Decentralized Hypothesis Testing in Wireless Sensor Networks in the Presence of Misbehaving Nodes

Erfan Soltanmohammadi, *Student Member, IEEE,* Mahdi Orooji, *Student Member, IEEE*, Mort Naraghi-Pour *Member, IEEE*

***Abstract*—Wireless sensor networks are prone to node misbehavior arising from tampering by an adversary (Byzantine attack), or due to other factors such as node failure resulting from hardware or software degradation. In this paper we consider the problem of decentralized detection in wireless sensor networks in the presence of one or more classes of misbehaving nodes. Binary hypothesis testing is considered where the honest nodes transmit their binary decisions to the fusion center (FC), while the misbehaving nodes transmit fictitious messages. The goal of the FC is to identify the misbehaving nodes and to detect the state of nature. We identify each class of nodes with an operating point (false alarm and detection probabilities) on the ROC (receiver operating characteristic) curve. Maximum likelihood estimation of the nodes' operating points is then formulated and solved using the expectation maximization (EM) algorithm with the nodes' identities as latent variables. The solution from the EM algorithm is then used to classify the nodes and to solve the decentralized hypothesis testing problem. Numerical results compared with those from the reputation-based schemes show a significant improvement in both classification of the nodes and hypothesis testing results. We also discuss an inherent ambiguity in the node classification problem which can be resolved if the honest nodes are in majority.**

***Index Terms*—Wireless sensor networks, decentralized hypothesis testing, expectation maximization, sensor node classification, Byzantine attack.**

I. Introduction

Wireless sensor networks (WSNs) consist of a large number of tiny battery-powered sensors that are densely deployed to sense their environment and report their findings to a central processor (fusion center) over wireless links. Due to size and energy constraints, sensor nodes have limited processing, storage and communication capabilities. In a large network of such sensors many nodes may fail due to hardware degradation or environmental effects. While in some cases a faulty node stops operating altogether, in other cases it may be misbehaving and reporting false data as in the case of stuck-at faults [1].

Sensor networks are also vulnerable to tampering. The networks are envisioned to be distributed over a large geographic area with unattended sensor nodes which may be captured and reprogrammed by an adversary. An adversary can also deploy its own sensor nodes to transmit false data in order to confuse the fusion center (FC). Sensors under an adversary's control are often referred to as Byzantine nodes.

In binary hypothesis testing, in order to lower their bandwidth requirement and energy expenditures, the sensors often make a local decision regarding the state of the hypothesis and only send their binary decision to the FC. Having received the messages from all the nodes, the FC will detect the hypothesis using a judicious decision rule [2].

The problem of decentralized detection in the presence of Byzantine nodes has been investigated by several authors [3]–[6]. In [4], it is assumed that through collaboration, the Byzantine nodes are aware of the true hypothesis. The authors formulate the problem in the context of Kullback-Leibler divergence and obtain optimal attacking distribution for the Byzantine nodes using a water-filling procedure. In [5], the authors consider data fusion schemes in a network under Byzantine attack and propose techniques for identifying the malicious users. In [6], the authors consider adding stochastic resonance noise at the honest and/or Byzantines in order to enhance the detection performance.

Cooperative spectrum sensing in cognitive radio networks (CRN) is another example of decentralized hypothesis testing where the secondary (unlicensed) users make a binary decision on whether a channel is vacant of the primary (licensed) user or not, and transmit that decision to the FC. The FC then processes the received data from all the secondary users and decides on the state of the channel. This problem is identical to the classical decentralized detection and recently several papers have considered cooperative spectrum sensing in the presence of Byzantine attacks (spectrum sensing data falsification) [7]–[13]. In [7], sequential probability ratio test is modified via a reputation-based mechanism in order to filter out the false data and only accept reliable messages. In [12], the authors present a scheme for identifying the Byzantine nodes and strategies for best fusion rule. In [14], a method is presented to detect the Byzantine nodes based on how their transmissions compare with those expected from honest nodes. These approaches are often categorized as reputation-based fusion rules [12], [15]. We note that in cooperative spectrum sensing we may also have more than one class of unreliable nodes. While some malicious users may send false data in order to gain unfair access to the channel, others may be sending incorrect data due to the malfunctioning of their sensing terminal. We should also point out that while a collaborative CRN may consist of at most tens of radios, a sensor network may comprise of hundreds or thousands of nodes. Therefore the proposed algorithms for CRNs may not

The authors are with the School of Electrical Engineering and Computer Science, Division of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA 70803 {e-mail: esolta1, morooj1, naraghi@lsu.edu}.

always be scalable for WSNs. However, the proposed method in this paper is also applicable in the case of cooperative spectrum sensing in CRNs.

In this paper we assume that there may be more than one class of misbehaving nodes. We show that from the point of view of the FC each class can be identified with a (operating) point on the receiver operating characteristic (ROC) that corresponds to the decision rule of the sensor nodes in that class. We first estimate the operating points of each class. For a fixed hypothesis vector, we formulate this problem as a maximum likelihood estimation problem with latent variables that correspond to the class identity of the nodes. This problem is then solved using the expectation maximization algorithm. Following this step we detect the class identity of each node and also detect the hypothesis vector.

The rest of this paper is organized as follows. The system model is presented in Section II. In Section III, the proposed node classifier is introduced, and in Section IV, the problem of counterpart networks for node classification is presented. Our performance metrics are introduced in Section V, and numerical results are provided and conclusions are drawn in Sections VI and VII, respectively.

II. System Model

We consider a wireless sensor network consisting of L nodes employed to detect the state of nature $H \in \{H_0, H_1\}$. It is assumed that there are K classes of nodes, $C = \{c_1, c_2, \cdots, c_K\}$, where c_1 is the class of honest nodes and $c_2, \cdots, c_K$ denote the other $K-1$ classes of (honest or misbehaving) nodes. Each node samples the environment once per unit time and makes a local decision on the state of H. It then transmits its binary decision to the FC which, after receiving a number of transmissions from the nodes, attempts to classify the nodes and also decide on the state of H.

Denote by $h_t \in \{H_0, H_1\}$ the state of H at time $t = 1, 2, \cdots, T$ and let $r_{l,t} \in \{0, 1\}, l = 1, 2, \cdots, L, t = 1, 2, \cdots, T$ denote the decision of the lth node at time t regarding the state of h_t. Since all the nodes in a class c_k are identical, the probabilities of detection and false alarm for class c_k are, respectively, given by

$$\tilde{p}_d(k) = P(r_{l,t} = 1 | h_t = H_1, l \in c_k), \tag{1}$$

and

$$\tilde{p}_f(k) = P(r_{l,t} = 1 | h_t = H_0, l \in c_k). \tag{2}$$

As in [4], [5], [12]–[14] we assume that the Byzantine nodes do not collaborate. While collaboration can improve the effectiveness of the adversary's attack, it has its own drawbacks. Collaboration requires additional infrastructure such as a FC to coordinate the attacks, as well as increased communication which can quickly deplete the resources of the Byzantine nodes. In the absence of such collaboration, we can assume that, given the hypothesis (H_0 or H_1), for any time t the sensor decisions $r_{l,t}, l = 1, 2, \ldots, L$ are conditionally independent [15]–[17]. In addition, we assume that the sensor decisions across time are conditionally independent given the hypothesis vector $\mathbf{h} = (h_1, h_2, \ldots, h_T)$, [12], [14] [1]. From these assumptions it follows that given the hypothesis vector $\mathbf{h}$, the sensor decisions $r_{l,t}, l = 1, 2, \ldots, L, t = 1, 2, \ldots, T$ are conditionally independent.

While an honest node $l \in c_1$ will transmit its decision $r_{l,t}$ to the FC, nodes in other classes may choose to do differently. In particular, let $d_{l,t} \in \{0, 1\}$ denote the message received at the FC from node l at time t and define

$$\rho_0(k) \triangleq P(d_{l,t} = 1 | r_{l,t} = 0, l \in c_k), \tag{3}$$

$$\rho_1(k) \triangleq P(d_{l,t} = 1 | r_{l,t} = 1, l \in c_k). \tag{4}$$

Clearly for honest nodes, $\rho_0(1) = 0$ and $\rho_1(1) = 1$. Let

$$\begin{aligned} p_d(k) &\triangleq P(d_{l,t} = 1 | h_t = H_1, l \in c_k) \\ &= \rho_1(k)\tilde{p}_d(k) + \rho_0(k)(1 - \tilde{p}_d(k)), \end{aligned} \tag{5}$$

and

$$\begin{aligned} p_f(k) &\triangleq P(d_{l,t} = 1 | h_t = H_0, l \in c_k) \\ &= \rho_1(k)\tilde{p}_f(k) + \rho_0(k)(1 - \tilde{p}_f(k)). \end{aligned} \tag{6}$$

One may view $p_d(k)$ and $p_f(k)$ as the detection and false alarm probabilities "perceived" by the FC for nodes in class c_k.

Recently in [13], the authors consider the problem of detecting statistical attacks in cognitive radios using belief propagation. This approach is similar to the reputation-based method of [12], [15]. The modeling assumptions in [13] are similar but somewhat simpler than those presented here. In particular two types of attackers are assumed. If node k is of Type-1, then it attempts to confuse the FC only when hypothesis H_1 is detected by sending a 0 with probability r_k and a 1 with probability $1 - r_k$. On the other hand, if node k is of Type-2, it tries to confuse the FC when the detected hypothesis is H_0 by sending a 1 with probability r_k and a 0 with probability $1 - r_k$. Note that $r_k = 0$ corresponds to honest nodes. It is also assumed that there is a subset of trusted nodes whose identities are known to the FC. In contrast, we do not assume that such prior information is available at the FC and our attacker model is more general in that the malicious nodes may try to confuse the FC under both hypotheses.

Remark 1. *In Section III, we present our method for estimating $(p_f(k), p_d(k))$ for $k = 1, 2, \cdots, K$. Our approach does not depend on how these probabilities are arrived at. In particular it includes the case that Byzantines, after detecting the hypothesis, flip their decisions and send it to the FC. This corresponds to $p_d(k) = 1 - \tilde{p}_d(k)$ and $p_f(k) = 1 - \tilde{p}_f(k)$. Furthermore, we have assumed error free channels between the sensors and the FC. However, the model presented here also includes noisy channel models between sensors and the FC. The effect of the channel transition probabilities can be included in the parameters $\rho_0(k)$ and $\rho_1(k)$.*

The receiver operating characteristic (ROC) of a node in class c_k is denoted by U_k, i.e., $\tilde{p}_d(k) = U_k(\tilde{p}_f(k))$. In the following we refer to the point $(p_f(k), p_d(k))$ as the

[1] This assumption holds for example when the sensors' observations across time are contaminated by white noise.

operating point of a node in class c_k. For the honest nodes, $p_d(k) = \tilde{p}_d(k)$ and $p_f(k) = \tilde{p}_f(k)$, and so their operating point is $(\tilde{p}_f(k), U_k(\tilde{p}_f(k))$. We show in Appendix A that for other nodes, the operating point is in a region bounded by $(\tilde{p}_f(k), U_k(\tilde{p}_f(k))$ and $(\tilde{p}_f(k), V_k(\tilde{p}_f(k))$, where $V_k(x)$ is the reflection of $U_k(x)$ with respect to the point $(0.5, 0.5)$, i.e., $V_k(x) = 1-U_k(1-x)$. These nodes can achieve any operating point in this region by choosing appropriate values for $\rho_0(k)$ and $\rho_1(k)$.

III. Classification of the Nodes

Let $\mathbf{Z} = [z_{l,k}],\ z_{l,k} \in \{0,1\}$ for $l = 1, 2, \cdots, L,\ k = 1, 2, \cdots, K$ denote the identification matrix of the nodes where $z_{l,k} = 1$ if $l \in c_k$ and 0, otherwise. To identify the nodes, the FC collects T messages from each node and stores them in a matrix $\mathbf{D} = [d_{l,t}],\ l = 1, 2, \cdots, L,\ t = 1, 2, \cdots, T$ subsequently referred to as the decision matrix. Using the decision matrix the FC must detect the identification matrix $\mathbf{Z}$ and the hypothesis vector $\mathbf{h} = (h_1, h_2, \cdots, h_T)$.

The maximum likelihood detection rule for $(\mathbf{Z}, \mathbf{h})$ is given by

$$(\hat{\mathbf{Z}}, \hat{\mathbf{h}}) = \arg\max_{\mathbf{Z},\mathbf{h}} P(\mathbf{D}|\mathbf{Z}, \mathbf{h}). \tag{7}$$

Evaluation of (7) requires the likelihood function $P(\mathbf{D}|\mathbf{Z}, \mathbf{h})$ which is computed below. For a given hypothesis vector $\mathbf{h}$, denote the number of H_0's and H_1's in $\mathbf{h}$ by N and $M = T - N$, respectively. Also denote the number of correct decisions of the lth node on hypotheses H_0 and H_1 by n_l and $m_l,\ 0 \leq l \leq L$, respectively. In other words, out of N occurrences of H_0 in $\mathbf{h}$, node l correctly detects n_l of them, and out of M occurrences of H_1 in $\mathbf{h}$, it correctly detects m_l of them. We note that for a given hypothesis vector $\mathbf{h}$, n_l and m_l can be calculated from the lth row of $\mathbf{D}$. We have,

$$P(\mathbf{D}|\mathbf{Z}, \mathbf{h}) = \prod_{l=1}^{L}\prod_{k=1}^{K}\Big(p_c(k)^{n_l}(1-p_c(k))^{(N-n_l)} \\ p_d(k)^{m_l}(1-p_d(k))^{(M-m_l)}\Big)^{z_{l,k}} \tag{8}$$

where $p_c(k) \triangleq 1 - p_f(k)$ is the probability of correct rejection.

It can be seen from (8) that the likelihood function $P(\mathbf{D}|\mathbf{Z}, \mathbf{h})$ depends on the unknown parameters $(p_f(k), p_d(k))$ for $k = 1, 2, \cdots, K$. Therefore for the detection problem in (7) the Bayesian or the Neyman-Pearson rule cannot be implemented. Generalized likelihood ratio test (GLRT) is often used in detection problems with unknown parameters [18]. However, for our problem GLRT is not mathematically tractable. Therefore, in this paper, we follow the following process. For a given hypothesis vector $\mathbf{h}$ we first estimate the operating points $(p_f(k), p_d(k))$ for $k = 1, 2, \cdots, K$. Using the estimated operating points, we can implement the maximum a posteriori (MAP) classification rule for $\mathbf{Z}$. The estimated operating points and identification matrix $\mathbf{Z}$ are then used to implement the maximum likelihood detection rule for the hypothesis vector $\mathbf{h}$. We have not been able to prove the optimality of the proposed method due to its mathematical intractability. In section VI our simulation results are compared with the Cramer-Rao lower bound and show a close match.

A. Estimation of Class Parameters

From (8), it is evident that to detect $\mathbf{Z}$ we need to first estimate the operating points $(p_f(k), p_d(k))$ for $k = 1, 2, \cdots, K$. Note that in the following it is assumed that the hypothesis vector $\mathbf{h}$ is fixed and all the probabilities are conditioned on $\mathbf{h}$. For ease of notation, however, we drop this condition from our notations.

In addition to the operating points of each class, the FC is also unaware of the fraction of nodes in each class. Let $\pi_k = P(z_{k,l} = 1)$ denote the probability that node l belong to class c_k and define the matrix of class parameters, $\mathbf{\Theta}$, where its kth row is given by

$$\boldsymbol{\theta}(k) \triangleq [p_c(k), p_d(k), \pi(k)]. \tag{9}$$

We would like to estimate the class parameters $\mathbf{\Theta}$ from the observation matrix $\mathbf{D}$. Since the conditional probability $P(\mathbf{D}|\mathbf{\Theta})$ is not given, we may write the maximum likelihood estimate for $\mathbf{\Theta}$ as,

$$\mathbf{\Theta}^* = \arg\max_{\mathbf{\Theta}} \sum_{\mathbf{Z}} P(\mathbf{D}, \mathbf{Z}|\mathbf{\Theta}). \tag{10}$$

This may be viewed as a mixture model (with $\mathbf{Z}$ as the latent variables since the nodes are not identified) and can be effectively solved using the iterative Expectation Maximization (EM) algorithm [19]. Let us define the log-likelihood function,

$$L(\mathbf{\Theta}; \mathbf{D}, \mathbf{Z}) \triangleq \log P(\mathbf{D}, \mathbf{Z}|\mathbf{\Theta}) \tag{11}$$

Due to the fact that $\mathbf{Z}$ is latent, with EM we consider the conditional expectation of (11) under the posterior distribution of $\mathbf{Z}$ given $\mathbf{D}$ and $\mathbf{\Theta}$. This is the expectation step of EM. In the maximization step, this expectation is maximized with respect to $\mathbf{\Theta}$. Denote the current and the revised estimate of $\mathbf{\Theta}$ by $\mathbf{\Theta}^{\text{old}}$ and $\mathbf{\Theta}^{\text{new}}$, respectively. The two steps of EM algorithm are described below.

1) Expectation: Using the current estimate of the matrix of class parameters, $\mathbf{\Theta}^{\text{old}}$, find the posterior distribution of $\mathbf{Z}$ given $\mathbf{D}$ and $\mathbf{\Theta}^{\text{old}}$. Using this distribution find the expectation of the log likelihood function in (11) for an arbitrary $\mathbf{\Theta}$ given by

$$Q(\mathbf{\Theta}; \mathbf{\Theta}^{\text{old}}) \triangleq E_{\mathbf{Z}}[L(\mathbf{\Theta}; \mathbf{D}, \mathbf{Z})|\mathbf{D}, \mathbf{\Theta}^{\text{old}}] \\ = \sum_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{D}, \mathbf{\Theta}^{\text{old}}) \times L(\mathbf{\Theta}; \mathbf{D}, \mathbf{Z}). \tag{12}$$

2) Maximization: Revise the estimate of class parameters to maximize the expectation calculated in the previous step, i.e., let

$$\mathbf{\Theta}^{\text{new}} = \arg\max_{\mathbf{\Theta}} Q(\mathbf{\Theta}; \mathbf{\Theta}^{\text{old}}). \tag{13}$$

It has been shown that each update of the EM algorithm is guaranteed to increase the log-likelihood function [20]. This implies that the EM algorithm will converge regardless of the

initial value of $\mathbf{\Theta}$, [19], [21]. We now present the two steps of EM algorithm for the problem at hand.

$$\begin{aligned} L(\mathbf{\Theta};\mathbf{D},\mathbf{Z}) &= \log P(\mathbf{D},\mathbf{Z}|\mathbf{\Theta}) \\ &= \log[P(\mathbf{D}|\mathbf{Z},\mathbf{\Theta})P(\mathbf{Z}|\mathbf{\Theta})] \\ &= \log \prod_{l=1}^{L}\prod_{k=1}^{K} \pi_k^{z_{k,l}} \Big[p_c(k)^{n_l}(1-p_c(k))^{(N-n_l)} \\ &\qquad p_d(k)^{m_l}(1-p_d(k))^{(M-m_l)}\Big]^{z_{k,l}} \\ &= \sum_{l=1}^{L}\sum_{k=1}^{K} z_{k,l}\,[\log\pi_k + n_l \log p_c(k) \\ &+ (N-n_l)\log(1-p_c(k)) + m_l \log p_d(k) \\ &+ (M-m_l)\log(1-p_d(k))\,]. \end{aligned} \tag{14}$$

To calculate $Q(\mathbf{\Theta};\mathbf{\Theta}^{\text{old}})$ in (12) for the expectation step, one should find the conditional expectation of $L(\mathbf{\Theta};\mathbf{D},\mathbf{Z})$ with respect to $\mathbf{Z}$. Hence,

$$\begin{aligned} &Q(\mathbf{\Theta};\mathbf{\Theta}^{\text{old}}) = \\ &\sum_{l=1}^{L}\sum_{k=1}^{K} E[z_{k,l}|\mathbf{D},\mathbf{\Theta}^{\text{old}}]\,[\log\pi_k + n_l\log p_c(k) \\ &+ (N-n_l)\log(1-p_c(k)) + m_l\log p_d(k) \\ &+ (M-m_l)\log(1-p_d(k))\,]. \end{aligned} \tag{15}$$

We now need to perform the maximization step in (15). Denoting $x_l \triangleq (n_l, m_l),\ 1 \le l \le L$, we have

$$\begin{aligned} \mathfrak{E}(l,k) &\triangleq E(z_{k,l}|\mathbf{D},\mathbf{\Theta}^{\text{old}}) = P(z_{l,k}=1|x_l;\mathbf{\Theta}^{\text{old}}) \\ &= \frac{\pi_k^{(\text{old})} P(x_l|z_{l,k}=1;\boldsymbol{\theta}^{(\text{old})}(k))}{\sum_{j=1}^{K}\pi_j^{(\text{old})}P(x_l|z_{l,j}=1;\boldsymbol{\theta}^{(\text{old})}(j))}, \end{aligned} \tag{16}$$

where,

$$\begin{aligned} &P(x_l|z_{l,k}=1;\boldsymbol{\theta}^{(\text{old})}(k)) \\ &\quad = [p_c^{(\text{old})}(k)]^{n_l}\ [1-p_c^{(\text{old})}(k)]^{(N-n_l)} \\ &\qquad \times [p_d^{(\text{old})}(k)]^{m_l}\ [1-p_d^{(\text{old})}(k)]^{(M-m_l)}, \end{aligned} \tag{17}$$

and where $\boldsymbol{\theta}^{(\text{old})}(k)$ (the kth row of $\mathbf{\Theta}^{\text{old}}$) is the current vector of parameters for the kth class. The quantity $\mathfrak{E}(l,k)$ can be interpreted as the probability that class c_k is responsible for the decisions made by the lth node. So, the effective number of nodes assigned to class c_k, denoted by L_k, is given by,

$$L_k \triangleq \sum_{l=1}^{L}\mathfrak{E}(l,k). \tag{18}$$

The estimation of the probability of correct rejection and the probability of detection for any $1 \le k \le K$ can be found by solving (13) as,

$$\frac{\partial Q(\mathbf{\Theta};\mathbf{\Theta}^{\text{old}})}{\partial p_c(k)} = \sum_{l=1}^{L}\mathfrak{E}(l,k)\left[\frac{n_l}{p_c(k)} - \frac{N-n_l}{1-p_c(k)}\right] = 0, \tag{19}$$

$$\frac{\partial Q(\mathbf{\Theta};\mathbf{\Theta}^{\text{old}})}{\partial p_d(k)} = \sum_{l=1}^{L}\mathfrak{E}(l,k)\left[\frac{m_l}{p_d(k)} - \frac{M-m_l}{1-p_d(k)}\right] = 0, \tag{20}$$

which after some manipulations results in,

$$p_c^{\text{new}}(k) = \frac{1}{L_k}\sum_{l=1}^{L}\frac{n_l}{N}\ \mathfrak{E}(k,l), \tag{21}$$

$$p_d^{\text{new}}(k) = \frac{1}{L_k}\sum_{l=1}^{L}\frac{m_l}{M}\ \mathfrak{E}(k,l). \tag{22}$$

Finally, we should maximize $Q(\mathbf{\Theta};\mathbf{\Theta}^{\text{old}})$ with respect to π_k with the constraint that $\sum_{k=1}^{K}\pi_k = 1$. This can be achieved using Lagrange multiplier method by maximizing the Lagrangian

$$\tilde{Q}(\mathbf{\Theta},\nu;\mathbf{\Theta}^{\text{old}}) \triangleq Q(\mathbf{\Theta};\mathbf{\Theta}^{\text{old}}) + \nu[\sum_{k=1}^{K}\pi_k - 1]. \tag{23}$$

We have

$$\frac{\partial\tilde{Q}}{\partial\pi_k} = \sum_{l=1}^{L}\frac{\mathfrak{E}(l,k)}{\pi_k} + \nu = 0 \tag{24}$$

Multiplying both sides by π_k and summing over k we get $\nu = -L$ which results in

$$\pi_k^{\text{new}} = \frac{L_k}{L} \tag{25}$$

Since the $\log(.)$ function is concave and $\mathfrak{E}(l,k) \ge 0$, $\forall l,k$, it can be seen from (15) that $Q(\mathbf{\Theta};\mathbf{\Theta}^{\text{old}})$ is a concave function of π_k's (in $\Re^+$). This followed by the fact that the constraint $\sum_{k=1}^{K}\pi_k = 1$ is linear in π_k's implies that the Lagrange multiplier method in (24) achieves the optimal solution [22].

B. Classification of the Nodes

Let $\mathbf{\Theta}^*$ denote the matrix of class parameters estimated by the EM algorithm. Given $\mathbf{\Theta}^*$, the conditional probability that node l belongs to class c_k is given by

$$\begin{aligned} &P(z_{l,k}=1|x_l;\boldsymbol{\theta}^*(k)) \\ &\quad = \frac{\pi_k^* P(x_l|z_{l,k}=1;\boldsymbol{\theta}^*(k))}{\sum_{j=1}^{K}\pi_j^* P(x_l|z_{l,j}=1;\boldsymbol{\theta}^*(j))}, \end{aligned} \tag{26}$$

where $\boldsymbol{\theta}^*(k)$ is the kth row of $\mathbf{\Theta}^*$. The denominator in (26) is independent of k. Therefore, the maximum a posteriori classification rule for node l (given $\mathbf{\Theta}^*$) is given by

$$k^* = \arg\max_k\ \{\pi_k^* P(x_l|z_{l,k}=1;\boldsymbol{\theta}^*(k)),\ k=1,2,\cdots,K\}, \tag{27}$$

and we set

$$z_{l,k}^* = \begin{cases} 1 & \text{for } k = k^* \\ 0 & \text{for } k \ne k^*. \end{cases} \tag{28}$$

C. Estimation of the Hypothesis Vector

In the previous section we showed how to estimate the class parameters $\mathbf{\Theta}^*$ and obtain the node identification matrix $\mathbf{Z}^*$ for a given hypothesis vector $\mathbf{h}$. Therefore, in the sequel we denote these parameters by $\mathbf{Z}^*(\mathbf{h}) = [z_{l,k}^*(\mathbf{h})]$ and $\mathbf{\Theta}^*(\mathbf{h})$. Similarly N, M, n_l, and m_l are substituted by $N(\mathbf{h})$, $M(\mathbf{h})$, $n_l(\mathbf{h})$, and $m_l(\mathbf{h})$, respectively. The maximum likelihood detection rule

for $\mathbf{h}$ obtained from the observation matrix $\mathbf{D}$ given $\mathbf{Z}^*(\mathbf{h})$ and $\mathbf{\Theta}^*(\mathbf{h})$ is now given by

$$\hat{\mathbf{h}} = \arg\max_{\mathbf{h}} P(\mathbf{D}|\mathbf{Z}^*(\mathbf{h}); \mathbf{\Theta}^*(\mathbf{h})) \tag{29}$$

where,

$$P(\mathbf{D}|\mathbf{Z}^*(\mathbf{h}); \mathbf{\Theta}^*(\mathbf{h})) = \tag{30}$$
$$\prod_{l=1}^{L}\prod_{k=1}^{K}\Big(p_c^*(k;\mathbf{h})^{n_l(\mathbf{h})}(1-p_c^*(k;\mathbf{h}))^{(N(\mathbf{h})-n_l(\mathbf{h}))}$$
$$p_d^*(k;\mathbf{h})^{m_l(\mathbf{h})}(1-p_d^*(k;\mathbf{h}))^{(M(\mathbf{h})-m_l(\mathbf{h}))}\Big)^{z_{l,k}(\mathbf{h})},$$

and where $[p_c^*(k;\mathbf{h}), p_d^*(k;\mathbf{h}), \pi^*(k;\mathbf{h})]$ is the kth row of $\mathbf{\Theta}^*(\mathbf{h})$ denoting the estimated parameters of the kth class for the hypothesis vector $\mathbf{h}$. The final estimation of all the network parameters is given by $(\hat{\mathbf{h}}, \hat{\mathbf{Z}}, \hat{\mathbf{\Theta}})$ where $\hat{\mathbf{Z}} = \mathbf{Z}^*(\hat{\mathbf{h}})$ and $\hat{\mathbf{\Theta}} = \mathbf{\Theta}^*(\hat{\mathbf{h}})$. The entire procedure is summarized in Algorithm 1.

Data: Decision matrix, $\mathbf{D}$
Result: Estimation of identification matrix, $\hat{\mathbf{Z}}$, the matrix of class parameters $\hat{\mathbf{\Theta}}$, and hypothesis vector $\hat{\mathbf{h}}$

begin
  forall the *possible hypothesis vectors,* $\mathbf{h} \in \{0,1\}^T$, **do**
    Estimate the matrix of class parameters, $\mathbf{\Theta}^*(\mathbf{h})$, *using EM-Algorithm:*
    Assume an initial value for $\mathbf{\Theta}^{\text{old}}$;
    while $\left\|\mathbf{\Theta}^{new} - \mathbf{\Theta}^{old}\right\| \geq \epsilon$ **do**
      E Step: Find $\mathfrak{E}(l,k)$ using (16);
      M Step: Estimate $\mathbf{\Theta}^{\text{new}}$ ($p_c^{\text{new}}(k)$, $p_d^{\text{new}}(k)$ and π_k^{new}) using (21), (22), and (25);
    end
    Classify the nodes by computing $\mathbf{Z}^*(\mathbf{h})$: for each node l find k^* using (27);
  end
  Detect the hypothesis vector, $\hat{\mathbf{h}}$ from (29);
  Find the $(\hat{\mathbf{h}}, \hat{\mathbf{Z}}, \hat{\mathbf{\Theta}})$ where $\hat{\mathbf{Z}} = \mathbf{Z}^*(\hat{\mathbf{h}})$ and $\hat{\mathbf{\Theta}} = \mathbf{\Theta}^*(\hat{\mathbf{h}})$.
end

Algorithm 1: Calculation of the identification matrix, the matrix of class parameters, and the hypothesis vector via the EM algorithm.

Remark 2. *We have assumed that the FC is aware of the number of classes K. The issue of how to select the number of classes known as model order selection is a well known problem in classification. While criteria such as Akaike information criterion (AIC) or Bayesian information criterion (BIC) have been proposed, they do not always work satisfactorily and tend to favor overly simple models [21]. The main issue in model selection is under- or overfitting the data. However, in large sensor networks this will not be an issue owing to the fact that the expected number of classes K is much smaller than the number of sensors L. Therefore K may be overestimated and yet be much smaller than L (in which case overfitting will not occur). If the actual number of classes is smaller, the proposed algorithm will not assign any nodes to the fictitious classes.*

In decentralized detection schemes such as ours, it is assumed that the nodes only transmit a (binary) quantized version of their measurement to the FC (instead of their actual measurement). A question then arises as to how the FC can identify the nodes. While the nodes can transmit a label for identification, the overhead associated with this approach may not be justified given the severely limited energy and transmission capability of the sensors. We believe that this issue can be resolved using the media access control mechanism. Clearly the sensors need some form of arbitration mechanism to access the channel. The information from that mechanism can be used by the FC to identify the nodes and determine which received bit corresponds to which node. For example the FC may use round-robin scheduling to collect the nodes' messages. The information from the nodes' turn in the schedule can be used to identify them.

D. Complexity

For a given hypothesis vector, the EM algorithm is very fast and converges in only a few steps. However, for a vector of T decisions from the sensors the EM algorithm must be performed 2^T times corresponding to the 2^T possible hypothesis vectors. This increases the complexity of the algorithm exponentially in terms of the observation interval. However, as discussed in the numerical section, the proposed algorithm converges much faster than the reputation-based algorithms in terms of the number of observation samples T (A brief description of the reputation-based algorithms is provided in Appendix B). Another point to observe is that the rate at which the state of nature changes is much lower than the rate at which the sensors sample the environment. In other words, during an observation time of T decisions from the sensors, the state of nature will not change more than a few times. In such a case the number of vectors $\mathbf{h}$ for which the EM algorithm is performed is only polynomial in T. For example, in order to detect a single change in $\mathbf{h}$ (from H_0 to H_1 or vice versa), EM is performed for only $2T$ possible vectors $\mathbf{h}$. Furthermore, the complexity of the proposed algorithm is linear in the number of nodes L and quadratic in the number of classes K. Given that sensor networks are expected to consist of hundreds or thousands of nodes, the linear complexity in the number of nodes is significant.

IV. Counterpart Networks

In this section we will show that any decision matrix $\mathbf{D}$ is equally likely to be generated by one of two different networks which we refer to as counterpart networks. For any matrix of class parameters $\mathbf{\Theta}$ we can define a counterpart matrix, $\mathbf{\Theta}^{(c)}$, whose kth row, $1 \leq k \leq K$, is given by

$$\begin{aligned}\boldsymbol{\theta}^{(c)}(k) &= [p_c^{(c)}(k), p_d^{(c)}(k), \pi^{(c)}(k)] \\ &= [1-p_d(k), 1-p_c(k), \pi(k)]\end{aligned} \tag{31}$$

Also define the counterpart hypothesis vector, $\mathbf{h}^{(c)} \triangleq \mathbf{1}_T - \mathbf{h}$ where $\mathbf{1}_T$ is a vector of all ones with length T. It can be verified that,

$$P(\mathbf{D}|\mathbf{Z}, \mathbf{\Theta}, \mathbf{h}) = P(\mathbf{D}|\mathbf{Z}, \mathbf{\Theta}^{(c)}, \mathbf{h}^{(c)}) \tag{32}$$

The intuition behind (32) is that the probability of transmitting a one (or a zero) for a node with the operating point (p_f, p_d) under $H_\eta,\ \eta \in \{0, 1\}$, is the same as a node with the operating point (p_d, p_f) under $H_{1-\eta}$. Therefore any observed decision matrix $\mathbf{D}$ is equally likely to be generated by one of two networks, namely $\{\mathbf{Z}, \mathbf{\Theta}\}$ under the hypothesis vector $\mathbf{h}$, or $\{\mathbf{Z}, \mathbf{\Theta}^{(c)}\}$ under the hypothesis vectors $\mathbf{h}^{(c)}$. This implies that regardless of the method used, there are always two solutions for the estimation of the class parameters and the detected hypothesis vector.

The ambiguity described above can be resolved by assuming some prior information on the network. In practice, the operating point of the honest nodes $(p_f(1), p_d(1))$ will be above the chance line $p_d = p_f$ [23]. If it is known that the class of honest nodes is the largest class, then the ambiguity can be resolved by choosing the solution for which the largest class is above the chance line.

V. Performance Assessment Metrics

To assess the performance of classifiers, two metrics of *discriminability* and *reliability* are often used [24]. Discriminability shows how well the classifier distinguishes the different classes, whereas reliability indicates how well the posterior probability that a node belongs to a class is estimated by the proposed method. To show the discriminability of the classifier, we define the misclassification rate by, [20],

$$\Delta_Z \triangleq \frac{1}{2L} \sum_{l=1}^{L} \sum_{k=1}^{K} |z_{l,k} - \hat{z}_{l,k}|. \tag{33}$$

Similarly the performance of our hypothesis detection scheme is evaluated by the *hypothesis discriminability* given by

$$\Delta_H \triangleq \frac{1}{T} \sum_{t=1}^{T} |h_t - \hat{h}_t|. \tag{34}$$

To estimate the accuracy of the estimation of the nodes' operating points we define the following measure based on the normalized Euclidean distance between the estimated and actual operating points, i.e.,

$$\Delta_{OP} \triangleq \frac{1}{\sqrt{2}} \sum_{k=1}^{K} \pi_k \sqrt{(p_d(k) - \hat{p}_d(k))^2 + (p_f(k) - \hat{p}_f(k))^2}. \tag{35}$$

Note that the three measure in (33)-(35) are appropriately normalized so as to be in the interval $[0, 1]$.

A. The Cramer-Rao Bound

To evaluate the efficacy of the expectation maximization algorithm in estimating the class parameters we would like to compare our results with the Cramer-Rao lower bound (CRLB). However, computation of CRLB for our estimation problem is difficult due to the mixture model which involves the latent variables $\mathbf{Z}$ and the hypothesis vector $\mathbf{h}$. However, CLRB can be computed for the case that the identification matrix $\mathbf{Z}$ and the hypothesis vector $\mathbf{h}$ are known. This provides a lower bound to the estimation errors of the proposed method in which $\mathbf{Z}$ and $\mathbf{h}$ are not assumed to be known a priori. For given $\mathbf{Z}$ and $\mathbf{h}$, we define $\zeta_k = \sum_{l=1}^{L} z_{l,k}$, $\mathcal{M} = \sum_{t=1}^{T} h_t$, and $\mathcal{N} = T - \mathcal{M}$. Let $\mathbf{D}_k$ be derived from $\mathbf{D}$ by removing any row j if $z_{j,k} \neq 1$ and let $\mathbf{D}_{k,\eta}$ be obtained from $\mathbf{D}_k$ by removing any column t such that $h_t \neq \eta$. It is clear that the dimension of $\mathbf{D}_{k,0}$ and $\mathbf{D}_{k,1}$ are $\zeta_k \times \mathcal{N}$ and $\zeta_k \times \mathcal{M}$, respectively. Finally, denote by $\mathbf{d}_{k,0}$ (resp. $\mathbf{d}_{k,1}$) the $1 \times \zeta_k\mathcal{N}$ (resp. $1 \times \zeta_k\mathcal{M}$) vector formed by stacking rows of $\mathbf{D}_{k,0}$ (resp. $\mathbf{D}_{k,1}$) next to each other. For any unbiased estimate $\hat{p}_f(k)$, the conditional variance of $\hat{p}_f(k)$ is bounded by [25],

$$\text{var}\{\hat{p}_f(k)|p_f(k)\} \geq \left[E\left[\frac{\partial \ln P(\mathbf{d}_{k,0}\,\mathbf{1}|p_f(k))}{\partial p_f(k)} \right]^2 \right]^{-1} \tag{36}$$

where $\mathbf{1}$ is a column vector of all 1's with length $\zeta_k\mathcal{N}$. Unbiasedness of the proposed algorithm has been shown through extensive simulations some of which is presented in Section VI. For known $\mathbf{Z}$ and $\mathbf{h}$, we have

$$P(\mathbf{d}_{k,0}\,\mathbf{1} = \ell|p_f) = [p_f(k)]^\ell [1 - p_f(k)]^{\zeta_k\mathcal{N} - \ell}. \tag{37}$$

Therefore after some manipulations we get

$$\begin{aligned} & E\left[\frac{\partial}{\partial p_f(k)} \ln P(\mathbf{d}_{k,0}|p_f) \right]^2 \qquad (38) \\ &= \sum_{\ell=0}^{\zeta_k\mathcal{N}} \binom{\zeta_k\mathcal{N}}{\ell} \frac{\ell^2 + \zeta_k^2\mathcal{N}^2[p_f(k)]^2 - 2\ell\zeta_k\mathcal{N}p_f(k)}{[p_f(k)]^{2-\ell}[1 - p_f(k)]^{2-\zeta_k\mathcal{N}+\ell}} \\ &= \frac{\zeta_k\mathcal{N}}{p_f(k)(1 - p_f(k))} \end{aligned}$$

Following the same approach for $\hat{p}_d(k)$, the Cramer-Rao lower bounds are given by

$$\text{var}\{\hat{p}_f(k)|p_f(k)\} \geq \frac{p_f(k)(1 - p_f(k))}{\zeta_k\mathcal{N}}, \tag{39}$$

$$\text{var}\{\hat{p}_d(k)|p_d(k)\} \geq \frac{p_d(k)(1 - p_d(k))}{\zeta_k\mathcal{M}}. \tag{40}$$

VI. Numerical Results

In this section, employing the metrics in Section V, we evaluate the performance of the proposed method referred to as maximum-likelihood classifier (MLC) and also compare our results with the reputation-based classifier (RBC) algorithm [12], [26]. In RBC when the network parameters (e.g., the nodes' operating points) are known, the optimal q-out-of-L rule can be computed (see for example [16], [27]). However, when the FC is not aware of all the network parameters as is the case here, majority rule has been used in [12] and is also used here for our comparisons. In addition, in (45) the threshold λ can be set following a Neyman-Pearson criterion, for example by setting a threshold on the probability of misclassifying the honest nodes as Byzantines. Moreover, if the fraction of honest nodes is known to the FC as in [12], then λ can be set to minimize the probability of classification error.

TABLE I
CLASS PARAMETERS OF EACH SET OF OPERATING POINTS

Set	p_f	p_d	π
OP1	0.1	0.9	0.6
	0.9	0.3	0.4
OP2	0.2	0.7	0.6
	0.9	0.15	0.4
OP3	0.2	0.7	0.4
	0.9	0.15	0.15
	0.9	0.9	0.2
	0.05	0.05	0.25

In our case, however, the FC is not aware of the fraction of honest nodes. Therefore we set the threshold $\lambda = .5$. For this choice of λ the probability that an honest node is misclassified as Byzantine is the same as the probability that a Byzantine node is misclassified as honest. Other values of the threshold can favor the classification of honest nodes as Byzantines or vice versa.

Simulation results are obtained from at least 10^4 independent trials. The EM algorithm is assumed to have converged when $\left\|\boldsymbol{\Theta}^{\text{new}} - \boldsymbol{\Theta}^{\text{old}}\right\| < \epsilon = 10^{-3}$. Moreover, to overcome the ambiguity of the counterpart networks, we assume that the honest nodes are in majority. This implies that for a network consisting of two classes the break down point of the algorithm is at 50% [28]. In Figs. 1, 2 and 9-12 where a performance metric is presented vs. T, the number of possible hypothesis vectors 2^T is too large to evaluate (29) exhaustively. Therefore in these cases it is assumed that during the observation period there is at most one change in the hypothesis vector $\mathbf{h}$ which may occur at random anywhere from time 2 to $T-1$. This assumption, which as mentioned in Section III-D is applicable in practice, is only made to reduce the computational complexity of our simulations. However, the efficacy of the proposed method is not affected by this assumption as other figures verify.

Three sets of operating points, denoted OP1, OP2 and OP3, are considered. Table I shows the class parameters corresponding to each operating point. For OP1 and OP2 there are two classes of honest and Byzantine nodes. The FC perceives the operating point of the Byzantines, (p_f, p_d), to be that listed in Table I. One may view the Byzantines as having an *actual operating point* $(1-p_f, 1-p_d)$, but flipping their decisions before transmission to the FC. Comparing the operating point of honest nodes and the actual operating point of Byzantine nodes in OP2 reveals that the Byzantine nodes are more capable of detecting the event under both hypotheses (i.e., with smaller probability of false alarm and higher probability of detection). For OP3, four classes of nodes are considered. The first class with the operating point $(.2, .7)$ comprises the honest nodes. The second class are Byzantine nodes with the operating point $(.9, .15)$, while the third and fourth classes are "almost-always-yes" and "almost-always-no" nodes. The almost-always-yes nodes try to convince the FC that the hypothesis is H_1 by transmitting a 1 most of the times, and increase the overall false alarm rate of the system. In contrast, the almost-always-no nodes transmit a 0 most of the time and decrease the overall probability of detection.

Figs. 1 and 2 show the performance of the classifiers vs. the number of received decisions, T. It is evident that the accuracy of node classification and the estimation of the operating points improve with T. Moreover the proposed algorithm converges much faster than the reputation-based method requiring fewer number of observation samples. Note that since RBC can only discriminate nodes into two classes, in the case of OP3 Δ_Z is not defined. The figures also show that the performance of classifiers for OP1 is better than for OP2 and OP3. The reason is that the misbehaving nodes are more capable in the latter two cases. In particular in the case of OP2, the RBC method fails completely. This is due to the fact that even though only 40% of the nodes are Byzantine, because of their operating point $(0.9, 0.15)$ vs. the operating point of the honest nodes $(0.2, 0.7)$, collectively the Byzantine nodes are more capable than the honest nodes and can mislead the FC.

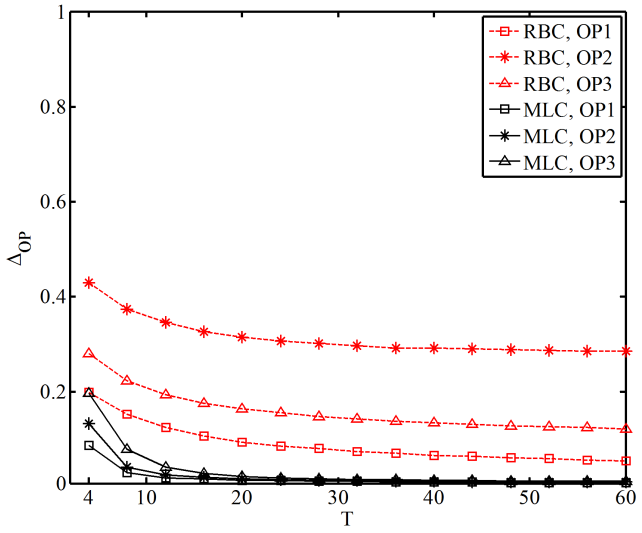

Fig. 1. Error in the estimation of the operating points vs. T for $L = 100$ nodes.

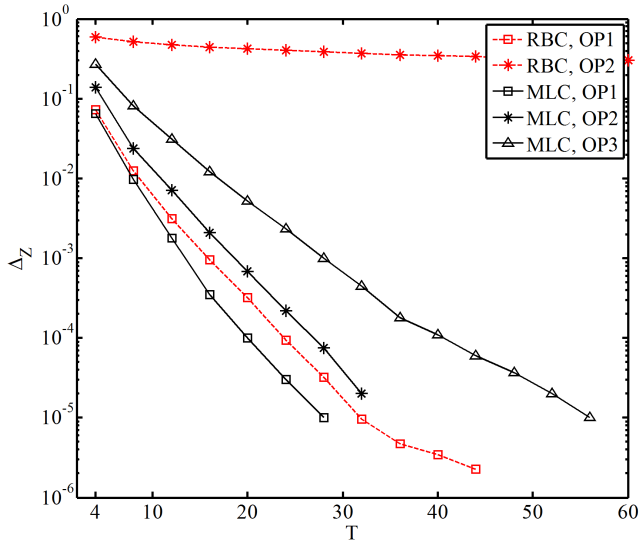

Fig. 2. Misclassification rate vs. T for $L = 100$ nodes.

Figs. 3 and 4 compare the performance of the classifiers vs. the ratio of the honest nodes to the total number of nodes

(denoted by α) for $T = 10$. The operating points are OP1 and OP2 shown in Table I. As expected the performance of the classifiers improves with α. It is seen that while RBC can effectively classify the nodes in the case of OP1, the computation of the operating points is not very accurate. Moreover for OP2 the performance of RBC is not acceptable and fails completely for $\alpha \leq .6$.

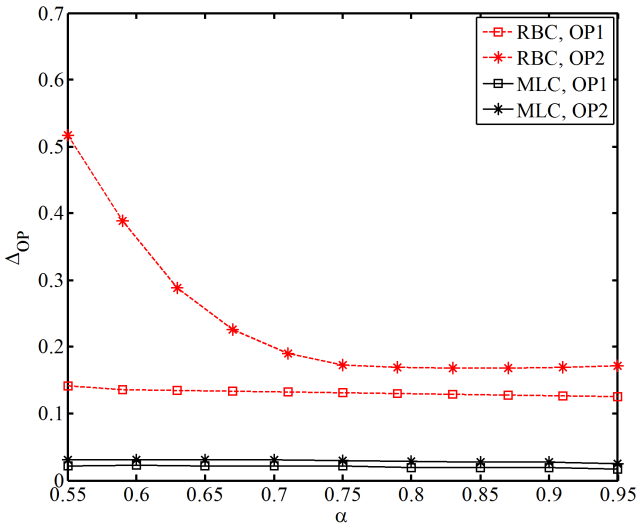

Fig. 3. Error in the estimation of the operating points vs. α for $T = 10$ and $L = 100$.

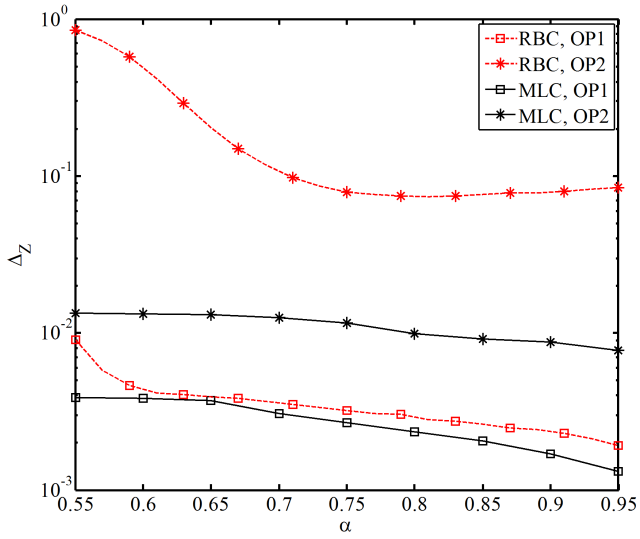

Fig. 4. Misclassification rate vs. α for $T = 10$ and $L = 100$.

In Figs. 5, 6 and 7 we compare the performance of the classifiers vs. the number of nodes L for $T = 4$ samples. For OP1, as the number of nodes increases, the classifier errors converge to zero. Again for OP2, the error for RBC does not converge to zero due to the fact that in this case the Byzantine nodes are collectively more capable than the honest nodes.

Figs. 8 and 9 show the efficacy of the proposed estimation method by comparing the variance of the estimated false alarm probability of the honest nodes and the Cramer-Rao lower bound of Section V-A. As these figures demonstrate, the

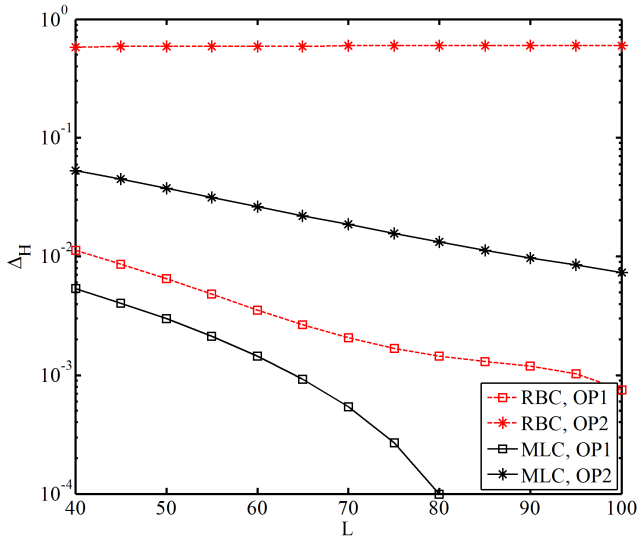

Fig. 5. Hypothesis discriminability vs. L for $T = 4$.

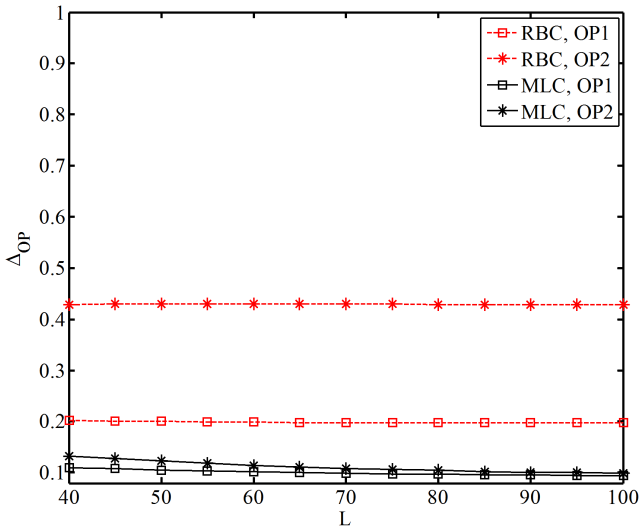

Fig. 6. Error in the estimation of the operating points vs. L for $T = 4$.

accuracy of the estimation increases as number of observations or number of nodes increases.

To show the robustness of the proposed method to possible time varying behavior of the Byzantine nodes, we consider a case where the Byzantines change their operating point during the observation period. Two classes of nodes are considered. The honest nodes have an operating point $(p_f, p_d) = (0.1, 0.8)$. For Byzantine nodes, for each time t, the probabilities of false alarm and detection are chosen at random with uniform distribution on $[0.75-.2,\ 0.75+.2]$ and $[0.3-.2,\ 0.3+.2]$, respectively. Moreover, these probabilities are independent for each time $t = 1, 2, \ldots, T$ and for each node. Finally the fraction of the Byzantine nodes is $\pi_2 = 0.4$. Figs. 10, 11 and 12 show Δ_{OP}, Δ_Z and Δ_H versus T, respectively. In evaluating Δ_{OP} for Byzantines we have compared the mean of their operating point given by $(.75, .3)$ with the estimated operating point. We also show the results for the case where the operating point of the Byzantines is fixed and

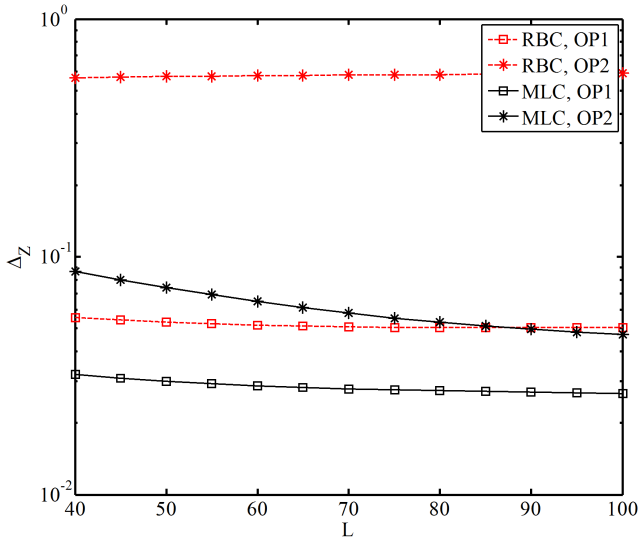

Fig. 7. Misclassification rate vs. L for $T = 4$.

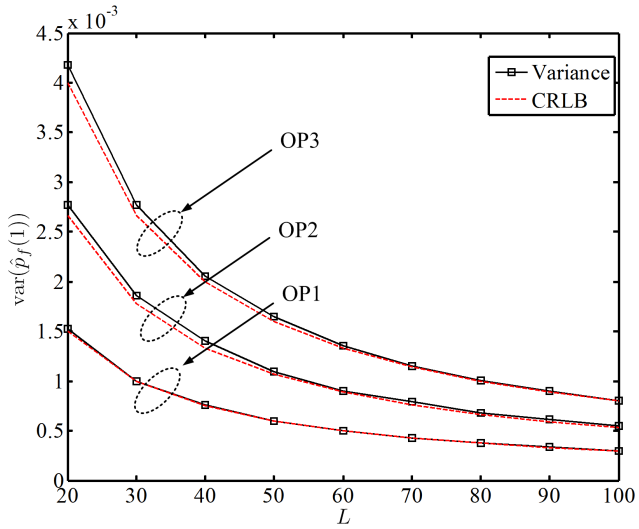

Fig. 8. The variance of $\hat{p}_f(1)$ and the Cramer-Rao lower bound vs. L for $T = 10$.

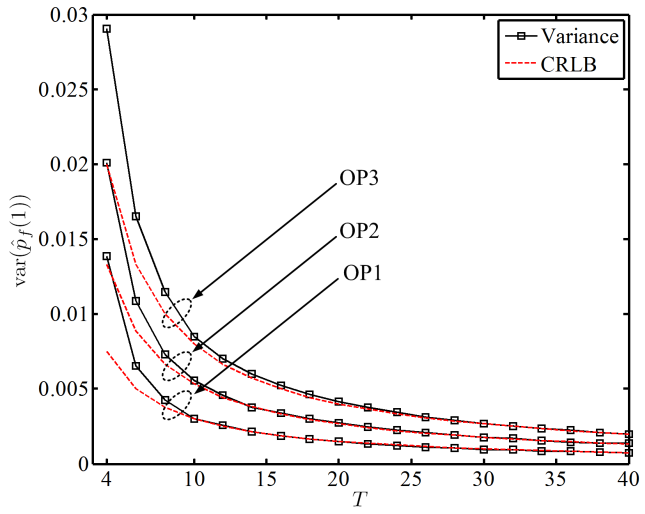

Fig. 9. The variance of $\hat{p}_f(1)$ and the Cramer-Rao lower bound vs. T for $L = 10$.

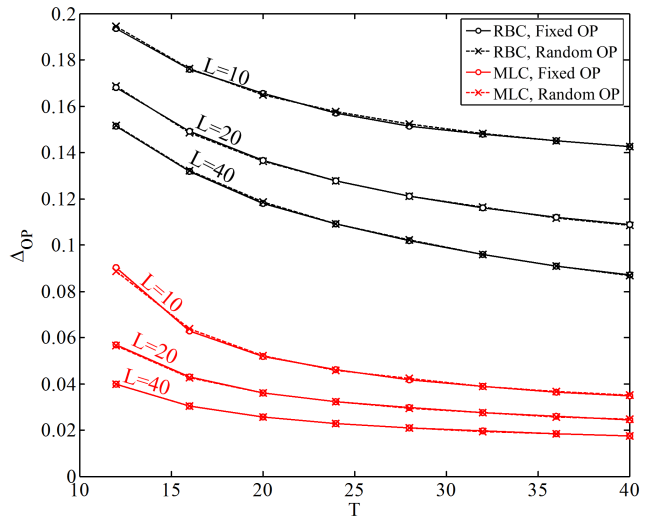

Fig. 10. Comparison of the error in the estimation of the operating points vs. T for fixed and randomly varying Byzantine operating points.

is equal to $(.75, .3)$. It can be seen that, as in the case of fixed operating points, the proposed method outperforms the RBC method. Moreover, the performances are very close for the two cases of fixed and randomly varying operating points. This can be explained by the fact that the estimation of probabilities of false alarm and detection in EM are obtained by evaluating the *average* number of ones transmitted under H_1 and H_0 as shown in (21) and (22).

VII. Conclusion

We consider the problem of decentralized detection in the presence of one or more classes of misbehaving nodes. The fusion center first estimates the nodes' operating points (false alarm and detection probabilities) on the ROC curve and then uses this estimation to classify the nodes and to detect the state of nature. We formulate and solve this problem in the framework of expectation maximization algorithm. Numerical results are presented that show the proposed algorithm significantly outperforms the reputation-based methods in classification of the nodes as well as the detection of the hypotheses. The estimated operating points are compared to the Cramer-Rao lower bound which shows the efficacy of the proposed method.

Appendix

A. Operating Region of Misbehaving Nodes

Consider a node in class c_k with the operating point $(\tilde{p}_f(k), U_k(\tilde{p}_f(k)))$ on its ROC curve. We show that by appropriate selection of $\rho_0(k)$ and $\rho_1(k)$ in (5)-(6), a desired operating point $(p_f(k), p_d(k))$ can be achieved in the region bounded by $(\tilde{p}_f(k), U_k(\tilde{p}_f(k)))$ and $(\tilde{p}_f(k), V_k(\tilde{p}_f(k)))$ where $V_k(x) = 1 - U_k(1 - x)$.

Consider Fig. 13. Denote by $A = (\tilde{p}_f(k), \tilde{p}_d(k))$ the operating point of a node and by $B = (1 - \tilde{p}_f(k), 1 - \tilde{p}_d(k))$ the reflection of A at $(0.5, 0.5)$. We consider two cases.

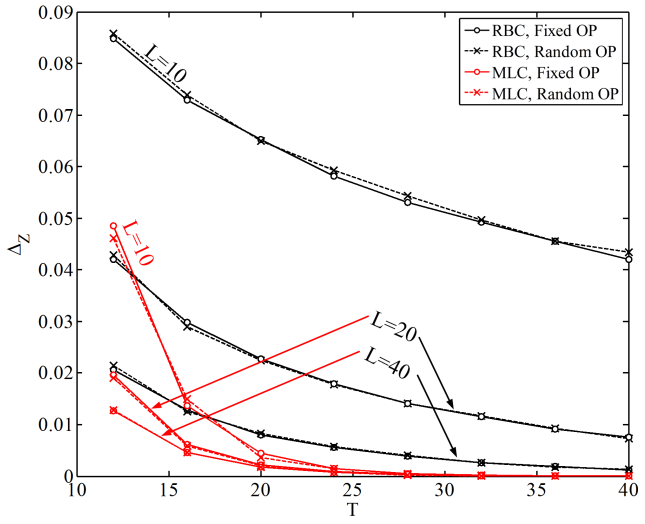

Fig. 11. Comparison of the misclassification rate vs. T for fixed and randomly varying Byzantine operating points.

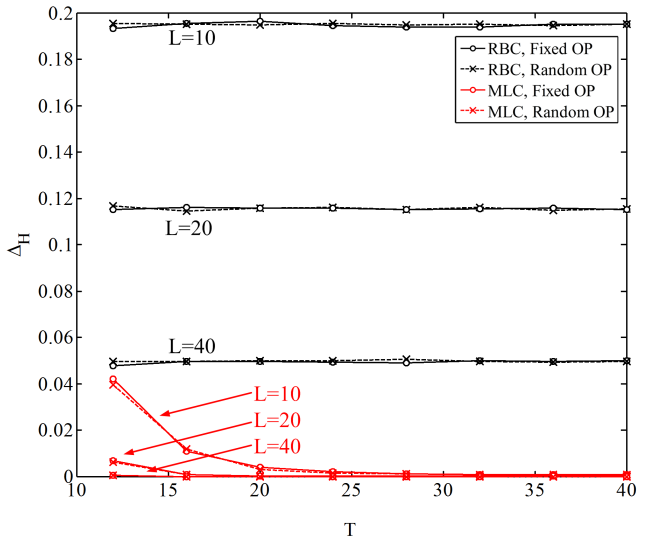

Fig. 12. Comparison of hypothesis discriminability vs. T for fixed and randomly varying Byzantine operating points.

1) Fixed $\rho_0(k)$: From (5) and (6), for fixed $\rho_0(k) = \delta$ we get

$$p_d(k) = m_\alpha p_f(k) + \delta(1 - m_\delta), \tag{41}$$

where $m_\delta \triangleq \frac{\tilde{p}_d(k)}{\tilde{p}_f(k)}$ is the slope of the line between the origin and $A = (\tilde{p}_f(k), U_k(\tilde{p}_f(k)))$. Therefore in this case $(p_f(k), p_d(k))$ is located on a set of parallel lines with slope m_δ and the y-intercept starting from the origin (corresponding to $\delta = 0$) up to $1 - m_\delta$ (corresponding to $\delta = 1$).

2) Fixed $\rho_1(k)$: Similar to the previous case, for fixed $\rho_1(k) = \beta$ and using (5) and (6), one can write

$$p_d(k) = m_\beta p_f(k) + \beta(1 - m_\beta) \tag{42}$$

where $m_\beta \triangleq \frac{1-\tilde{p}_d(k)}{1-\tilde{p}_f(k)}$ is the slope of line OB. As a result, in this case the region of operating points $(p_f(k), p_d(k))$ is a set of parallel lines with slope m_β and the y-intercept starting from the origin ($\beta = 0$) and up to $1 - m_\beta$ ($\beta = 1$).

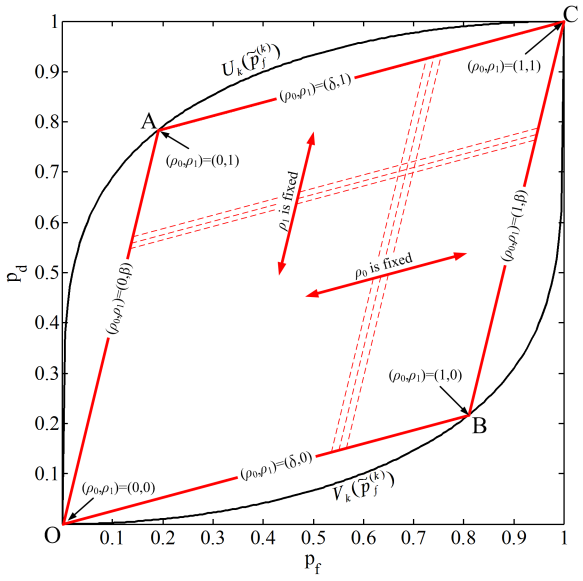

Fig. 13. Region of achievable operating points for the nodes.

Combining the two cases above we see that the loci of the operating point of the node will be in the parallelogram $OACB$ where points O and C correspond to $\rho_0(k) = \rho_1(k) = 0$ and $\rho_0(k) = \rho_1(k) = 1$, respectively.

Consider a Byzantine node l in class c_k. With its transmitted message $d_{l,t}$, this node attempts to mislead the FC regarding the state of h_t. For this, however, the Byzantine must first detect the sate of h_t as represented by $r_{l,t}$. There is an ROC and an operating point (denoted by $(\tilde{p}_f(k), \tilde{p}_d(k))$ in Section II) associated with this detection rule. Since the transmitted message $d_{l,t}$ must be based on this detection ($r_{l,t}$), the above results show that the operating point as perceived by the FC $(p_f(k), p_d(k))$ cannot be arbitrary and must lie in the region described above.

B. Reputation-Based Node Classifier

Voting rules or q-out-of-L rules [2] are commonly employed in the FC to detect the occurrence of an event in decentralized sensing [10], [11], [26], [29], [30]. Based on this rule, the detected hypothesis is H_1 if at least q out of L nodes vote in favor of this event. When $q = 1$, $q = L$ and $q = L/2$, this rule is denoted by "OR-rule", "AND-rule", and the "Majority-rule", respectively.

The operating point of the lth node, $1 \le l \le L$, can be estimated using the transmitted decisions of the node under the estimated hypothesis, i.e.,

$$\hat{p}_f(l) = \frac{\sum_{t=1}^{T}(1 - \hat{h}_t)d_{l,t}}{T - \sum_{t=1}^{T}\hat{h}_t} \tag{43}$$

$$\hat{p}_d(l) = \frac{\sum_{t=1}^{T}\hat{h}_t d_{l,t}}{\sum_{t=1}^{T}\hat{h}_t}, \tag{44}$$

where $\hat{h}_t$, $1 \le t \le T$ is the detected hypothesis from the voting rule at time t, and $d_{l,t}$ is the corresponding transmitted decision of the lth node.

The reputation-based classification [12] is based on the reputation metric, R_l, given by

$$R_l \triangleq \frac{T - \sum_{t=1}^{T} |d_{l,t} - \hat{h}_t|}{T} \underset{\text{Byzantine}}{\overset{\text{Honest}}{\gtrless}} \lambda, \qquad (45)$$

In other words, a node belongs to the class of honest nodes if the fraction of its decisions that do not match the detected hypotheses is less than some threshold η.

REFERENCES

[1] M. Franceschelli, A. Giua, and C. Seatzu, "Decentralized fault diagnosis for sensor networks," in *Automation Science and Engineering, 2009. CASE 2009. IEEE International Conference on*, aug. 2009, pp. 334 – 339.

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

[3] S. Marano, V. Matta, and L. Tong, "Distributed inference in the presence of Byzantine sensors," in *Signals, Systems and Computers, 2006. ACSSC '06. Fortieth Asilomar Conference on*, 29 2006-nov. 1 2006, pp. 281 – 284.

[4] ——, "Distributed detection in the presence of Byzantine attacks," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 16 –29, jan. 2009.

[5] M. Abdelhakim, L. E. Lightfoot, and T. Li, "Reliable data fusion in wireless sensor networks under Byzantine attacks," in *MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011*, nov. 2011, pp. 810 –815.

[6] M. Gagrani, P. Sharma, S. Iyengar, V. Nadendla, A. Vempaty, H. Chen, and P. Varshney, "On noise-enhanced distributed inference in the presence of Byzantines," in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, sept. 2011, pp. 1222 –1229.

[7] R. Chen, J.-M. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, april 2008, pp. 1876 –1884.

[8] A. Rawat, P. Anand, H. Chen, and P. Varshney, "Countering Byzantine attacks in cognitive radio networks," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, march 2010, pp. 3098 –3101.

[9] P. Anand, A. Rawat, H. Chen, and P. Varshney, "Collaborative spectrum sensing in the presence of Byzantine attacks in cognitive radio networks," in *Communication Systems and Networks (COMSNETS), 2010 Second International Conference on*, jan. 2010, pp. 1 –9.

[10] M. Abdelhakim, L. Zhang, J. Ren, and T. Li, "Cooperative sensing in cognitive networks under malicious attack," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, may 2011, pp. 3004 –3007.

[11] H. Wang, L. Lightfoot, and T. Li, "On phy-layer security of cognitive radio: Collaborative sensing under malicious attacks," in *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, march 2010, pp. 1 –6.

[12] A. Rawat, P. Anand, H. Chen, and P. Varshney, "Collaborative spectrum sensing in the presence of Byzantine attacks in cognitive radio networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 774 –786, feb. 2011.

[13] F. Penna, Y. Sun, L. Dolecek, and D. Cabric, "Detecting and counteracting statistical attacks in cooperative spectrum sensing," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 1806 –1822, april 2012.

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

[15] B. Chen, R. Jiang, T. Kasetkasem, and P. Varshney, "Channel aware decision fusion in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 52, no. 12, pp. 3454 – 3458, dec. 2004.

[16] Q. Zhang, P. Varshney, and R. Wesel, ""Optimal bi-level quantization of i.i.d. sensor observations for binary hypothesis testing"," *IEEE Transactions on Information Theory*, vol. 48, no. 7, pp. 2105 –2111, jul 2002.

[17] R. Niu, B. Chen, and P. Varshney, "Fusion of decisions transmitted over rayleigh fading channels in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1018 – 1027, march 2006.

[18] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*, 1st ed. Upper Saddle River, New Jersey, USA: Prentice Hall, 1998.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, vol. 39, no. 1, pp. 1–38, 1977.

[20] A. R. Webb, *Statistical Pattern Recognition*, 2nd ed. Chichester, West Sussex, England: John Wieley & Sons, 2001.

[21] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[22] S. Boyd and L. Vandenberghe, *Convex Optimization*, 1st ed. New York: Cambridge University Press, 2004.

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

[24] D. J. Hand, *Construction and Assessment of Classification Rules*, 1st ed. Chichester, West Sussex, England: John Wiley & Sons, 1997.

[25] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, 1st ed. Upper Saddle River, New Jersey, USA: Prentice Hall, 1993.

[26] X. Luo, M. Dong, and Y. Huang, "On distributed fault-tolerant detection in wireless sensor networks," *IEEE Transactions on Computers*, vol. 55, no. 1, pp. 58 – 70, jan. 2006.

[27] W. Zhang, R. Mallik, and K. Ben Letaief, "Cooperative spectrum sensing optimization in cognitive radio networks," in *Communications, 2008. ICC '08. IEEE International Conference on*, may 2008, pp. 3411 –3415.

[28] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*. New York: John Wiley & Sons, Inc., 1987.

[29] R. Viswanathan and P. Varshney, "Distributed detection with multiple sensors i. fundamentals," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 54 –63, jan 1997.

[30] R. Soosahabi and M. Naraghi-Pour, "Scalable phy-layer security for distributed detection in wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. PP, no. 99, p. 1, 2012.

Erfan Soltanmohammadi (S12) was born in Karaj, Iran, in 1984. He received the B.Sc. in electrical engineering from Khaje Nasir University of Technology (KNTU), Tehran, Iran, in 2007, and the M.S. from Amirkabir University of Technology (AUT), Tehran, Iran, in 2010. He is currently working towards the Ph.D. degree in systems (communication & signal processing) in the School of Electrical Engineering and Computer Science, Louisiana State University, Baton Rouge, Louisiana, U.S.A, where he is also a Graduate Research/Teaching Assistant. His current research interests include security in wireless sensor networks, cognitive radio, signal processing for communications, MIMO systems, and blind communication techniques.

Mahdi Orooji (S'11) was born in Tehran, Iran, in 1980. He received the B.Sc. degree in electrical engineering from University of Tehran in 2003. He is currently working towards the Ph.D. degree in the School of Electrical Engineering and Computer Science, Louisiana State University, Baton Rouge, Louisiana, USA. His research interests are wireless communication and statistical signal processing. Mr. Orooji received the Huel D. Perkins Doctoral Fellowship Award from LSU, 2009-2013.

Mort Naraghi-Pour (S'81-M'87) was born in Tehran, Iran, on May 15, 1954. He received the B.S.E. degree from Tehran University, Tehran, in 1977 and the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, in 1983 and 1987, respectively. In 1978, he was a student at the Philips International Institute, Eindhoven, The Netherlands, where he also did research with the Telecommunication Switching Group of the Philips Research Laboratories. Since August 1987, he has been with the School of Electrical Engineering and Computer Science, Louisiana State University, Baton Rouge, where he is currently an Associate Professor. From June 2000 to January 2002, he was a Senior Member of Technical Staff at Celox Networks, Inc., a network equipment manufacturer in St. Louis, MO. His research and teaching interests include wireless communications, broadband networks, information theory, and coding.

Dr. Naraghi-Pour has served as a Session Organizer, Session Chair, and member of the Technical Program Committee for many international conferences.