Consensus based Detection in the Presence of Data Falsification Attacks

Bhavya Kailkhura, *Student Member, IEEE*, Swastik Brahma, *Member, IEEE*,

Pramod K. Varshney, *Fellow, IEEE*

arXiv:1504.03413v1 [cs.SY] 14 Apr 2015

Abstract

This paper considers the problem of detection in distributed networks in the presence of data falsification (Byzantine) attacks. Detection approaches considered in the paper are based on fully distributed consensus algorithms, where all of the nodes exchange information only with their neighbors in the absence of a fusion center. In such networks, we characterize the negative effect of Byzantines on the steady-state and transient detection performance of the conventional consensus based detection algorithms. To address this issue, we study the problem from the network designer's perspective. More specifically, we first propose a distributed weighted average consensus algorithm that is robust to Byzantine attacks. We show that, under reasonable assumptions, the global test statistic for detection can be computed locally at each node using our proposed consensus algorithm. We exploit the statistical distribution of the nodes' data to devise techniques for mitigating the influence of data falsifying Byzantines on the distributed detection system. Since some parameters of the statistical distribution of the nodes' data might not be known *a priori*, we propose learning based techniques to enable an adaptive design of the local fusion or update rules.

Index Terms

distributed detection, consensus algorithms, data falsification attacks, Byzantines

I. INTRODUCTION

Distributed detection is a well studied topic in the detection theory literature [1]–[3]. The traditional distributed detection framework comprises of a group of spatially distributed nodes which acquire the observations regarding the phenomenon of interest and send them to the fusion center (FC) where a global decision is made. However, in many scenarios a centralized FC may not be available or in large networks, the FC can become an information bottleneck that may cause degradation of system performance, and may even lead to system failure. Also, due to the distributed nature of future communication networks, and various practical constraints, e.g., absence of the FC, transmit power or hardware constraints and dynamic characteristic of wireless medium, it may be desirable to employ alternate peer-to-peer local information exchange in order to reach a global decision. One such distributed approach for peer-to-peer local information exchange and inference is the use of a consensus algorithm [27].

Recently, distributed detection based on consensus algorithms has been explored in [4]–[9]. In consensus based detection approaches, each node communicates only with its neighbors and updates its local state information about the phenomenon (summary statistic) by a local fusion rule that employs a weighted combination of its own value and those received from its neighbors. Nodes continue with this consensus iteration until the whole network converges to a steady-state value which is the global test statistic. In particular, the authors in [5], [6] considered average consensus based distributed detection and emphasized network designs based on the small world phenomenon for faster convergence [7]. A bio-inspired consensus scheme was introduced for spectrum sensing in [8]. However, these consensus-based fusion algorithms only ensure equal gain combining of local measurements. The authors in [9] proposed to use distributed weighted fusion algorithms for cognitive radio spectrum sensing. They showed that weighted average consensus based schemes outperform average consensus based schemes and achieve much better detection performance than the equal gain combining based schemes. However, the weighted average consensus based detection schemes are quite vulnerable to different types of attacks. One typical attack on such networks is a Byzantine attack. While Byzantine attacks (originally proposed in [10]) may, in general, refer to many types of malicious behavior, our focus in this paper is on data-falsification attacks [11]–[18]. Thus far, research on detection in the presence of Byzantine attacks has predominantly focused on addressing these attacks under the centralized

model [13], [14], [18], [19]. A few attempts have been made to address the security threats in the distributed or consensus based schemes in recent research [20]–[25]. Most of these existing works on countering Byzantine or data falsification attacks in distributed networks rely on a threshold for detecting Byzantines. The main idea is to exclude nodes from neighbors list whose state information deviates significantly from the mean value. In [22] and [25], two different defense schemes against data falsification attacks for distributed consensus-based detection were proposed. In [22], the scheme eliminates the state value with the largest deviation from the local mean at each iteration step and, therefore, it can only deal with the situation in which only one Byzantine node exists. It excludes one state value even if there is no Byzantine node. In [25], the vulnerability of distributed consensus-based spectrum sensing was analyzed and an outlier detection algorithm with an adaptive threshold was proposed. The authors in [24] proposed a Byzantine mitigation technique based on adaptive local thresholds. This scheme mitigates the misbehavior of Byzantine nodes and tolerates the occasional large deviation introduced by honest users. It adaptively reduces the corresponding coefficients so that the Byzantines will eventually be isolated from the network.

Excluding the Byzantine nodes from the fusion process may not be the best strategy from the network designer's perspective. As shown in our earlier work [18] in the context of distributed detection with one-bit measurements under a centralized model, an intelligent way to improve the performance of the network is to use the information of the identified Byzantines to the network's benefit. More specifically, learning based techniques have the potential to outperform the existing exclusion based techniques. In this paper, we pursue such a design philosophy in the context of raw data based fusion in decentralized networks.

To design methodologies for defending against Byzantine attacks, fundamental challenges that arise are two-fold. First, how do nodes recognize the presence of attackers? Second, after identification of an attacker or group of attackers, how do nodes adapt their operating parameters? Due to the large number of nodes and complexity of the distributed network, we develop and analyze schemes that would update their own operating parameters autonomously. Our approach further introduces an adaptive fusion based detection algorithm which supports the learning of the attacker's behavior. Our scheme differs from all existing work on Byzantine mitigation based on exclusion strategies [21]–[25], where the only defense is to identify and exclude the attackers from the consensus process.

A. Main Contributions

In this paper, we focus on the susceptibility and protection of consensus based detection algorithms. Our main contributions are summarized as follows:

- We characterize the effect of Byzantines on the steady-state performance of the conventional consensus based detection algorithms. More specifically, we quantify the minimum fraction of Byzantines needed to make the deflection coefficient of the global statistic equal to zero.
- Using probability of detection and probability of false alarm as measures of detection performance, we investigate the degradation of transient detection performance of the conventional consensus algorithms with Byzantines.
- We propose a robust distributed weighted average consensus algorithm and obtain closed-form expressions for optimal weights to mitigate the effect of data falsification attacks.
- Finally, we propose a technique based on the expectation-maximization algorithm and maximum likelihood estimation to learn the operating parameters (or weights) of the nodes in the network to enable an adaptive design of the local fusion or update rules.

The rest of the paper is organized as follows. In Sections II and III, we introduce our system model and Byzantine attack model, respectively. In Section IV, we study the security performance of weighted average consensus based detection schemes. In Section V, we propose a protection mechanism to mitigate the effect of data falsification attacks on consensus based detection schemes. Finally, Section VI concludes the paper.

II. SYSTEM MODEL

First, we define the network model used in this paper.

A. Network Model

We model the network topology as an undirected graph $G = (V, E)$, where $V = \{v_1, \cdots, v_N\}$ represents the set of nodes in the network with $|V| = N$. The set of communication links in the network correspond to the set of edges E, where $(v_i, v_j) \in E$, if and only if there is a communication link between v_i and v_j (so that, v_i and v_j can directly communicate with each other). The adjacency matrix A of the graph is defined as

$$a_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$
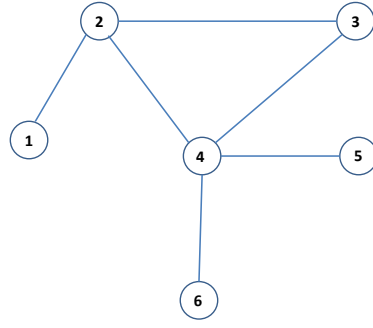
Figure 1. A distributed network with 6 nodes

The neighborhood of a node i is defined as

$$\mathcal{N}_i = \{v_j \in V : (v_i, v_j) \in E\}, \forall i \in \{1, 2, \cdots, N\}.$$

The degree of a node v_i in a graph G, denoted by d_i, is the number of edges in E which include v_i as an endpoint, i.e., $d_i = \sum_{j=1}^{N} a_{ij}$.

The degree matrix D is defined as a diagonal matrix with $\text{diag}(d_1, \cdots, d_N)$ and the Laplacian matrix L is defined as

$$l_{ij} = \begin{cases} d_i & \text{if } j = i, \\ -a_{ij} & \text{otherwise.} \end{cases}$$

or, in other words, $L = D - A$. For example, consider a network with six nodes trying to reach consensus (see Figure 1). The Laplacian matrix L for this network is given by

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 4 & -1 & -1 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}.$$

The consensus based distributed detection scheme usually contains three phases: sensing, information fusion, and decision making. In the sensing phase, each node acquires the summary statistic about the phenomenon of interest. In this paper, we adopt the energy detection method so that the local summary statistic is the received signal energy. Next, in the information fusion

phase, each node communicates with its neighbors to update their state values (summary statistic) and continues with the consensus iteration until the whole network converges to a steady state which is the global test statistic. Finally, in the decision making phase, nodes make their own decisions about the presence of the phenomenon. Next, we describe each of these phases in more detail.

B. Sensing Phase

We consider an N-node network using the energy detection scheme [26]. For the ith node, the sensed signal x_i^k at time instant k is given by

$$x_i^k = \begin{cases} n_i^k, & \text{under } H_0 \\ h_i s^k + n_i^k & \text{under } H_1 \end{cases}$$

where h_i is the channel gain, s^k is the signal at time instant k, n_i^k is AWGN, i.e., $n_i^k \sim N(0, \sigma_i^2)$ and independent across time. Each node i calculates a summary statistic Y_i over a detection interval of M samples, i.e.,

$$Y_i = \sum_{k=1}^{M} |x_i^k|^2$$

where M is determined by the time-bandwidth product. Since Y_i is the sum of the square of M i.i.d. Gaussian random variables, it can be shown that $\frac{Y_i}{\sigma_i^2}$ follows a central chi-square distribution with M degrees of freedom (χ_M^2) under H_0, and, a non-central chi-square distribution with M degrees of freedom and parameter η_i under H_1, i.e.,

$$\frac{Y_i}{\sigma_i^2} \sim \begin{cases} \chi_M^2, & \text{under } H_0 \\ \chi_M^2(\eta_i) & \text{under } H_1 \end{cases}$$

where $\eta_i = E_s |h_i|^2 / \sigma_i^2$ is the local SNR at the ith node and $E_s = \sum_{k=1}^{M} |s^k|^2$ represents the sensed signal energy over M detection instants. Note that the local SNR is M times the average SNR at the output of the energy detector, which is $\frac{E_s |h_i|^2}{M \sigma_i^2}$.

C. Information Fusion Phase

Next, we give a brief introduction to conventional consensus algorithms [27]. Consensus is reached in two steps.

Step 1: All nodes establish communication links with their neighbors, and broadcast their information state $x_i(0) = Y_i$.

Step 2: Each node updates its local state information by a local fusion rule (weighted combination of its own value and those received from its neighbors). We denote node i's updated information at iteration k by $x_i(k)$. Node i continues to broadcast information $x_i(k)$ and update its local information state until consensus is reached. This information state updating process can be written in a compact form as

$$x_i(k+1) = x_i(k) + \frac{\epsilon}{w_i} \sum_{j \in \mathcal{N}_i} (x_j(k) - x_i(k)) \tag{1}$$

where ϵ is the time step and w_i is the weight assigned to node i's information. Using the notation $x(k) = [x_1(k), \cdots, x_N(k)]^T$, network dynamics in the matrix form can be represented as,

$$x(k+1) = Wx(k)$$

where, $W = I - \epsilon \, \mathrm{diag}(1/w_1, \cdots, 1/w_N)L$ is referred to as a Perron matrix. The consensus algorithm is nothing but a local fusion or update rule that fuses the nodes' local information state with information coming from neighbor nodes and every node asymptotically reaches the same information state for arbitrary initial values.

D. Decision Making Phase

The final information state after reaching consensus for the above consensus algorithm will be the weighted average of the initial states of all the nodes [27] or $x_i^* = \sum_{i=1}^{N} w_i Y_i / \sum_{i=1}^{N} w_i$, $\forall i$. Average consensus can be seen as a special case of weighted average consensus with $w_i = w$, $\forall i$. After the whole network reaches a consensus, each node makes its own decision about the hypothesis using a predefined threshold λ[1]

$$\text{Decision} = \begin{cases} H_1 & \text{if } x_i^* > \lambda \\ H_0 & \text{otherwise} \end{cases}$$

[1]In practice, parameters such as threshold λ and consensus time step ϵ can be set off-line. This study is beyond the scope of this work.

where weights are given by [9]

$$w_i = \frac{\eta_i/\sigma_i^2}{\sum_{i=1}^{N} \eta_i/\sigma_i^2}. \tag{2}$$

Note that, after reaching consensus $x_i^* = \Lambda, \forall i$. Thus, in rest the of the paper, Λ is referred to as the final test statistic.

Next, we discuss Byzantine attacks on consensus based detection schemes and analyze the performance degradation of the weighted average consensus based detection algorithm due to these attacks.

III. ATTACKS ON CONSENSUS BASED DETECTION ALGORITHMS

When there are no adversaries in the network, we noted in the last section that consensus can be accomplished to the weighted average of arbitrary initial values by having the nodes use the update strategy $x(k+1) = Wx(k)$ with an appropriate weight matrix W. Suppose, however, that instead of broadcasting the true sensing statistic Y_i and applying the update strategy (1), some nodes (referred to as Byzantines) deviate from the prescribed strategies. Accordingly, Byzantines can attack in two ways: data falsification (nodes falsify their initial data or weight values) and consensus disruption (nodes do not follow update rule given by (1)). More specifically, Byzantine node i can do the following

$$\text{Data falsification:} \quad x_i(0) = Y_i + \Delta_i, \quad \text{or} \quad w_i \to \tilde{w}_i$$

$$\text{Consensus disruption:} \quad x_i(k+1) = x_i(k) + \frac{\epsilon}{w_i} \sum_{j \in \mathcal{N}_i} (x_j(k) - x_i(k)) + u_i(k),$$

where $(\Delta_i, \tilde{w}_i)$ and $u_i(k)$ are introduced at the initialization step and at the update step k, respectively. The attack model considered above is extremely general, and allows Byzantine node i to update its value in a completely arbitrary manner (via appropriate choices of $(\Delta_i, \tilde{w}_i)$, and $u_i(k)$, at each time step). An adversary performing consensus disruption attack has the objective to disrupt the consensus operation. However, consensus disruption attacks can be easily detected because of the nature of the attack. The identification of consensus disruption attackers has been investigated in the past literature (e.g., see [7], [28]) where control theoretic techniques were developed to identify disruption attackers in a single consensus iteration. Knowing the existence of such an identification mechanism, a smart adversary will aim to disguise itself while degrading the detection performance. In contrast to disruption attackers, data falsification attackers are more

capable and can manage to disguise themselves while degrading the detection performance of the network by falsifying their data. Susceptibility and protection of consensus strategies to data falsification attacks has received scant attention, and this is the focus of our work. In this paper, we assume that an attacker performs only a data falsification attack by introducing $(\Delta_i, \tilde{w}_i)$ during initialization. We exploit the statistical distribution of the initial values and devise techniques to mitigate the influence of Byzantines on the distributed detection system.

A. Data Falsification Attack

In data falsification attacks, attackers try to manipulate the final test statistic (i.e., $\Lambda = (\sum_{i=1}^{N} w_i Y_i)/(\sum_{i=1}^{N} w_i)$) in a way that the detection performance is degraded. We consider a network with N nodes that uses Algorithm (1) for reaching consensus. Algorithm (1) can be interpreted as, weight w_i, given to node i's data Y_i in the final test statistic, is assigned by node i itself. So by falsifying initial values Y_i or weights w_i, the attackers can manipulate the final test statistic. Detection performance will be degraded because Byzantine nodes can always set a higher weight to their manipulated information. Thus, the final statistic's value across the whole network will be dominated by the Byzantine node's local statistic that will lead to degraded detection performance.

Next, we define a mathematical model for data falsification attackers. We analyze the degradation in detection performance of the network when Byzantines falsify their initial values Y_i for fixed arbitrary weights $\tilde{w}_i$.

B. Attack Model

The objective of Byzantines is to degrade the detection performance of the network by falsifying their data (Y_i, w_i). By assuming that Byzantines are intelligent and know the true hypothesis, we analyze the worst case detection performance of the data fusion schemes. We consider the case when weights of the Byzantines have already been tampered to $\tilde{w}_i$ and analyze the effect of falsifying the initial values Y_i. This analysis provides the most favorable case from the point of view of Byzantines and yields the maximum performance degradation that the Byzantines can cause. Now a mathematical model for a Byzantine attack is presented. Byzantines tamper with their initial values Y_i and send $\tilde{Y}_i$ such that the detection performance is degraded.

Under H_0:

$$\tilde{Y}_i = \begin{cases} Y_i + \Delta_i & \text{with probability } P_i \\ Y_i & \text{with probability } (1 - P_i) \end{cases}$$

Under H_1:

$$\tilde{Y}_i = \begin{cases} Y_i - \Delta_i & \text{with probability } P_i \\ Y_i & \text{with probability } (1 - P_i) \end{cases}$$

where P_i is the attack probability and Δ_i is a constant value which represents the attack strength, which is zero for honest nodes. As we show later, Byzantine nodes will use a large value of Δ_i so that the final statistic's value is dominated by the Byzantine node's local statistic that will lead to a degraded detection performance. We use deflection coefficient [29] to characterize the security performance of the detection scheme due to its simplicity and its strong relationship with the global detection performance. Deflection coefficient of the global test statistic is defined as: $\mathcal{D}(\Lambda) = \dfrac{(\mu_1 - \mu_0)^2}{\sigma_{(0)}^2}$, where $\mu_k = \mathbb{E}[\Lambda|H_k]$ is the conditional mean and $\sigma_{(k)}^2 = \mathbb{E}[(\Lambda - \mu_k)^2|H_k]$ is the conditional variance. The deflection coefficient is also closely related to other performance measures, e.g., the Receiver Operating Characteristics (ROC) curve. In general, the detection performance monotonically increases with an increasing value of the deflection coefficient. We define the critical point of the distributed detection network as the minimum fraction of Byzantine nodes needed to make the deflection coefficient of global test statistic equal to zero (in which case, we say that the network becomes *blind*) and denote it by α_{blind}. We assume that the communication between nodes is error-free and our network topology is fixed during the whole consensus process and, therefore, consensus can be reached without disruption.

In the next section, we analyze the security performance of consensus based detection schemes in the presence of data falsifying Byzantines.

IV. Performance analysis of consensus based detection algorithms

In this section, we analyze the effect of data falsification attacks on conventional consensus based detection algorithms.

First, in Section IV-A, we characterize the effect of Byzantines on the steady-state performance of the consensus based detection algorithms and determine α_{blind}. Next, in Section IV-B, using probability of detection and probability of false alarm as measures of detection performance, we

$$\mu_0 = \sum_{i=1}^{N_1} \left[P_i \frac{\tilde{w}_i}{\text{sum}(w)}(M\sigma_i^2 + \Delta_i) + (1 - P_i)\frac{\tilde{w}_i}{\text{sum}(w)}(M\sigma_i^2) \right] + \sum_{i=N_1+1}^{N} \left[\frac{w_i}{\text{sum}(w)}(M\sigma_i^2) \right] \quad (3)$$

$$\mu_1 = \sum_{i=1}^{N_1} \left[P_i \frac{\tilde{w}_i}{\text{sum}(w)}((M + \eta_i)\sigma_i^2 - \Delta_i) + (1 - P_i)\frac{\tilde{w}_i}{\text{sum}(w)}((M + \eta_i)\sigma_i^2) \right]$$
$$+ \sum_{i=N_1+1}^{N} \left[\frac{w_i}{\text{sum}(w)}((M + \eta_i)\sigma_i^2) \right] \quad (4)$$

$$\sigma_{(0)}^2 = \sum_{i=1}^{N_1} \left(\frac{\tilde{w}_i}{\text{sum}(w)} \right)^2 \left[P_i(1 - P_i)\Delta_i^2 + 2M\sigma_i^4 \right] + \sum_{i=N_1+1}^{N} \left(\frac{w_i}{\text{sum}(w)} \right)^2 \left[2M\sigma_i^4 \right] \quad (5)$$

investigate the degradation of transient detection performance of the consensus algorithms with Byzantines.

A. Steady-State Performance Analysis with Byzantines

Without loss of generality, we assume that the nodes corresponding to the first N_1 indices $i = 1, \cdots, N_1$ are Byzantines and the rest corresponding to indices $i = N_1 + 1, \cdots, N$ are honest nodes. Let us define $w = [\tilde{w}_1, \cdots, \tilde{w}_{N_1}, w_{N_1+1} \cdots, w_N]^T$ and $\text{sum}(w) = \sum_{i=1}^{N_1} \tilde{w}_i + \sum_{i=N_1+1}^{N} w_i$.

Lemma 1. *For data fusion schemes, the condition to blind the network or to make the deflection coefficient zero is given by*

$$\sum_{i=1}^{N_1} \tilde{w}_i(2P_i\Delta_i - \eta_i\sigma_i^2) = \sum_{i=N_1+1}^{N} w_i\eta_i\sigma_i^2.$$

Proof: The local test statistic Y_i has the mean

$$mean_i = \begin{cases} M\sigma_i^2 & \text{if } H_0 \\ (M + \eta_i)\sigma_i^2 & \text{if } H_1 \end{cases}$$
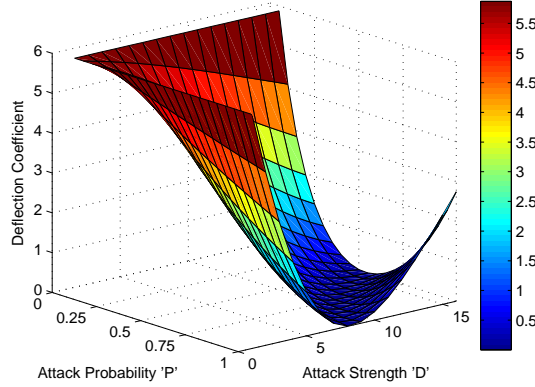
and the variance

Figure 2. Deflection Coefficient as a function of attack parameters P and D.

$$
Var_i = \begin{cases}
2M\sigma_i^4 & \text{if } H_0 \\
2(M + 2\eta_i)\sigma_i^4 & \text{if } H_1.
\end{cases}
$$

The goal of Byzantine nodes is to make the deflection coefficient as small as possible. Since the Deflection Coefficient is always non-negative; the Byzantines seek to make $\mathcal{D}(\Lambda) = \dfrac{(\mu_1 - \mu_0)^2}{\sigma_{(0)}^2} = 0$. The conditional mean $\mu_k = \mathbb{E}[\Lambda | H_k]$ and conditional variance $\sigma_{(0)}^2 = \mathbb{E}[(\Lambda - \mu_0)^2 | H_0]$ of the global test statistic, $\Lambda = (\sum_{i=1}^{N_1} \tilde{w}_i \tilde{Y}_i + \sum_{i=N_1+1}^{N} w_i Y_i)/(\text{sum}(w))$, can be computed and are given by (3), (4) and (5), respectively. After substituting values from (3), (4) and (5), the condition to make $\mathcal{D}(\Lambda) = 0$ becomes

$$
\sum_{i=1}^{N_1} \tilde{w}_i(2P_i\Delta_i - \eta_i\sigma_i^2) = \sum_{i=N_1+1}^{N} w_i\eta_i\sigma_i^2
$$

∎

Note that, when $w_i = \tilde{w}_i = z, \eta_i = \eta, \sigma_i = \sigma, P_i = P, \Delta_i = D, \ \forall i$, the blinding condition simplifies to $\dfrac{N_1}{N} = \dfrac{1}{2}\dfrac{\eta\sigma^2}{PD}$. This condition indicates that by appropriately choosing attack parameters (P, D), an adversary needs less than 50% of sensing data falsifying Byzantines to make the deflection coefficient zero.

Next, to gain insights into the solution, we present some numerical results in Figure 2. We plot the deflection coefficient of global test statistic as a function of attack parameters $P_i = P, \Delta_i = D, \forall i$. We consider a 6-node network with the topology given by the undirected graph

shown in Figure 1 to detect a phenomenon. Nodes 1 and 2 are considered to be Byzantines. Channel gains of the nodes are assumed to be $h = [0.8, 0.7, 0.72, 0.61, 0.69, 0.9]$ and weights are given by (2). We also assume that $M = 12$, $E_s = 5$ and $\sigma_i^2 = 1$. Notice that, the deflection coefficient is zero when the condition in Lemma 1 is satisfied. Another observation to make is that the deflection coefficient can be made zero even when only two out of six nodes are Byzantines. Thus, by appropriately choosing attack parameters (P, D), less than 50% of data falsifying Byzantines are needed to blind the network.

B. Transient Performance Analysis with Byzantines

Next, we analyze the detection performance of the data fusion schemes, denoted as $x(t+1) = W^t x(0)$, as a function of consensus iteration t in the presence of Byzantines. For analytical tractability, we assume that $P_i = P$, $\forall i$. We denote by w_{ji}^t the element of matrix W^t in the jth row and ith column. Using these notations, we calculate the probability of detection and the probability of false alarm at the jth node at consensus iteration t.

For sufficiently large M, the distribution of Byzantine's data $\tilde{Y}_i$ given H_k is a Gaussian mixture which comes from $\mathcal{N}((\mu_{1k})_i, (\sigma_{1k})_i^2)$ with probability $(1 - P)$ and from $\mathcal{N}((\mu_{2k})_i, (\sigma_{2k})_i^2)$ with probability P, where $\mathcal{N}$ denotes the normal distribution and

$$(\mu_{10})_i = M\sigma_i^2, \ (\mu_{20})_i = M\sigma_i^2 + \Delta_i$$

$$(\mu_{11})_i = (M + \eta_i)\sigma_i^2, \ (\mu_{21})_i = (M + \eta_i)\sigma_i^2 - \Delta_i$$

$$(\sigma_{10})_i^2 = (\sigma_{20})_i^2 = 2M\sigma_i^4, \ and \ (\sigma_{11})_i^2 = (\sigma_{21})_i^2 = 2(M + \eta_i)\sigma_i^4.$$

Now, the probability density function (PDF) of $x_{ji}^t = w_{ji}^t \tilde{Y}_i$ conditioned on H_k can be derived as

$$f(x_{ji}^t | H_k) = (1 - P)\phi(w_{ji}^t(\mu_{1k})_i, (w_{ji}^t(\sigma_{1k})_i)^2)$$

$$+ P\phi(w_{ji}^t(\mu_{2k})_i, (w_{ji}^t(\sigma_{2k})_i)^2) \tag{6}$$

where $\phi(x | \mu, \sigma^2)$ (for notational convenience denoted as $\phi(\mu, \sigma^2)$) is the PDF of $X \sim \mathcal{N}(\mu, \sigma^2)$ and $\phi(x | \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$. Next, for clarity of exposition, we first derive our results for a small network with two Byzantine nodes and one honest node. Later we generalize our

results for an arbitrary number of nodes, N.

Notice that, for the three node case, the transient test statistic $\tilde{\Lambda}_j^t = w_{j1}^t \tilde{Y}_1 + w_{j2}^t \tilde{Y}_2 + w_{j3}^t Y_3$, is a summation of independent random variables. The conditional PDF of $X_{ji}^t = w_{ji}^t \tilde{Y}_i$ is given in (6). Notice that, PDF of $\tilde{\Lambda}_j^t$ is the convolution ($*$) of $f(x_{j1}^t) = (1 - P)\phi(\mu_1^1, (\sigma_1^1)^2) + P\phi(\mu_1^2, (\sigma_1^2)^2)$, $f(x_{j2}^t) = (1 - P)\phi(\mu_2^1, (\sigma_2^1)^2) + P\phi(\mu_2^2, (\sigma_2^2)^2))$ and $f(x_{j3}^t) = \phi(\mu_3^1, (\sigma_3^1)^2)$.

$$f(z_j^t) = f(x_{j1}^t) * f(x_{j2}^t) * f(x_{j3}^t)$$

$$f(z_j^t) = [(1 - P)\phi(\mu_1^1, (\sigma_1^1)^2) + P\phi(\mu_1^2, (\sigma_1^2)^2)]*$$

$$[(1 - P)\phi(\mu_2^1, (\sigma_2^1)^2) + P\phi(\mu_2^2, (\sigma_2^2)^2)] * \phi(\mu_3^1, (\sigma_3^1)^2)$$

$$= (1 - P)^2[\phi(\mu_1^1, (\sigma_1^1)^2) * \phi(\mu_2^1, (\sigma_2^1)^2) * \phi(\mu_3^1, (\sigma_3^1)^2)]$$

$$+(P)^2[\phi(\mu_1^2, (\sigma_1^2)^2) * \phi(\mu_2^2, (\sigma_2^2)^2)) * \phi(\mu_3^1, (\sigma_3^1)^2)]$$

$$+P(1 - P)[\phi(\mu_1^2, (\sigma_1^2)^2) * \phi(\mu_2^1, (\sigma_2^1)^2) * \phi(\mu_3^1, (\sigma_3^1)^2)]$$

$$+(1 - P)P[\phi(\mu_1^1, (\sigma_1^1)^2) * \phi(\mu_2^2, (\sigma_2^2)^2) * \phi(\mu_3^1, (\sigma_3^1)^2)]$$

Now, using the fact that convolution of two normal PDFs $\phi(\mu_i, \sigma_i^2)$ and $\phi(\mu_j, \sigma_j^2)$ is again normally distributed with mean $(\mu_i + \mu_j)$ and variance $(\sigma_i^2 + \sigma_j^2)$, we can derive the results below.

$$f(z_j^t) = (1 - P)^2[\phi(\mu_1^1 + \mu_2^1 + \mu_3^1, (\sigma_1^1)^2 + (\sigma_2^1)^2 + (\sigma_3^1)^2)]$$

$$+P^2[\phi(\mu_1^2 + \mu_2^2 + \mu_3^1, (\sigma_1^2)^2 + (\sigma_2^2)^2 + (\sigma_3^1)^2)]$$

$$+P(1 - P)[\phi(\mu_1^2 + \mu_2^1 + \mu_3^1, (\sigma_1^2)^2 + (\sigma_2^1)^2 + (\sigma_3^1)^2)]$$

$$+(1 - P)P[\phi(\mu_1^1 + \mu_2^2 + \mu_3^1, (\sigma_1^1)^2 + (\sigma_2^2)^2 + (\sigma_3^1)^2)].$$

Due to the probabilistic nature of the Byzantine's behavior, it may behave as an honest node with a probability $(1 - P_i)$. Let S denote the set of all combinations of such Byzantine strategies:

$$S = \{\{b_1, b_2\}, \{h_1, b_2\}, \{b_1, h_2\}, \{h_1, h_2\}\} \tag{7}$$

where by b_i we mean that Byzantine node i behaves as a Byzantine and by h_i we mean that Byzantine node i behaves as an honest node. Let $A_s \in U$ denote the indices of honest nodes in the strategy combination s, then, from (7) we have

$$U = \{A_1 = \{\}, A_2 = \{1\}, A_3 = \{2\}, A_4 = \{1, 2\}\}$$

$$U^c = \{A_1^c = \{1, 2\}, A_2^c = \{2\}, A_3^c = \{1\}, A_4^c = \{\}\}$$

where $\{\}$ is used to denote the null set and m_s to denote the cardinality of subset $A_s \in U$. Using these notations, we generalize our results for any arbitrary N.

Lemma 2. *The test statistic of node j at consensus iteration t, i.e., $\tilde{\Lambda}_j^t = \sum_{i=1}^{N_1} w_{ji}^t \tilde{Y}_i + \sum_{i=N_1+1}^{N} w_{ji}^t Y_i$ is a Gaussian mixture with PDF*

$$f(\tilde{\Lambda}_j^t | H_k) = \sum_{A_s \in U} P^{N_1 - m_s}(1 - P)^{m_s} \phi \left((\mu_k)_{A_s} + \sum_{i=N_1+1}^{N} w_{ji}^t(\mu_{1k})_i, \sum_{i=1}^{N} (w_{ji}^t(\sigma_{1k})_i)^2 \right)$$

$$with \ (\mu_k)_{A_s} = \sum_{u \in A_s} w_{ju}^t(\mu_{1k})_j + \sum_{u \in A_s^c} w_{ju}^t(\mu_{2k})_j.$$

The performance of the detection scheme in the presence of Byzantines can be represented in terms of the probability of detection and the probability of false alarm of the network.

Proposition 1. *The probability of detection and the probability of false alarm of node j at consensus iteration t in the presence of Byzantines can be represented as*

$$P_d^t(j) = \sum_{A_s \in U} P^{N_1 - m_s}(1 - P)^{m_s} Q \left(\frac{\lambda - (\mu_1)_{A_s} - \sum_{i=N_1+1}^{N} w_{ji}^t(\mu_{11})_i}{\sqrt{\sum_{i=1}^{N} (w_{ji}^t(\sigma_{11})_i)^2}} \right),$$

$$P_f^t(j) = \sum_{A_s \in U} P^{N_1 - m_s}(1 - P)^{m_s} Q \left(\frac{\lambda - (\mu_0)_{A_s} - \sum_{i=N_1+1}^{N} w_{ji}^t(\mu_{10})_i}{\sqrt{\sum_{i=1}^{N} (w_{ji}^t(\sigma_{10})_i)^2}} \right).$$

Remark 1. *Notice that, the expressions of probability of detection $P_d^t(j)$ and probability of false alarm $P_f^t(j)$ for the N_1 Byzantine node case involves 2^{N_1} combinations (cardinality of U is 2^{N_1}). It, however, can be represented compactly by vectorizing the expressions, i.e.,*

$$P_d^t(j) = \mathbf{1}^\mathbf{T} \left(\mathbf{b} \otimes Q \left(\frac{\lambda \mathbf{1} \text{-} \boldsymbol{\mu_1}}{\sqrt{\sum_{i=1}^{N} (w_i(\sigma_{10})_i)^2}} \right) \right)$$

with $\boldsymbol{\mu_1} = [A\boldsymbol{w_j^t}\boldsymbol{\mu_{11}} + A^c\boldsymbol{w_j^t}\boldsymbol{\mu_{21}} - \sum_{i=N_1+1}^{N} w_{ji}^t(\mu_{11})_i], B = (1-P)A + PA^c$ *and* $\mathbf{b} = [\mathbf{B^1} \otimes \cdots \otimes \mathbf{B^{N_1}}]$, *where boldface letters represent vectors,* $\otimes$ *symbol represents element-wise multiplication,* $Q(\cdot)$ *represents element wise* Q *function operation, i.e.,* $Q(x_1, \cdots, x_{N_1}) = [Q(x_1), \cdots, Q(x_{N_1})]^T$, $\mathbf{B^i}$ *is* i*th column of matrix* B, $\boldsymbol{w_j^t}\boldsymbol{\mu_{u1}} = [w_{j1}^t\mu_{u1}^1, \cdots, w_{jN_1}^t\mu_{u1}^{N_1}]^T$, *matrix* $A_{(2^{N_1}*N_1)}$ *is the binary representation of decimal numbers from* 0 *to* $N_1 - 1$ *and* A^c *is the matrix after interchanging* 1 *and* 0 *in matrix* A.

Similarly, the expression for the probability of false alarm $P_f^t(j)$ *can be vectorized into a compact form.*

Next, to gain insights into the results given in Proposition 1, we present some numerical results in Figures 3 and 4. We consider the 6-node network shown in Figure 1 where the nodes employ the consensus algorithm 1 with $\epsilon = 0.6897$ to detect a phenomenon. Nodes 1 and 2 are considered to be Byzantines. We also assume that $\eta_i = 10$, $\sigma_i^2 = 2$, $\lambda = 33$ and $w_i = 1$. Attack parameters are assumed to be $(P_i, \Delta_i) = (0.5, 6)$ and $\tilde{w}_i = 1.1$. To characterize the transient performance of the weighted average consensus algorithm, in Figure 3(a), we plot the probability of detection as a function of the number of consensus iterations when Byzantines are not falsifying their data, i.e., $(\Delta_i = 0, \tilde{w}_i = w_i)$. Next, in Figure 3(b), we plot the probability of detection as a function of the number of consensus iterations in the presence of Byzantines. It can be seen that the detection performance degrades in the presence of Byzantines. In Figure 4(a), we plot the probability of false alarm as a function of the number of consensus iterations when Byzantines are not falsifying their data, i.e., $(\Delta_i = 0, \tilde{w}_i = w_i)$. Next, in Figure 4(b), we plot the probability of false alarm as a function of the number of consensus iterations in the presence of Byzantines. From both Figures 3 and 4, it can be seen that the Byzantine attack can severely degrade transient detection performance.

From the discussion in this section, we can see that Byzantines can severely degrade both the steady-state and the transient detection performance of conventional consensus based detection algorithms. As mentioned earlier, a data falsifying Byzantine i can tamper its weight w_i as well as its sensing data Y_i to degrade detection performance. One approach to mitigate the effect of sensing data falsification is to assign weights based on the quality of the data. In other words, lower weight is assigned to the data of the node identified as a Byzantine. However, to implement this approach one has to address the following two issues.
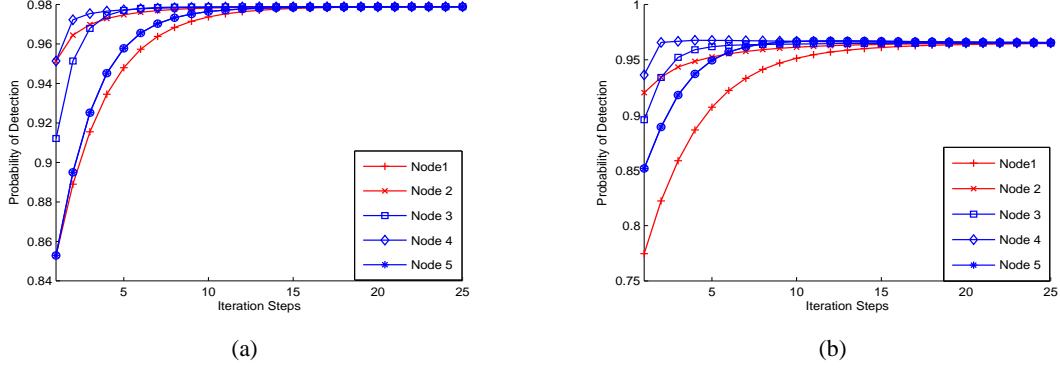
Figure 3. (a) Probability of detection as a function of consensus iteration steps. (b) Probability of detection as a function of consensus iteration steps with Byzantines.
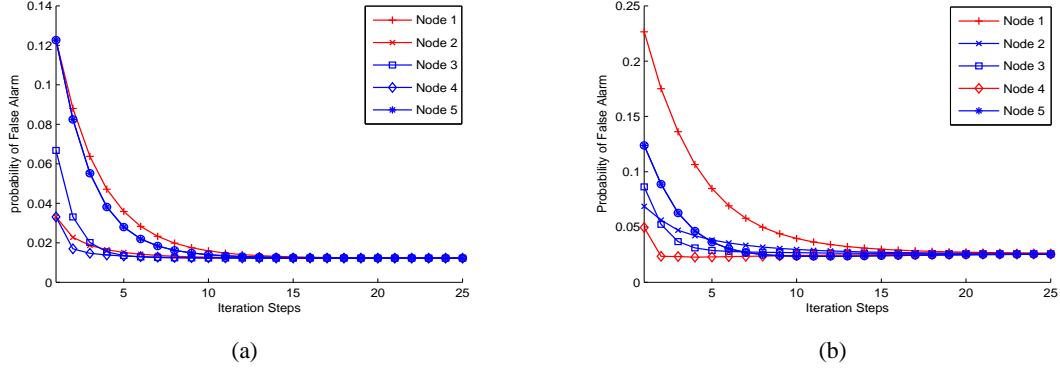

Figure 4. (a) Probability of false alarm as a function of consensus iteration steps. (b) Probability of false alarm as a function of consensus iteration steps with Byzantines.

First, in the conventional weighted average consensus algorithm, weight w_i given to node i's data is assigned by the node itself. Thus, a Byzantine node can always set a higher weight to its manipulated information and the final statistics will be dominated by the Byzantine nodes' local statistic that will lead to degraded detection performance. It will be impossible for any algorithm to detect this type of malicious behavior, since any weight that a Byzantine chooses for itself is a legitimate value that could also have been chosen by a node that is functioning correctly. Thus, the conventional consensus algorithm cannot be used in the presence of an attacker.

Second, as will be seen later, the optimal weights assigned to nodes' sensing data depend on the following unknown parameters: identity of the nodes (i.e., honest or Byzantine) and underlying statistical distribution of the nodes' data.

In the next section, we address these concerns by proposing a learning based robust weighted

average consensus algorithm.

V. A ROBUST CONSENSUS BASED DETECTION ALGORITHM

In order to address the first issue, we propose a consensus algorithm in which the weight for node i's information is assigned (or updated) by neighbors of the node i rather than by node i itself. Note that, networks deploying such an algorithm is more robust to weight manipulation because if a Byzantine node j wants to lower the weight assigned to the data of its honest neighbor i in the global test statistic, it has to make sure that a majority of the neighbors of i put the same lower weight as j. In other words, every honest node should have majority of its neighbors that are Byzantines, otherwise, it can be treated as a consensus disruption attack and Byzantines can be easily identified detected by techniques such as those given in [7], [28].

A. Distributed Algorithm for Weighted Average Consensus

In this section, we address the following questions: does there exist a distributed algorithm that solves the weighted average consensus problem while satisfying the condition that weights must be assigned or updated by neighbors $\mathcal{N}_i$ of the node i rather than by the node i itself? If it exists, then, under what conditions or constraints does the algorithm converge?

We consider a network with N nodes with a fixed and connected topology $G(V, E)$. Next, we state Perron-Frobenius theorem [30], which will be used later for the design and analysis of our robust weighted average consensus algorithm.

Theorem 1 ([30]). *Let W be a primitive nonnegative matrix with left and right eigenvectors u and v, respectively, satisfying $Wv = v$ and $u^T W = u^T$. Then, $\lim_{k \to \infty} W^k = \frac{vu^T}{v^T u}$.*

Using the above theorem, we take a reverse-engineering approach to design a modified Perron matrix $\hat{W}$ which has the weight vector $w = [w_1, w_2, \cdots, w_N]^T$, $w_i > 0$, $\forall i$ as its left eigenvector and $\vec{1}$ as its right eigenvector corresponding to eigenvalue 1. From the above theorem, if the modified Perron matrix $\hat{W}$ is primitive and nonnegative, then, a weighted average consensus can be achieved. Now, the problem boils down to designing such a $\hat{W}$ which meets our requirement that weights are assigned or updated by the neighbors $\mathcal{N}_i$ of node i rather than by node i itself.

Next, we propose a modified Perron matrix $\hat{W} = I - \epsilon(T \otimes L)$ where L is the original graph Laplacian, $\otimes$ is element-wise matrix multiplication operator, and T is a transformation given by

$$[T]_{ij} = \begin{cases} \dfrac{\sum\limits_{j \in \mathcal{N}_i} w_j}{l_{ii}} & \text{if } i = j \\ w_j & \text{otherwise.} \end{cases}$$

Observe that, the above transformation T satisfies the condition that weights are assigned or updated by neighbors $\mathcal{N}_i$ of node i rather than by node i itself. Based on the above transformation T, we propose our distributed consensus algorithm:

$$x_i(k+1) = x_i(k) + \epsilon \sum_{j \in \mathcal{N}_i} w_j(x_j(k) - x_i(k)).$$

Let us denote the modified Perron matrix by $\hat{W} = I - \epsilon\hat{L}$.

Next, we explore the properties of the modified Perron matrix $\hat{W}$ and show that it satisfies the requirements of the Perron-Frobenius theorem [30]. These properties will later be utilized to prove the convergence of our proposed consensus algorithm.

Lemma 3. *Let G be a connected graph with N nodes. Then, the modified Perron matrix $\hat{W} = I - \epsilon(T \otimes L)$, with $0 < \epsilon < \dfrac{1}{\max\limits_i \left(\sum\limits_{j \in \mathcal{N}_i} w_j \right)}$ satisfies the following properties.*

1) *$\hat{W}$ is a nonnegative matrix with left eigenvector w and right eigenvector $\vec{1}$ corresponding to eigenvalue 1;*
2) *All eigenvalues of $\hat{W}$ are in a unit circle;*
3) *$\hat{W}$ is a primitive matrix[2].*

Proof: Notice that, $\hat{W}\vec{1} = \vec{1} - \epsilon(T \otimes L)\vec{1} = \vec{1}$ and $w^T\hat{W} = w^T - \epsilon w^T(T \otimes L) = w^T$. This implies that $\hat{W}$ has left eigenvector w and right eigenvector $\vec{1}$ corresponding to eigenvalue 1. To show that $\hat{W} = I + \epsilon T \otimes A - \epsilon T \otimes D$ is non-negative, it is sufficient to show that: $w > 0$, $\epsilon > 0$ and $\epsilon(\max_i(\sum\limits_{j \in \mathcal{N}_i} w_j)) \leq 1, \forall i$. Since w is the left eigenvector of $\hat{L}$ and $w > 0$, $\hat{W}$ is non-negative if and only if

$$0 < \epsilon \leq \frac{1}{\max_i(\sum\limits_{j \in \mathcal{N}_i} w_j)}.$$

---

[2]A matrix is primitive if it is non-negative and its mth power is positive for some natural number m.

To prove part 2), notice that all the eigenvectors of $\hat{W}$ and $\hat{L}$ are the same. Let γ_j be the jth eigenvalue of $\hat{L}$, then, the jth eigenvalue of $\hat{W}$ is $\lambda_j = 1 - \epsilon\gamma_j$. Now, part 2) can be proved by applying Gershgorin theorem [30] to the modified Laplacian matrix $\hat{L}$.

To prove part 3), note that G is strongly connected and, therefore, $\hat{W}$ is an irreducible matrix [30]. Thus, to prove that $\hat{W}$ is a primitive matrix, it is sufficient[3] to show that $\hat{W}$ has a single eigenvalue with maximum modulus of 1. In [27], the authors showed that when $0 < \epsilon < \max_i(\sum_{j \neq i} a_{ij})$, the original Perron matrix W has only one eigenvalue with maximum modulus 1 at its spectral radius. Using a similar logic, $\hat{W}$ is a primitive matrix if

$$0 < \epsilon < \frac{1}{\max_i\left(\sum_{j \in \mathcal{N}_i} w_j\right)}.$$

■

Theorem 2. *Consider a network with fixed and strongly connected undirected topology $G(V, E)$ that employs the distributed consensus algorithm*

$$x_i(k+1) = x_i(k) + \epsilon \sum_{j \in \mathcal{N}_i} w_j(x_j(k) - x_i(k))$$

where

$$0 < \epsilon < \frac{1}{\max_i\left(\sum_{j \in \mathcal{N}_i} w_j\right)}.$$

Then, consensus with $x_i^* = \frac{\sum_{i=1}^N w_i x_i(0)}{\sum_{i=1}^n w_i}, \forall i$ is reached asymptotically.

Proof: A consensus is reached asymptotically, if the limit $\lim_{k \to \infty} \hat{W}^k$ exists. According to Perron-Frobenius theorem [30], this limit exists for primitive matrices. Note that, $\vec{1} = [1, \cdots, 1]^T$ and w are right and left eigenvectors of the primitive nonnegative matrix $\hat{W}$ respectively. Thus,

---

[3]An irreducible stochastic matrix is primitive if it has only one eigenvalue with maximum modulus.
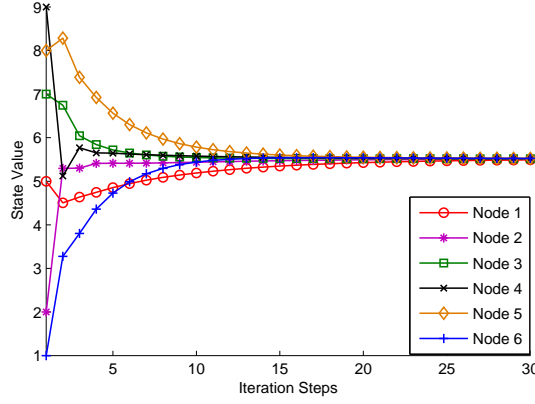
Figure 5.   Convergence of the network with a 6-nodes ($\epsilon = 0.3$).

from [30]

$$\lim_{k \to \infty} x(k) = \lim_{k \to \infty} (\hat{W})^k x(0)$$

$$x^* = \vec{1} \frac{\vec{w}^T x(0)}{w^T \vec{1}}$$

$$x^* = \vec{1} \frac{\sum_{i=1}^{N} w_i x_i(0)}{\sum_{i=1}^{n} w_i}$$

$\blacksquare$

Next, to gain insights into the convergence property of the proposed algorithm, we present some numerical results in Figure 5. We consider the 6-node network shown in Figure 1 where the nodes employ the proposed algorithm (with $\epsilon = 0.3$) to reach a consensus. Next, we plot the updated state values at each node as a function of consensus iterations. We assume that the initial data vector is $x(0) = [5, 2, 7, 9, 8, 1]^T$ and the weight vector is $w = [0.65, 0.55, 0.48, 0.95, 0.93, 0.90]^T$. Note that, the parameter ϵ satisfies the condition mentioned in Theorem 2. Figure 5 shows the convergence of the proposed algorithm iterations for a fixed communication graph. It is observed that within 20 iterations consensus has been reached on the global decision statistics or weighted average of the initial values (states).

In the proposed consensus algorithm, weights assigned to node i's data are updated by neighbors of the node i rather than by node i itself which addresses the first issue.

B. Adaptive Design of the Update Rules based on Learning of Nodes' Behavior

Next, to address the second issue, we exploit the statistical distribution of the sensing data and devise techniques to mitigate the influence of Byzantines on the distributed detection system. We propose a three-tier mitigation scheme where the following three steps are performed at each node: 1) identification of Byzantine neighbors, 2) estimation of parameters of identified Byzantine neighbors, and 3) adaptation of consensus algorithm (or update weights) using estimated parameters.

We first present the design of distributed optimal weights for the honest/Byzantine nodes assuming that the identities of the nodes are known. Later we will explain how the identity of nodes (i.e., honest/Byzantine) can be determined.

1) Design of Distributed Optimal Weights in the Presence of Byzantines: In this subsection, we derive closed form expressions for the distributed optimal weights which maximize the deflection coefficient. First, we consider the global test statistic $\Lambda = \frac{\sum_{i=1}^{N_1} w_i^B \tilde{Y}_i + \sum_{i=N_1+1}^{N} w_i^H Y_i}{\sum_{i=1}^{N_1} w_i^B + \sum_{i=N_1+1}^{N} w_i^H}$ and obtain a closed form solution for optimal centralized weights. Then, we extend our analysis to the distributed scenario. Let us denote by δ_i^B, the centralized weight given to the Byzantine node and by δ_i^H, the centralized weight given to the Honest node. By considering $\delta_i^B = w_i^B / \left(\sum_{i=1}^{N_1} w_i^B + \sum_{i=N_1+1}^{N} w_i^H \right)$ and $\delta_i^H = w_i^H / \left(\sum_{i=1}^{N_1} w_i^B + \sum_{i=N_1+1}^{N} w_i^H \right)$, the optimal weight design problem can be stated formally as:

$$\max_{\{\delta_i^B\}_{i=1}^{N_1}, \{\delta_i^H\}_{i=N_1+1}^{N}} \frac{(\mu_1 - \mu_0)^2}{\sigma_{(0)}^2}$$

$$\text{st.} \quad \sum_{i=1}^{N_1} \delta_i^B + \sum_{i=N_1+1}^{N} \delta_i^H = 1$$

where μ_1, μ_0 and $\sigma_{(0)}^2$ are given as in (3), (4) and (5), respectively. The solution of the above problem is presented in the next lemma.
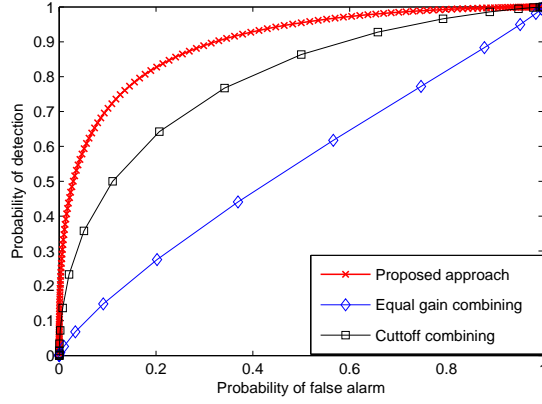
Figure 6. ROC for different protection approaches

Lemma 4. *Optimal centralized weights which maximize the deflection coefficient are given as*

$$\delta_i^B \;=\; \frac{w_i^B}{\displaystyle\sum_{i=1}^{N_1} w_i^B + \sum_{i=N_1+1}^{N} w_i^H},$$

$$\delta_i^H \;=\; \frac{w_i^H}{\displaystyle\sum_{i=1}^{N_1} w_i^B + \sum_{i=N_1+1}^{N} w_i^H}$$

where $w_i^B = \dfrac{(\eta_i \sigma_i^2 - 2P_i \Delta_i)}{\Delta_i^2 P_i(1 - P_i) + 2M\sigma_i^4}$ *and* $w_i^H = \dfrac{\eta_i}{2M\sigma_i^2}.$

Proof: The above results can be obtained by equating the derivative of the deflection coefficient to zero. ∎

Remark 2. *Distributed optimal weights can be chosen as* w_i^B *and* w_i^H. *Thus, the value of the global test statistic (or final weighted average consensus) is the same as the optimal centralized weighted combining*[4].

Next, to gain insights into the solution, we present some numerical results in Figure 6 that corroborate our theoretical results. We assume that $M = 12$, $\eta_i = 3$, $\sigma_i^2 = 0.5$ and the attack

---

[4]Note that, weights w_i^B can be negative and in that case convergence of the proposed algorithm is not guaranteed. However, this situation can be dealt off-line by adding a constant value to make $w_i^B \geq 0$ and changing the threshold λ accordingly. More specifically, by choosing a constant c such that $\left(w_i^B + \frac{c}{x_i(0)}\right) \geq 0, \forall i$ and $\lambda \leftarrow \lambda + \beta c$ where β is number of nodes with $w_i^B < 0$.

parameters are $(P_i, \Delta_i) = (0.5, 9)$. In Figure 6, we compare our proposed weighted average consensus based detection scheme with the equal gain combining scheme[5] and the scheme where Byzantines are cut off or removed from the fusion process.

It can be clearly seen from the figure that our proposed scheme performs better than the rest of the schemes.

Notice that, the optimal weights for the Byzantines are functions of the attack parameters $(P_i, \; \Delta_i)$, which may not be known to the neighboring nodes in practice. In addition, the parameters of the honest nodes might also not be known. Thus, we propose a technique to learn or estimate these parameters. We then use these estimates to adaptively design the local fusion rule which are updated after each learning iteration.

2) Identification, Estimation, and Adaptive Fusion Rule: The first step at each node m is to determine the identity ($I^i \in \{H, B\}$) of its neighboring nodes $i \in \mathcal{N}_m$. Notice that, if node i is an honest node, its data under hypothesis H_k is normally distributed $\mathcal{N}((\mu_{1k})_i, (\sigma_{1k})_i^2)$. On the other hand, if node i is a Byzantine node, its data under hypothesis H_k is a Gaussian mixture which comes from $\mathcal{N}((\mu_{1k})_i, (\sigma_{1k})_i^2)$ with probability $(\alpha_1^i = 1 - P_i)$ and from $\mathcal{N}((\mu_{2k})_i, (\sigma_{2k})_i^2)$ with probability $\alpha_2^i = P_i$. Therefore, determining the identity ($I^i \in \{H, B\}$) of neighboring nodes $i \in \mathcal{N}_m$ can be posed as a hypothesis testing problem:

$I_0 \, (I^i = H) :$ Y_i is generated from a Gaussian distribution under each hypothesis H_k;

$I_1 \, (I^i = B) :$ Y_i is generated from a Gaussian mixture distribution under each hypothesis H_k.

Node classification can then be achieved using the maximum likelihood decision rule:

$$f(Y_i|\, I_0) \quad \underset{B}{\overset{H}{\gtrless}} \quad f(Y_i|\, I_1) \tag{8}$$

where $f(\cdot|\, I_l)$ is the probability density function (PDF) under each hypothesis I_l. However, the parameters of the distribution are not known. Next, we propose a technique to learn these parameters. For an honest node i, the parameters to be estimated are $((\mu_{1k})_i, (\sigma_{1k})_i^2)$ and for Byzantines the unknown parameter set to be estimated is $\theta = \{\alpha_j^i, (\mu_{jk})_i, (\sigma_{jk})_i^2\}$, where $k = \{0, 1\}$, $j = \{1, 2\}$ and $i = 1, \cdots, N_m$, for N_m neighbor nodes. These parameters are estimated by observing the data over multiple learning iterations. In each iteration t, every node in the

---

[5]In equal gain combining scheme, all the nodes (including Byzantines) are assigned the same weight.

$$(\hat{\sigma}_{10})_i^2(t+1) = \frac{(\sum_{r=1}^{t} D_1(r))[(\hat{\sigma}_{10})_i^2(t) + ((\hat{\mu}_{10})_i(t+1) - (\hat{\mu}_{10})_i(t))^2] + \sum_{d=1}^{D_1(t+1)} [y_i^0(d) - (\hat{\mu}_{10})_i(t+1)]^2}{\sum_{r=1}^{t+1} D_1(r)} \quad (9)$$

$$(\hat{\sigma}_{11})_i^2(t+1) = \frac{\sum_{r=1}^{t} (D - D_1(r))[(\hat{\sigma}_{11})_i^2(t) + ((\hat{\mu}_{11})_i(t+1) - (\hat{\mu}_{11})_i(t))^2] + \sum_{d=1}^{D-D_1(t+1)} [y_i^1(d) - (\hat{\mu}_{11})_i(t+1)]^2}{\sum_{r=1}^{t+1} (D - D_1(r))}$$

$$(10)$$

network observes the data coming from their neighbors for D detection intervals to learn their respective parameters. It is assumed that each node has the knowledge of the true hypothesis for D detection intervals (or history) through a feedback mechanism.

First, we explain how the unknown parameter set for the distribution under the null hypothesis (I_0) can be estimated. Let us denote the data coming from an honest neighbor node i as $\mathbf{Y}_i(t) = [y_i^0(1), \cdots, y_i^0(D_1(t)), y_i^1(D_1(t)+1), \cdots, y_i^1(D)]$ where $D_1(t)$ denotes the number of times H_0 occurred in learning iteration t and y_i^k denotes the data of node i when the true hypothesis was H_k. To estimate the parameter set, $((\mu_{1k})_i, (\sigma_{1k})_i^2)$, of an honest neighboring node, one can employ a maximum likelihood based estimator (MLE). We use $((\hat{\mu}_{1k})_i(t), (\hat{\sigma}_{1k})_i^2(t))$ to denote the estimates at learning iteration t, where each learning iteration consists of D detection intervals. The ML estimate of $((\mu_{1k})_i, (\sigma_{1k})_i^2)$ can be written in a recursive form as following:

$$(\hat{\mu}_{10})_i(t+1) = \frac{\sum_{r=1}^{t} D_1(r)}{\sum_{r=1}^{t+1} D_1(r)} (\hat{\mu}_{10})_i(t) + \frac{1}{\sum_{r=1}^{t+1} D_1(r)} \sum_{d=1}^{D_1(t+1)} y_i^0(d) \quad (11)$$

$$(\hat{\mu}_{11})_i(t+1) = \frac{\sum_{r=1}^{t} (D - D_1(r))}{\sum_{r=1}^{t+1} (D - D_1(r))} (\hat{\mu}_{11})_i(t) + \frac{1}{\sum_{r=1}^{t+1} (D - D_1(r))} \sum_{d=D_1(t+1)}^{D} y_i^1(d) \quad (12)$$

where expressions for $(\hat{\sigma}_{10})_i^2$ and $(\hat{\sigma}_{11})_i^2$ are given in (9) and (10). Observe that, by writing these expressions in a recursive manner, we need to store only D data samples at any given learning iteration t, but effectively use all tD data samples to determine the estimates.

Next, we explain how the unknown parameter set for the distribution under the alternate hypothesis (I_1) can be estimated. Since the data is distributed as a Gaussian mixture, we employ the expectation-maximization (EM) algorithm to estimate the unknown parameter set for Byzantines. Let us denote the data coming from a Byzantine neighbor i as $\tilde{\mathbf{Y}}_i(t) = [\tilde{y}_i^0(1), \cdots, \tilde{y}_i^0(D_1(t)), \tilde{y}_i^1(D_1(t)+1), \cdots, \tilde{y}_i^1(D)]$ where $D_1(t)$ denotes the number of times H_0 occurred in learning iteration t and $\tilde{y}_i^k$ denotes the data of node i when the true hypothesis was H_k. Let us denote the hidden variable as z_j with $j = \{1, 2\}$ or $(Z = [z_1, z_2])$. Now, the joint conditional PDF of $\tilde{y}_i^k$ and z_j, given the parameter set, can be calculated to be

$$
\begin{aligned}
P(\tilde{y}_i^k(d), z_j | \theta) &= P(z_j | \tilde{y}_i^k(d), \theta) P(\tilde{y}_i^k(d) | (\mu_{jk})_i, (\sigma_{jk})_i^2) \\
&= \alpha_j^i P(\tilde{y}_i^k(d) | (\mu_{jk})_i, (\sigma_{jk})_i^2)
\end{aligned}
$$

In the expectation step of EM, we compute the expectation of the log-likelihood function with respect to the hidden variables z_j, given the measurements $\tilde{\mathbf{Y}}_i$, and the current estimate of the parameter set θ^l. This is given by

$$
\begin{aligned}
Q(\theta, \theta^l) &= E[\log P(\tilde{\mathbf{Y}}_i, Z | \theta) | \tilde{\mathbf{Y}}_i, \theta^l] \\
&= \sum_{j=1}^{2} \sum_{d=1}^{D_1(t)} \log[\alpha_j^i P(\tilde{y}_i^0(d) | (\mu_{j0})_i, (\sigma_{j0})_i^2) P(z_j | \tilde{y}_i^0(d), \theta^l)] \\
&+ \sum_{j=1}^{2} \sum_{d=D_1(t)+1}^{D} \log[\alpha_j^i P(\tilde{y}_i^1(d) | (\mu_{j1})_i, (\sigma_{j1})_i^2) P(z_j | \tilde{y}_i^1(d), \theta^l)]
\end{aligned}
$$

where

$$
P(z_j | \tilde{y}_i^k(d), \theta^l) = \frac{\alpha_j^i(l) P(\tilde{y}_i^k(d) | (\mu_{jk})_i(l), (\sigma_{jk})_i^2(l))}{\sum\limits_{n=1}^{2} \alpha_n^i(l) P(\tilde{y}_i^k(d) | (\mu_{nk})_i(l), (\sigma_{nk})_i^2(l))}. \tag{13}
$$

In the maximization step of EM algorithm, we maximize $Q(\theta, \theta^l)$ with respect to the parameter set θ so as to compute the next parameter set:

$$
\theta^{l+1} = \arg\max_{\theta} \ Q(\theta, \theta^l).
$$

First, we maximize $Q(\theta, \theta^l)$ subject to the constraint $\left(\sum_{j=1}^{2} \alpha_j^i = 1\right)$. We define the Lagrangian $\mathcal{L}$ as

$$\mathcal{L} = Q(\theta, \theta^l) + \lambda\{\sum_{j=1}^{2} \alpha_j^i - 1\}.$$

Now, we equate the derivative of $\mathcal{L}$ to zero:

$$\frac{d}{d\alpha_j^i}\mathcal{L} = \lambda + \frac{\sum_{d=1}^{D_1(t)} P(z_j|\tilde{y}_i^0(d), \theta^l)}{\alpha_j^i} + \frac{\sum_{d=D_1(t)+1}^{D} P(z_j|\tilde{y}_i^1(d), \theta^l)}{\alpha_j^i} = 0.$$

Multiplying both sides by α_j^i and summing over j gives $\lambda = -D$. Similarly, we equate the derivative of $Q(\theta, \theta^l)$ with respect to $(\mu_{jk})_i$ and $(\sigma_k)_i^2$ to zero. Now, an iterative algorithm for all the parameters is

$$\alpha_j^i(l+1) = \frac{1}{D}\left[\sum_{d=1}^{D_1(t)} P(z_j|\tilde{y}_i^0(d), \theta^l) + \sum_{d=D_1(t)+1}^{D} P(z_j|\tilde{y}_i^1(d), \theta^l)\right] \tag{14}$$

$$(\mu_{j0})_i(l+1) = \frac{\sum_{d=1}^{D_1(t)} P(z_j|\tilde{y}_i^0(d), \theta^l)\tilde{y}_i^0(d)}{\sum_{d=1}^{D_1(t)} P(z_j|\tilde{y}_i^0(d), \theta^l)} \tag{15}$$

$$(\mu_{j1})_i(l+1) = \frac{\sum_{d=D_1(t)+1}^{D} P(z_j|\tilde{y}_i^1(d), \theta^l)\tilde{y}_i^1(d)}{\sum_{d=D_1(t)+1}^{D} P(z_j|\tilde{y}_i^1(d), \theta^l)} \tag{16}$$

$$(\sigma_{j0})_i^2(l+1) = \frac{\sum_{j=1}^{2}\sum_{d=1}^{D_1(t)} P(z_j|\tilde{y}_i^0(d), \theta^l)(\tilde{y}_i^0(d) - (\mu_{j0})_i(l+1))^2}{\sum_{j=1}^{2}\sum_{d=1}^{D_1(t)} P(z_j|\tilde{y}_i^0(d), \theta^l)} \tag{17}$$

$$(\sigma_{j1})_i^2(l+1) = \frac{\sum_{j=1}^{2}\sum_{d=D_1(t)+1}^{D} P(z_j|\tilde{y}_i^1(d), \theta^l)(\tilde{y}_i^1(d) - (\mu_{j1})_i(l+1))^2}{\sum_{j=1}^{2}\sum_{d=D_1(t)+1}^{D} P(z_j|\tilde{y}_i^1(d), \theta^l)} \tag{18}$$

In the learning iteration t, let the estimates after the convergence of the above algorithm be denoted by $\hat{\theta}(t) = \{\hat{\alpha}_j^i(t), (\hat{\mu}_{jk})_i(t), (\hat{\sigma}_{jk})_i^2(t)\}$. These estimates are then used as the initial values
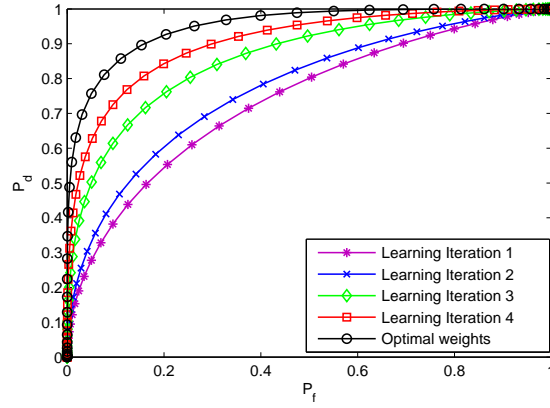
Figure 7. ROC for different learning iterations

for the next learning iteration $t + 1$ that uses a new set of D data samples.

After learning the unknown parameter set under I_0 and I_1, node classification can be achieved using the following maximum likelihood decision rule:

$$\hat{f}(\mathbf{Y}_i | I_0) \quad \underset{B}{\overset{H}{\gtrless}} \quad \hat{f}(\mathbf{Y}_i | I_1) \tag{19}$$

where $\hat{f}(\cdot)$ is the PDF based on estimated parameters.

Using the above estimates and node classification, the optimal distributed weights for honest nodes after learning iteration t can be written as

$$w_i^H(t) = \frac{(\hat{\mu}_{11})_i(t) - (\hat{\mu}_{10})_i(t)}{(\hat{\sigma}_{10})_i^2(t)}. \tag{20}$$

Similarly, the optimal distributed weights for Byzantines after learning iteration t can be written as

$$w_i^B(t) = \frac{\sum_{j=1}^{2} \hat{\alpha}_j^i(t)[(\mu_{j1})_i(t) - (\hat{\mu}_{j0})_i(t)]}{\hat{\alpha}_1^i(t)\hat{\alpha}_2^i(t)\left((\hat{\mu}_{10}(t))_i - (\hat{\mu}_{20}(t))_i\right)^2 + \left(\hat{\alpha}_1^i(t)(\hat{\sigma}_{10})_i^2(t) + \hat{\alpha}_2^i(t)(\hat{\sigma}_{20})_i^2(t)\right)} \tag{21}$$

Next, we present some numerical results in Figure 7 to evaluate the performance of our proposed scheme. Consider the scenario where 6 nodes organized in an undirected graph (as shown in Figure 1) are trying to detect a phenomenon. Node 1 and node 2 are considered

to be Byzantines. We assume that $((\mu_{10})_i, (\sigma_{10})_i^2) = (3, 1.5)$, $((\mu_{11})_i, (\sigma_{11})_i^2) = (4, 2)$ and the attack parameters are $(P_i, \Delta_i) = (0.5, 9)$. In Figure 7, we plot ROC curves for different learning iterations. For every learning iteration, we assume that $D_1 = 10$ and $D = 20$. It can be seen from Figure 7 that within 4 learning iterations, detection performance of the learning based weighted gain combining scheme approaches the detection performance of weighted gain combining with known optimal weight based scheme.

Note that, the above learning based scheme can be used in conjunction with the proposed weighted average consensus based algorithm to mitigate the effect of Byzantines.

VI. CONCLUSION AND FUTURE WORK

In this paper, we analyzed the security performance of conventional consensus based algorithms in the presence of data falsification attacks. We showed that above a certain fraction of Byzantine attackers in the network, existing consensus based detection algorithm are ineffective. Next, we proposed a robust distributed weighted average consensus algorithm and devised a learning technique to estimate the operating parameters (or weights) of the nodes. This enables an adaptive design of the local fusion or update rules to mitigate the effect of data falsification attacks. There are still many interesting questions that remain to be explored in the future work such as an analysis of the problem for time varying topologies. Note that, some analytical methodologies used in this paper are certainly exploitable for studying the attacks in time varying topologies. Other questions such as the optimal topology which incurs the fastest convergence rate can also be investigated.

REFERENCES

[1] P. K. Varshney, *Distributed Detection and Data Fusion*. New York:Springer-Verlag, 1997.

[2] R. Viswanathan and P. K. Varshney, "Distributed detection with multiple sensors: Part I - Fundamentals," *Proc. IEEE*, vol. 85, no. 1, pp. 54 –63, Jan 1997.

[3] V. Veeravalli and P. K. Varshney, "Distributed inference in wireless sensor networks," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 370, pp. 100–117, 2012.

[4] G. Xiong and S. Kishore, "Consensus-based distributed detection algorithm in wireless ad hoc networks," in *Signal Processing and Communication Systems, 2009. ICSPCS 2009. 3rd International Conference on*, Sept 2009, pp. 1–6.

[5] S. Aldosari and J. Moura, "Distributed Detection in Sensor Networks: Connectivity Graph and Small World Networks," in *Signals, Systems and Computers, 2005. Conference Record of the Thirty-Ninth Asilomar Conference on*, Oct 2005, pp. 230–234.

[6] S. Kar, S. Aldosari, and J. Moura, "Topology for Distributed Inference on Graphs," *Signal Processing, IEEE Transactions on*, vol. 56, no. 6, pp. 2609–2613, June 2008.

[7] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus Computation in Unreliable Networks: A System Theoretic Approach," *Automatic Control, IEEE Transactions on*, vol. 57, no. 1, pp. 90–104, Jan 2012.

[8] F. Yu, M. Huang, and H. Tang, "Biologically inspired consensus-based spectrum sensing in mobile Ad Hoc networks with cognitive radios," *Network, IEEE*, vol. 24, no. 3, pp. 26–30, May 2010.

[9] W. Zhang, Z. Wang, Y. Guo, H. Liu, Y. Chen, and J. Mitola, "Distributed Cooperative Spectrum Sensing Based on Weighted Average Consensus," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, Dec 2011, pp. 1–6.

[10] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982. [Online]. Available: http://doi.acm.org/10.1145/357172.357176

[11] A. Fragkiadakis, E. Tragos, and I. Askoxylakis, "A survey on security threats and detection techniques in cognitive radio networks," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 428–445, 2013.

[12] H. Rifà-Pous, M. J. Blasco, and C. Garrigues, "Review of robust cooperative spectrum sensing techniques for cognitive radio networks," *Wirel. Pers. Commun.*, vol. 67, no. 2, pp. 175–198, Nov. 2012. [Online]. Available: http://dx.doi.org/10.1007/s11277-011-0372-x

[13] S. Marano, V. Matta, and L. Tong, "Distributed detection in the presence of byzantine attacks," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 16 –29, Jan. 2009.

[14] A. Rawat, P. Anand, H. Chen, and P. Varshney, "Collaborative spectrum sensing in the presence of byzantine attacks in cognitive radio networks," *IEEE Trans. Signal Process.*, vol. 59, no. 2, pp. 774 –786, Feb 2011.

[15] B. Kailkhura, S. Brahma, Y. S. Han, and P. K. Varshney, "Distributed Detection in Tree Topologies With Byzantines," *IEEE Trans. Signal Process.*, vol. 62, pp. 3208–3219, June 2014.

[16] B. Kailkhura, S. Brahma, and P. K. Varshney, "Optimal byzantine attack on distributed detection in tree based topologies," in *Proc. International Conference on Computing, Networking and Communications Workshops (ICNC-2013)*, San Diego, CA, January 2013, pp. 227–231.

[17] B. Kailkhura, S. Brahma, Y. S. Han, and P. K. Varshney, "Optimal distributed detection in the presence of byzantines," in *Proc. The 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, Vancouver, Canada, May 2013.

[18] A. Vempaty, K. Agrawal, H. Chen, and P. K. Varshney, "Adaptive learning of Byzantines' behavior in cooperative spectrum sensing," in *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC)*, march 2011, pp. 1310 –1315.

[19] A. Min, K.-H. Kim, and K. Shin, "Robust cooperative sensing via state estimation in cognitive radio networks," in *New Frontiers in Dynamic Spectrum Access Networks (DySPAN), 2011 IEEE Symposium on*, May 2011, pp. 185–196.

[20] Z. Li, F. Yu, and M. Huang, "A Distributed Consensus-Based Cooperative Spectrum-Sensing Scheme in Cognitive Radios," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 1, pp. 383–393, Jan 2010.

[21] H. Tang, F. Yu, M. Huang, and Z. Li, "Distributed consensus-based security mechanisms in cognitive radio mobile ad hoc networks," *Communications, IET*, vol. 6, no. 8, pp. 974–983, May 2012.

[22] F. Yu, H. Tang, M. Huang, Z. Li, and P. Mason, "Defense against spectrum sensing data falsification attacks in mobile ad hoc networks with cognitive radios," in *Military Communications Conference, 2009. MILCOM 2009. IEEE*, Oct 2009, pp. 1–7.

[23] F. Yu, M. Huang, and H. Tang, "Biologically inspired consensus-based spectrum sensing in mobile Ad Hoc networks with cognitive radios," *Network, IEEE*, vol. 24, no. 3, pp. 26–30, May 2010.

[24] S. Liu, H. Zhu, S. Li, X. Li, C. Chen, and X. Guan, "An adaptive deviation-tolerant secure scheme for distributed cooperative spectrum sensing," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, Dec 2012, pp. 603–608.

[25] Q. Yan, M. Li, T. Jiang, W. Lou, and Y. Hou, "Vulnerability and protection for distributed consensus-based spectrum sensing in cognitive radio networks," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 900–908.

[26] F. Digham, M.-S. Alouini, and M. K. Simon, "On the energy detection of unknown signals over fading channels," in *Communications, 2003. ICC '03. IEEE International Conference on*, vol. 5, May 2003, pp. 3575–3579 vol.5.

[27] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan 2007.

[28] S. Sundaram and C. Hadjicostis, "Distributed Function Calculation via Linear Iterative Strategies in the Presence of Malicious Agents," *Automatic Control, IEEE Transactions on*, vol. 56, no. 7, pp. 1495–1508, July 2011.

[29] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*. ser. Prentice Hall Signal Processing Series, A. V. Oppenheim, Ed. Prentice Hall PTR, 1998.

[30] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1987.