, each node $i$ shares the real-time $D_i$ with its neighbor nodes.

After malicious nodes are isolated from normal nodes, the obtained subgraph is denoted as $G_N$, where $G_N = (\mathcal{V} \setminus \mathcal{A}, \mathcal{E}_N)$, and $\mathcal{E}_N = \{(i,j) \mid i \in \mathcal{V} \setminus \mathcal{A}, j \in \mathcal{N}_i \setminus \mathcal{A}_i\}$. It is proven in [29] that as long as the set of malicious nodes in the neighborhood is correctly identified by each normal node, and subgraph $G_N$ is connected, the average consensus can be achieved with either the maximum-degree protocol or the Metropolis protocol.

C. REASSESSMENT OF ATTACKERS WITH PUNISHMENT SCHEME

As we know, the false-alarm is inevitable for an attack detection method. Without the reassessment mechanism, normal nodes implementing the proposed real-time FDIA detection method will eventually regard all of their neighbors as attackers as long as the number of iterations tends to infinity. To mitigate the effect of false-alarm of the proposed real-time FDIA detection method on the estimation performance, we propose a reassessment mechanism to reassess malicious nodes.

If normal node $i$ identifies its neighbor node $j$ as an attacker, the link between node $i$ and node $j$ will be cut off, and $w_{ij} = 0$. After that, node $i$ keeps assessing the behavior of node $j$. Once node $i$ observes that the behavior of node $j$ returns to normal, the link between node $i$ and node $j$ will be rebuilt, and $w_{ij} > 0$.

As shown in Subsection III-C, an intelligent attacker can surveil the behavior of normal nodes and dynamically switch between launching attack and performing normal consensus update. In order to resist the surveillance and response mechanism, malicious nodes should be inspected continuously. Utilizing the spatial-difference-based statistic, the reassessment mechanism can be expressed as

\[
K_R \prod_{k=1}^{T} \prod_{t=1}^{T} g \left( \eta_R - |s_{ij}^k(t)| \right) = \begin{cases} 
1, & \text{if } \mathcal{R}_{ij}^k \leq \mathcal{R}_{ij}^0, \\
0, & \text{otherwise.}
\end{cases}
\]

(30)

where $g(\cdot)$ denotes the step function, and $g(x) = [\text{sgn}(x) + 1]/2$; $\mathcal{R}_{ij}^k$ represents the hypothesis that node $j$ returns to normal; $\mathcal{R}_{ij}^0$ represents the hypothesis that node $j$ is still malicious; $K_R$ denotes the number of instances used for the reassessment; and $\eta_R$ is the decision threshold for the inspection at each iteration. The reassessment mechanism in (30) indicates that a malicious node can be declared to return to normal only if the magnitude of spatial-difference-based
statistic is lower than the decision threshold at each iteration during the reassessing period. In practical application, the decision threshold should be chosen to satisfy the upper bound of the false-alarm probability required by the system. Therefore, the relationship between the decision threshold and the false-alarm probability should be derived.

Similar to [14], we assume that the initial estimation of each node is independent and identically distributed, which follows the normal distribution with mean \( \mu_x \) and variance \( \sigma_x^2 \). Thus, the spatial-difference-based statistic \( x_{ij}^k (t) \) follows the normal distribution with zero-mean and variance \( \sigma_{ij}^2 (t) = e_i^T W_t [W_t]^T e_j \sigma_x^2 \). The false-alarm probability of the inspection at each iteration can be expressed as

\[
P_{FA} = 1 - \prod_{t=1}^{T} \left[ 1 - P_{FA, INS} (t) \right]^{K_R} = 1 - \prod_{t=1}^{T} \left( 2 \Phi \left( \eta_R / \sigma_S (t) \right) - 1 \right)^{K_R}. \tag{31}
\]

Furthermore, in order to adaptively adjust the judging criteria of the reassessment mechanism to resist an intelligent attacker, we propose a punishment scheme for the reassessment mechanism.

If node \( j \) is identified as an attacker, the total number of instances used for reassessing node \( j \) is updated as

\[
K_R = K_R + \Delta,
\]

where \( \Delta \) is a punishment factor, and \( \Delta \in \mathbb{Z}^+ \). With this punishment scheme, the more times a node being identified as an attacker, the longer the reassessment will last for.

V. NUMERICAL RESULTS AND DISCUSSIONS

A network with 12 nodes is considered, as shown in Fig. 1. For each node \( i \) and its neighbor node \( j \), the value of \( w_{ij} \) follows the normal distribution with mean \( \mu_w \) and variance \( \sigma_w^2 \), and \( w_{ii} = 1 - \sum_{j \in \mathcal{N}_i} w_{ij} \). The true value of unknown scalar parameter to be estimated is \( x_0 \). The initial estimation of each node is assumed to be independent and identically distributed, and the initial state vector follows the multivariate normal distribution as \( \mathbf{x}^k (0) \sim \mathcal{N} (x_0 \mathbf{1}, \sigma_x^2 \mathbf{1}) \), where \( x_0 = 10 \) and \( \sigma_x^2 = 1 \).

The performance of learning-based cooperative FDIA strategy and the corresponding FDIA detection method is evaluated in scenario 1 and scenario 2, respectively. As shown in Fig. 1a, in scenario 1, node 11 is set to be the injector, and node 1 and node 6 are set to be the listeners. As shown in Fig. 1b, the injector is node 11 and the listener is node 1 in scenario 2. The observable node set is \( \mathcal{V}_A = \{ 1, 2, 3, 4, 7, 8, 9, 11, 12 \} \), and the vulnerable node set is \( \mathcal{V}_V = \{ 3, 7, 8 \} \). Node 7 is assumed to deploy the proposed FDIA mitigation techniques.

A. PERFORMANCE OF THE LISTENING AND LEARNING PROCESS IN STAGE ONE OF THE ATTACK

In this subsection, we evaluate the performance of the listening and learning process in scenario 1 and scenario 2, respectively.

1) SCENARIO 1

Fig. 2 shows the estimation error of weight matrix, where the estimation error is defined as \( \text{Err}_{\text{scenario 1}} = \frac{\Delta}{\sum_{i=1}^{N} \sum_{j=1}^{N} (w_{ij} - \hat{w}_{ij})^2} \), and \( \sigma_w = 0.02 \). From Fig. 2, we observe that the estimation error of weight matrix decreases as the redundancy of observations, \( \Delta N \), increases. This is because redundant observations can bring extra...
FIGURE 3. The performance of the process of searching for vulnerable nodes.

information for the estimation. Moreover, we observe that the estimation error decreases as the value of $\mu_w$ increases. The reason for this phenomenon is that each diagonal element of weight matrix, $w_{ii}$, $1 \leq i \leq N$, becomes smaller as $\mu_w$ increases, which makes the columns of matrix $X_k (0 : N - 1)$ be less correlated with each other, and the condition number of $X_k (0 : N - 1)$ becomes smaller [27].

2) SCENARIO 2

Fig. 3 shows the performance of the process of identifying vulnerable nodes, where $\rho = 1.1$, $\mu_w = 0.1$, and $\sigma_w = 0.02$. The false-alarm probability for identifying vulnerable nodes is defined as

$$P_{FA, Vul}^i \triangleq \Pr \left( \right. \bigg| \begin{array}{l} X_{kV} (0 : N - 1) \bigg|_1 < \eta_{V} i \notin V_{v} \bigg) \right.$$

and the detection probability for identifying vulnerable nodes is defined as

$$P_{D, Vul}^i \triangleq \Pr \left( \right. \bigg| \begin{array}{l} X_{kV} (0 : N - 1) \bigg|_1 < \eta_{V} i \in V_{v} \bigg) \right.$$

From Fig. 3a, we observe that the false-alarm probability for identifying vulnerable nodes decreases as the redundancy of observations increases. The reason is that for an invulnerable node $i$, $i \notin V_{v}$, if the redundancy of observations increases, the 1-norm of the component of state vector $\left[ x_k^i (1 : N') \right]_1$ in $\mathcal{R} \left( X_{kV}^i (0 : N - 1) \bigg|_1 \right)$ becomes larger, which makes node $i$ is easier to be identified as an invulnerable node. Moreover, the false-alarm probability for identifying nodes 2 and 9 is smaller than that for identifying nodes 4 and 12. The reason for this phenomenon is that nodes 2 and 9 have two neighbor nodes outside set $V_{A}$, whereas nodes 4 and 12 only have one neighbor node outside set $V_{A}$. Therefore, the component of state vectors of nodes 2 and 9 in $\mathcal{R} \left( X_{kV}^i (0 : N - 1) \bigg|_1 \right)$ is higher than that of nodes 4 and 12.

From Fig. 3b, we observe that the redundancy of observations has little effect on the detection probability for identifying vulnerable nodes. The reason is that the injector can succeed in identifying vulnerable nodes with 100% accuracy in theory. However, due to the limited calculation precision, the projection of state vector of a vulnerable node $v$ on $\mathcal{R} \left( X_{kV}^v (0 : N - 1) \bigg|_1 \right)$ may not exactly equal to 0. Hence, some missed detections may happen occasionally. Moreover, the detection probability for identifying nodes 3 and 8 is higher than that for identifying node 7. The reason for this phenomenon is that nodes 3 and 8 have respectively four and five neighbors, while node 7 only has three neighbors. Hence, the state vectors of nodes 3 and 8 are more evenly distributed in $\mathcal{R} \left( X_{kV}^v (0 : N - 1) \bigg|_1 \right)$ than node 7. Under the same calculation precision, the components of state vectors of nodes 3 and 8 in $\mathcal{R} \left( X_{kV}^v (0 : N - 1) \bigg|_1 \right)$ are smaller than that of node 7, which means nodes 3 and 8.
can be correctly identified as vulnerable nodes with a higher probability.

Fig. 4 shows the estimation error of weight vectors for vulnerable nodes, where the estimation error is defined as

$$\text{Err}_{\text{scenario 2}} = \sum_{i \in V} \sum_{j \in V_{/i}} (w_{ij} - \hat{w}_{ij})^2, \quad \text{and} \quad \sigma_w = 0.02.$$  

From Fig. 4, we observe that the estimation error decreases as the redundancy of observations or parameter $\mu_w$ increases. Compared with the result in Fig. 2, the estimation error in scenario 2 is much smaller than that in scenario 1. The reason for this phenomenon is two-fold. First, the number of elements to be estimated in scenario 2 is less than that in scenario 1, which makes the cumulative sum be smaller in scenario 2. Second, for simplicity, we assume that the identifying of vulnerable nodes has been completed without error when we evaluate the estimation error in scenario 2.

**B. PERFORMANCE OF THE FALSE DATA INJECTION IN STAGE TWO OF THE ATTACK**

In this subsection, we evaluate the effectiveness of the FDIA strategy, including the sudden FDIA and the dynamic FDIA.
we observe that a larger µ (injector) and node 2 under dynamic FDIA. From Fig. 7, for scenario 1 and scenario 2.

For simplicity, we assume that normal nodes do not adopt any defense strategy to resist the attack. In the simulations, we set µw = 0.1, σw = 0.02, and ΔN = 8. The attacker launches the FDIA after iteration N + ΔN = 20. The divergence between the goal of the injector and the true value of unknown scalar parameter is defined as ΔG = Gk − x0.

1) SUDDEN FDIA STRATEGY
Fig. 5 shows the consensus update process of node 11 (injector) and node 2 under sudden FDIA. From Fig. 5, we observe that the states of node 11 and node 2 gradually converge to the true value of unknown parameter if no attack is launched. However, if the sudden FDIA is launched by node 11, the state of node 11 experiences a steep rise at iteration 21. After iteration 21, the states of node 11 and node 2 gradually converge to Gk. Moreover, a larger value of ΔG leads to a steeper rise of the injector’s state at iteration 21 and a longer convergence time.

2) DYNAMIC FDIA STRATEGY
Fig. 6 shows the injected data under dynamic FDIA at each iteration, where ΔG = 5. From Fig. 6, we observe that a larger µG results in a larger injected data at the beginning of the attack and a shorter convergence time. Compared between the results in Fig. 6a and Fig. 6b, we observe that the injected data at the beginning of FDIA in scenario 1 is slightly different from that in scenario 2. The reason is that the attack parameter wA,0,k is calculated in different ways for scenario 1 and scenario 2.

Fig. 7 shows the consensus update process of node 11 (injector) and node 2 under dynamic FDIA. From Fig. 7, we observe that a larger µA corresponds to a shorter convergence time and a steeper rise of the injector’s state, xA,k (t), at the beginning of attack.

Fig. 8 shows the cumulative sum of injected data, ∑t uA,k (t). From Fig. 8, we observe that the cumulative sum of injected data converges to a constant value for different values of µA. The reason for this phenomenon is that no matter what value of µA is, the expectation of limit value of ∑t uA,k (t), E[∑t→∞ uA,k (t)], equals to ΔG/σA. Therefore, the cumulative sum of injected data only depends on the weight matrix. Furthermore, the injected data in the sudden FDIA is also ΔG/σA. Therefore, if the goal of the injector is determined, the cumulative sum of injected data is a constant value regardless the type of FDIA or the value of µA.

C. PERFORMANCE OF THE REAL-TIME FDIA DETECTION METHOD
In this subsection, the performance of real-time FDIA detection method against the learning-based cooperative FDIA is evaluated, where µw = 0.1, σw = 0.02, and ΔN = 8.

In order to implement the real-time FDIA detection method, eigenvalues λ2, λ3, and λ4 are first estimated with the algorithm in Theorem 1. Fig. 9 shows the estimation error of the eigenvalues, where the estimation error is defined as Errλi = λi − ̂λi, where ̂λi is the estimation of the eigenvalues. From Fig. 9, we observe that the estimation error decreases as the total number of iterations increases. The conclusion is consistent with the analysis in Appendix. Moreover, the estimation errors satisfy Errλ1 > Errλ2 > Errλ3, regardless the value of T. The reason is that eigenvalues λ2, λ3, and λ4 are estimated sequentially.

1) REAL-TIME ATTACK DETECTION METHOD AGAINST THE SUDDEN FDIA
Fig. 10 shows the performance of the proposed real-time attack detection method against the sudden FDIA, where the spatial-difference-based statistic, the one-shot statistic, and

![FIGURE 10. The performance of real-time attack detection against the sudden FDIA.](image)

![FIGURE 11. The ROC curve of attack detection against the sudden FDIA at iteration 21.](image)
the real-time statistic obtained by node 7 are shown in 10a, 10b and 10c, respectively.

As shown in Fig. 10a, at iteration 21, the spatial-difference-based statistic of the injector, \( s^{k}_{7,11} (t) \), experiences a steep increase, whereas the spatial-difference-based statistics of the normal nodes, \( s^{k}_{7,3} (t) \) and \( s^{k}_{7,8} (t) \), almost remain unchanged. After iteration 21, \( s^{k}_{7,11} (t) \) gradually declines to 0; meanwhile, the effect of injected data propagate to normal nodes 3 and 8, and makes \( s^{k}_{7,3} (t) \) and \( s^{k}_{7,8} (t) \) change in the opposite direction. Moreover, a larger value of \( \Delta G \) leads to a steeper increase of \( s^{k}_{7,11} (t) \) at iteration 21 and a longer convergence time. As shown in Fig. 10b and Fig. 10c, at iteration 21, the one-shot statistic and the real-time statistic of node 11 experience a steep increase, whereas these two statistics of nodes 3 and 8 almost remain unchanged. Therefore, as long as the decision threshold \( h \) in (17) is chosen appropriately, the injector can be identified with the real-time FDIA detection method. By cutting off the link connecting with the injector (i.e., setting \( w^{7,11} \) to be 0), node 7 can resist the sudden FDIA effectivelly.

To verify the timeliness of the real-time FDIA detection method, we evaluate the performance of attack detection at iteration 21. Fig. 11 shows the receiver operation characteristic (ROC) curve of attack detection against the sudden FDIA at iteration 21, where \( \mu_w = 0.02, \sigma_w = 0.005 \), the false-alarm probability for attack detection is defined as 
\[
P_{FA, Att} = \frac{1}{2} Pr \left[ S^{k}_{7,3} (21) > h \right] + \frac{1}{2} Pr \left[ S^{k}_{7,8} (21) > h \right],
\]
and the detection probability for attack detection is defined as 
\[
P_{D, Att} = \frac{1}{2} Pr \left[ S^{k}_{7,11} (21) > h \right].
\]

From Fig. 11, we observe that a larger magnitude of \( \Delta G \) results in a better detection performance. The reason is that the data injected by the injector at iteration 21 increases as the magnitude of \( \Delta G \) increases, which make the injector be identified more easily. Hence, the real-time FDIA detection method can achieve a high detection probability, as well as maintaining a low false-alarm probability at iteration 21.

2) REAL-TIME ATTACK DETECTION METHOD AGAINST THE DYNAMIC FDIA

Figs. 12 and 13 show the performance of real-time attack detection method against the dynamic FDIA in scenario 1 and scenario 2, respectively, where \( \Delta G = 5 \).

As shown in Figs. 12a and 13a, at iteration 21, the spatial-difference-based statistic of the injector, \( s^{k}_{7,11} (t) \), experiences a steep increase, whereas the spatial-difference-based statistics of the normal nodes, \( s^{k}_{7,3} (t) \) and \( s^{k}_{7,8} (t) \), almost remain unchanged. After iteration 21, the spatial-difference-based statistics of the normal nodes, \( s^{k}_{7,3} (t) \) and \( s^{k}_{7,8} (t) \), increase first and then decrease. Moreover, we also observe that a larger value of \( \mu_A \) results in a steeper change of
spatial-difference-based statistics at the beginning of attack and a shorter convergence time.

As shown in Figs. 12b, 12c, 13b and 13c, at iteration 21, the one-shot statistics and the real-time statistic of the injector, $\Psi_{7,11}^i(t)$ and $S_{7,11}^i(t)$, experience a steep increase, whereas the one-shot statistics and the real-time statistics of normal nodes, $\Psi_{7,3}^k(t)$, $S_{7,3}^k(t)$ and $S_{7,8}^k(t)$, almost remain unchanged. Hence, the dynamic FDIA can be detected by the real-time attack detection method with an appropriate decision threshold. From Figs. 12c and 13c, we also observe that a larger value of $\mu_A$ results in a steeper increase of $S_{7,11}^i(t)$ at iteration 21, which makes the injector be identified more easily. Moreover, from Figs. 12c and 13c, we also observe that the real-time statistic of each node will converge to a constant value for different values of $\mu_A$. The reason for this phenomenon is that if the dynamic FDIA is launched, the limit value of the real-time statistic, $\lim_{t \to \infty} S_{ij}^k (t)$, approximately equals to $e_j^T \left( \sum_{t=1}^{\infty} W^t \right) e_{A} E \left[ \sum_{t=1}^{\infty} u_{ij}^k (t) \right] = e_j^T \left( \sum_{t=1}^{\infty} W^t \right) e_{A} \Delta G / \nu_{A}$, which only depends on the goal of the attacker and the weight matrix.

Fig. 14 shows the ROC curve of the real-time attack detection method against the dynamic FDIA at iteration 21, where $\mu_w = 0.1$, and $\sigma_w = 0.02$. From Fig. 14, we observe that the detection performance improves as the value of $\Delta G$ or $\mu_A$ increases. The reason for this phenomenon is that the data injected by the injector at iteration 21 increases as the value of $\Delta G$ or $\mu_A$ increases, which make the attacker be identified more easily. Hence, the real-time FDIA detection method can achieve a high detection probability, as well as ensuring a low false-alarm probability at iteration 21.

VI. CONCLUSION

In this paper, we investigated the FDIA problem in the consensus-based distributed estimation. Since the weight matrix is important for both attackers and defenders, the effect of the information of weight matrix on the FDIA and the FDIA mitigation techniques was studied. First, a learning-based cooperative FDIA strategy was designed, where the weight matrix was estimated and utilized to launch the sudden FDIA and the dynamic FDIA. With the obtained information of weight matrix, the attacker can tamper the consensus result of the network to a pre-designed false value. Furthermore, to enhance the covertness of FDIA, a real-time surveillance and response mechanism was constructed for malicious nodes to surveil the consensus update process of normal nodes. By switching between launching the FDIA and performing normal consensus update dynamically, the attacker can bypass existing FDIA detection methods. In order to ensure the security of consensus-based distributed estimation, a real-time attack detection method was proposed to defend against the learning-based cooperative FDIA. Considering that the false-alarm of attack detection system may degrade the performance of consensus-based distributed estimation, we further presented a reassessment mechanism to reassess malicious nodes. With the embedded punishment scheme, the reassessment mechanism can resist the surveillance and response mechanism in the learning-based cooperative FDIA effectively. Comprehensive simulation results show that the attacker can acquire the information of weight matrix and tamper the consensus result to a false value through launching the learning-based cooperative FDIA. Moreover, the proposed real-time FDIA detection method can identify the attacker promptly.

APPENDIX

PROOF OF THEOREM 1

The proofs of 27a, 27b, and 27c are sequentially provided as follows.

According to (21), the spatial-difference-based statistic is expressed as $s_{ij}^k (t) = \sum_{n=2}^{N} \lambda_n c_n$. Since all the eigenvalues in $\lambda'$ are non-zero and different from each other, we have $\lim_{t \to \infty} (\lambda'_i / \lambda'_2) = 0$, $i \geq 3$. Hence, (21) can be rewritten as

$$s_{ij}^k (t) = c_2 \lambda'_2 + o (\lambda'_2), \quad (A.1)$$

where $o (\lambda'_2)$ is an infinitesimal of higher order than $\lambda'_2$. According to (A.1), we have

$$\lim_{t \to \infty} \frac{s_{ij}^k (t)}{s_{ij}^k (t - 1)} = \lambda'_2. \quad (A.2)$$

This completes the proof of (27a).
Similar to (A.1), the spatial-difference-based statistic $s_{ij}^k(t)$ can be rewritten as

$$s_{ij}^k(t) = c_2 \lambda_2^2 + c_3 \lambda_3 + o(\lambda_3^3). \quad (A.3)$$

Hence, we have

$$s_{ij}^k(t) = s_{ij}^k(t) s_{ij}^k(t-2) - \left[ s_{ij}^k(t-1) \right]^2 = (c_2 \lambda_2^2 + c_3 \lambda_3^2) (c_2 \lambda_2^2 + c_3 \lambda_3^2) - (c_2 \lambda_2^2 + c_3 \lambda_3^2)^2 + o(\lambda_3^3), \quad (A.4)$$

and

$$\lim_{t \to \infty} s_{ij}^k(t) = \lambda_2^2 \lambda_3. \quad (A.5)$$

This completes the proof of (27a).

Similar to (A.4), statistic $s_{ij}^k(t)$ can be rewritten as

$$s_{ij}^k(t) = c_2 \lambda_2^2 \lambda_3^2 + c_2 c_4 \lambda_2^2 \lambda_3^2 (\lambda_2 - \lambda_3)^2 + c_3 c_4 \lambda_2^2 \lambda_3^2 (\lambda_3 - \lambda_4)^2 + o(\lambda_3^3), \quad (A.6)$$

Hence, we have

$$s_{ij}^k(t) = s_{ij}^k(t) s_{ij}^k(t-2) - \left[ s_{ij}^k(t-1) \right]^2 = (\lambda_2 - \lambda_3)^2 (\lambda_2 - \lambda_3)^2 (\lambda_3 - \lambda_4)^2 c_2 c_4 \lambda_2^2 \lambda_3^2 \lambda_4^2 + o(\lambda_3^3), \quad (A.7)$$

and

$$\lim_{t \to \infty} s_{ij}^k(t) = \lambda_2^2 \lambda_3 \lambda_4. \quad (A.8)$$

This completes the proof of (27c).

Similar to (A.7), statistic $s_{ij}^k(t)$ can be further rewritten as

$$s_{ij}^k(t) = (\lambda_2 - \lambda_3)^2 (\lambda_2 - \lambda_4)^2 (\lambda_3 - \lambda_4)^2 + c_2 c_3 c_4 \lambda_2^2 \lambda_3^2 \lambda_4^2 + o(\lambda_3^3), \quad (A.9)$$

In the second and third items of the right side of (A.9), since the relationship between the magnitudes of $\left[ 2 \lambda_2^2 \lambda_3^2 \lambda_4^2 \right]$ and $\left[ \lambda_2^2 \lambda_3^2 \lambda_4^2 \right]$ is unknown, eigenvalues $\lambda_i$, $i \geq 5$ cannot be uniquely determined with Algorithm 1.

REFERENCES

[1] F. Garin and L. Schenato, “A survey on distributed estimation and control applications using linear consensus algorithms,” in *Networked Control Systems* (Springer Lecture Notes in Control and Information Sciences), vol. 406, New York, NY, USA: Springer, 2011, pp. 75–107.

[2] F. Garin and S. Zampieri, “Mean square performance of consensus-based distributed estimation over regular geometric graphs,” *SIAM J. Control Optim.*, vol. 50, no. 1, pp. 306–333, Jan. 2012.

[3] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, Sep. 2004.

[4] B. Kailkhura, S. Brahma, and P. K. Varshney, “Data falsification attacks on consensus-based detection systems,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 1, pp. 145–158, Mar. 2017.

[5] P. Pasqualetti, A. Bicchi, and F. Bullo, “Distributed intrusion detection for secure consensus computations,” in *Proc. 46th IEEE Conf. Decis. Control*, New Orleans, LA, USA, Dec. 2007, pp. 5594–5599.

[6] C. Zhao, J. He, and J. Chen, “Resilient consensus with mobile detectors against malicious attacks,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 1, pp. 60–69, Mar. 2018.

[7] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, “Resilient asymptotic consensus in robust networks,” *IEEE J. Sel. Areas Commun.*, vol. 31, no. 4, pp. 766–781, Apr. 2013.

[8] H. Zhang and S. Sundaram, “Robustness of information diffusion algorithms to locally bounded adversaries,” in *Proc. Amer. Control Conf. (ACC)*, Montreal, QC, Canada, Jun. 2012, pp. 5855–5861.

[9] D. Saldana, A. Prorok, S. Sundaram, M. F. M. Campos, and V. Kumar, “Resilient consensus for time-varying networks of dynamic agents,” in *Proc. Amer. Control Conf. (ACC)*, Seattle, WA, USA, May 2017, pp. 252–258.

[10] Y. Chen, S. Kar, and J. M. F. Moura, “Resilient distributed estimation through adversary detection,” *IEEE Trans. Signal Process.*, vol. 66, no. 9, pp. 2455–2469, May 2018.

[11] Y. Chen, S. Kar, and J. M. F. Moura, “The Internet of Things: Secure distributed inference,” *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 64–75, Sep. 2018.

[12] X. Ge, Q. Han, M. Zhong, and X. Zhang, “Distributed Krein space-based attack detection over sensor networks under deception attacks,” *Automatica*, vol. 109, no. 108557, pp. 1–10, Nov. 2019.

[13] R. Gentz, H.-T. Wai, A. Scaglione, and A. Leshem, “Detection of data injection attacks in decentralized learning,” in *Proc. 49th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2015, pp. 350–354.

[14] R. Gentz, S. Xiaoxiao Wu, H.-T. Wai, A. Scaglione, and A. Leshem, “Data injection attacks in randomized gossiping,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 4, pp. 523–538, Dec. 2016.

[15] S. Sundaram and C. N. Hadjicostis, “Distributed function calculation via linear iterative strategies in the presence of malicious agents,” *IEEE Trans. Automat. Control*, vol. 56, no. 7, pp. 1495–1508, Jul. 2011.

[16] F. Pasqualetti, A. Bicchi, and F. Bullo, “Consensus computation in unreliable networks: A system theoretic approach,” *IEEE Trans. Automat. Control*, vol. 57, no. 1, pp. 90–104, Jan. 2012.

[17] F. Pasqualetti, F. Dorrler, and F. Bullo, “Attack detection and identification in cyber-physical systems,” *IEEE Trans. Automat. Control*, vol. 58, no. 11, pp. 2715–2729, Nov. 2013.

[18] S. Sundaram and C. N. Hadjicostis, “Finite-time distributed consensus in graphs with time-invariant topologies,” in *Proc. Amer. Control Conf.*, New York, NY, USA, Jul. 2007, pp. 711–716.

[19] E. Montijano, J. I. Montijano, and C. Sagues, “Chebyshev polynomials in distributed consensus applications,” in *IEEE Trans. Signal Process.*, vol. 61, no. 3, pp. 693–706, Feb. 2013.

[20] J. M. Dion, C. Commault, and J. van der Woude, “Generic properties and control of linear structured systems: A survey,” *Automatica*, vol. 39, no. 7, pp. 1125–1144, Jul. 2003.

[21] K. J. Reinschke, *Multivariable Control: A Graph-Theoretic Approach*. New York, NY, USA: Springer-Verlag, 1988.

[22] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA, USA: SIAM, 2000.

[23] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, Formulas*. Philadelphia, PA, USA: Princeton Univ. Press, 2009.

[24] E. S. Page, “Continuous inspection schemes,” *Biometrika*, vol. 41, nos. 1–2, pp. 100–115, Jun. 1954.

[25] Y. Huang, J. Tang, Y. Cheng, H. Li, K. A. Campbell, and Z. Han, “Real-time detection of false data injection in smart grid networks: An adaptive CUSUM method and analysis,” *IEEE Syst. J.*, vol. 10, no. 2, pp. 532–543, Jun. 2016.
[26] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 1996.

[27] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[28] S.-H. Hou, “Classroom note: A simple proof of the Leverrier–Faddeev characteristic polynomial algorithm,” *SIAM Rev.*, vol. 40, no. 3, pp. 706–709, Sep. 1998.

[29] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in *Proc. IEEE/ACM Int. Symp. Inf. Process. Sensor Netw.*, Los Angeles, CA, USA, Apr. 2005, pp. 63–70.

**QIAOMU JIANG** received the B.S. degree in communication engineering from Jinan University, Guangzhou, China, in 2015. He is currently pursuing the Ph.D. degree with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. His current research interests include cyber security and distributed signal processing.

**HUIFANG CHEN** (Member, IEEE) received the B.S. degree in electronic engineering, the M.S. degree in communications and electronic systems, and the Ph.D. degree in communications and information systems from Zhejiang University, Hangzhou, China, in 1994, 1997, and 2000, respectively. Since 2000, she has been with Zhejiang University, where she is currently a Professor with the College of Information Science and Electronic Engineering. She is the coauthor of one book. She has published more than 180 articles. Her current research interests include wireless networks, underwater acoustic networks, adaptive networks, and network security. Dr. Chen is a member of ACM and a Senior Member of CIC.

**LEI XIE** (Member, IEEE) received the B.S. degree in electronic engineering, the M.S. degree in communications and electronic systems, and the Ph.D. degree in communications and information systems from Zhejiang University, Hangzhou, China, in 1994, 1997, and 2002, respectively. Since 1997, he has been with Zhejiang University, where he is currently an Associate Professor with the College of Information Science and Electronic Engineering. His current research interests include information theory and coding, network security, and multimedia streaming in heterogeneous wireless networks. Dr. Xie is a Senior Member of CIC.

**KUANG WANG** received the B.S. degree in electronic engineering and the Ph.D. degree in communications and information systems from Zhejiang University, Hangzhou, China, in 1989 and 1995, respectively. He is currently a Professor with the College of Information Science and Electronic Engineering, Zhejiang University. His current research interests include digital signal processing, coded modulation systems, and resource management in wireless networks.