Trust-aware Consensus-inspired Distributed Cooperative Spectrum Sensing for Cognitive Radio Ad Hoc Networks

Aida Vosoughi, *Student Member, IEEE,* Joseph R. Cavallaro, *Fellow, IEEE,* and Alan Marshall, *Senior Member, IEEE*

***Abstract*—Cooperation among cognitive radios for spectrum sensing is deemed essential for environments with deep shadows. In this paper, we study cooperative spectrum sensing for cognitive radio ad hoc networks where there is no fusion center to aggregate the information from various secondary users. We propose a novel consensus-inspired cooperative sensing scheme based on linear iterations that is fully distributed and low-cost. In addition, the trade-offs on the number of consensus iterations are explored for scenarios with different shadow fading characteristics. Furthermore, we model Insistent Spectrum Sensing Data Falsification (ISSDF) attack aimed at consensus-based iterative schemes and show its destructive effect on the cooperation performance which accordingly results in reduced spectrum efficiency and increased interference with primary users. We propose a trust management scheme to mitigate these attacks and evaluate the performance improvement through extensive Monte Carlo simulations for large-scale cognitive radio ad hoc networks in TV white space. Our proposed trust management reduces the harm of a set of collusive ISSDF attackers up to two orders of magnitude in terms of missed-detection and false alarm error rates. Moreover, in a hostile environment, integration of trust management into cooperative schemes considerably relaxes the sensitivity requirements on the cognitive radio devices.**

***Index Terms*—Dynamic spectrum access, Cognitive radio ad hoc networks, Distributed consensus-based cooperative spectrum sensing, Trust management, Insistent spectrum sensing data falsification attack**

I. Introduction

THE radio frequency spectrum shortage problem is originated from the static assignment of the frequency bands to the primary users (PUs or licensees) of the bands. The non-adaptive spectrum assignment leaves a significant portion of RF spectrum underutilized [1]. Dynamic spectrum access (DSA), enabled by cognitive radios, introduces an adaptive approach for spectrum use that facilitates more flexibility by allowing secondary users (SUs) to use licensed spectrum bands on an opportunistic non-interference basis. As a result, DSA offers a better utilization of the spectrum and is essential for solving the spectrum shortage problem. Cognitive radios that sense and dynamically share the spectrum empower today's smart technologies such as cognitive Internet of Things [2] and heterogeneous networks with cognitive Femtocells [3].

Spectrum sensing is an important step for DSA. However, when an SU senses the spectrum, it is possible that it does not detect the PU due to a deep shadow and this in turn increases the risk of interference to the PU. In order to improve the SUs' detection accuracy *cooperative spectrum sensing* has been proposed. In this approach, a set of SUs cooperate by sharing their sensing information with each other and collectively deciding on the presence or absence of the PU [4]. In a centralized cognitive radio network (e.g. IEEE 802.22 [5]), the final decision is made by a fusion center that aggregates the sensing data from all of the SUs in the network. In contrast, in a decentralized network (e.g. a cognitive radio ad hoc network or CRAHN), the nodes must perform a distributed cooperation. Distributed cooperative spectrum sensing (DCSS) is preferred to a centralized scheme because a distributed scheme is scalable, fault-tolerant and more efficient.

DCSS is performed by exploiting existing distributed consensus algorithms that have been previously used for applications such as sensor fusion [6] or Peer-to-Peer systems [7]. These consensus algorithms are based on iterative diffusion and aggregation of data through linear iteration-based or gossip-based schemes [8] and involve communication with direct neighbors in the network graph. However, the consensus-based DCSS schemes that have been proposed previously are not practical for ad hoc networks as they require the individual nodes to have knowledge about the topology of the network.

Another known issue with cooperative schemes is that in a realistic potentially hostile environment, malicious SUs can broadcast falsified sensing data to their neighbors in order to mislead them and compromise the spectrum sharing in the cognitive radio network. This attack is called Spectrum Sensing Data Falsification (SSDF) [9] attack. A more serious and less studied attack particularly aimed at consensus-based schemes is an iterative attack that we call Insistent SSDF (ISSDF). The ISSDF attacker not only falsifies its own initial data but it also broadcasts the falsified value in every iteration of the consensus and refrains from performing updates according to the protocol. The ISSDF attack compromises the cooperation significantly and it may cause divergence from the correct consensus. In order to address the above-mentioned problems, we introduce a trust-aware consensus-inspired DCSS scheme which is low-overhead and resilient to ISSDF attacks. Our contributions can be summarized as follows:

- We propose a practical distributed scheme for cooperative spectrum sensing in cognitive radio ad hoc networks that is inspired by a linear iterative average consensus algorithm and uses an equal-weighting update strategy that does not require any topology knowledge by SUs. Through extensive simulations for realistic large-scale

A. Vosoughi and J. R. Cavallaro are with the Department of Electrical and Computer Engineering, Rice University, Houston, TX, 77005 USA e-mail: {vosoughi, cavallar}@rice.edu.

A. Marshall is with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, UK email: alan.marshall@liverpool.ac.uk

mobile networks in outdoor environments with correlated shadow fading, we show that our proposed scheme offers the same level of performance compared to the existing more complex consensus-based schemes.
- We analyze the performance-complexity trade-offs on the number of consensus iterations for a typical simulated network under different shadowing severities.
- We show the significant potential of the ISSDF attackers in crippling the consensus-based schemes. We propose a trust management scheme that can be integrated with any consensus-based DCSS scheme to mitigate the ISSDF attacks. We show that our proposed trust-aware DCSS scheme is robust even in the presence of a large set of ISSDF attackers that act in harmony and simultaneously. In addition, we propose a trust-aided outlier detection technique that when combined with the proposed trust scheme can effectively mitigate dynamic attackers.
- We analyze the impact of malicious attacks and trust management mitigations on the sensitivity requirements of cognitive radio devices which has direct relationship with the system's cost and flexibility.

II. Background and Related Work

Recently, average consensus algorithms [8] including gossip-based protocols [7] and linear iteration-based schemes [6], [10] have been exploited for the DCSS applications [11]–[14]. However, all of the existing consensus-based DCSS schemes require the individual SUs to have some type of knowledge about the network topology. For instance, some of these schemes require the nodes to know the maximum degree in the network (or at least an upper bound), while others require the nodes to know the degree of the neighbor nodes [6]. These limitations make the existing DCSS schemes impractical for cognitive radio ad hoc networks. In this paper, we propose a consensus-inspired DCSS scheme that is practical for a dynamic network because the SUs are completely topology-agnostic.

The other significant issue in the current cooperative spectrum sensing schemes is ensuring the robustness of the cooperation against malicious SSDF [9] attackers that broadcast falsified sensing data. Moreover, in the context of iterative consensus-based DCSS schemes, ISSDF attackers, that do not follow the consensus update protocol and broadcast falsified data in every iteration, are much more destructive than the conventional SSDF attackers. In addition, a set of collusive ISSDF attackers can amplify the effect of each-other. Sundaram et. al. prove that a set of conspiring malicious nodes, who do not follow the update protocol, are able to prevent the network from converging to the correct answer [15], [16].

ISSDF attackers are in a sense similar to the stubborn agents [17] that have been studied in the context of opinion propagation and convergence. It is shown that the stubborn agents can cause the network to converge to their opinions. Moreover, the optimal placement of stubborn agents for maximized impact on a fixed network is investigated [17]–[19]. In contrast, in this paper, we consider a mobile network of SU nodes, where a random subset of nodes are ISSDF attackers and they move randomly similar to the normal nodes. We do not make any assumption that ISSDF attackers collude to move in a way to maximize their effect on the network. This may be an interesting scenario for further research.

The conventional SSDF attacks and mitigation approaches against them have been well-studied in the literature for the centralized schemes [9], [20]–[22]; however, the problem of coping with ISSDF attacks in the consensus-based DCSS schemes is hardly explored. A proposed approach to mitigate the effect of ISSDF attackers in consensus-based DCSS schemes is adaptive outlier detection [23], [24] which is based on detecting the nodes that broadcast values that are deviated from the the rest of the neighbors. This approach is distributed however, it requires every node to compute a deviation threshold at each consensus iteration which imposes a significant computational overhead on each SU. As will be described in the following sections, in our proposed scheme the SUs update the trust scores only once the consensus iterations are completed and therefore the computational overhead is low. Zhang et. al. propose a weighted average consensus scheme to count for channel conditions and multi-path fading in DCSS, however, they do not address the ISSDF attacks nor the impact of correlated shadow fading.

In this paper, we introduce trust scores as weights for the average consensus update rule to mitigate ISSDF attacks. Liu et. al. [25] propose a trust scheme using trust propagation and a set of pre-trusted nodes to mitigate the effect of Byzantine adversaries in linear iterative consensus in sensor networks. However, trust propagation is costly and generally there are no pre-trusted nodes in an ad hoc network. A trust-aware DCSS based on single neighbor gossip (sharing binary decisions) has been proposed in [12] which can mitigate SSDF attacks; however, due to the nature of wireless networks, this model is less efficient compared to a broadcast model which we consider in this paper. In addition, in this paper we consider the SU nodes share raw PU power values.

In this paper, we extend our previous work [26] by analyzing our trust-aware consensus-inspired DCSS scheme for a mobile CRAHN in realistic environments with correlated shadow fading of various severity. In addition, we study the trade-offs that determine the best choice for number of consensus iterations. Moreover, we analyze the operating characteristics of a CRAHN under various detection thresholds in the presence of ISSDF attackers and show the significant improvement through trust management. Our trust management scheme does not depend on pre-trusted nodes and only requires the nodes to perform a single local trust evaluation per sensing round for each direct neighbor. These features make our proposed scheme practical and low-cost for CRAHNs.

III. System Model

Our model consists of a network of n SUs that form a CRAHN in a square location area which is far away from a PU transmitter. The PU transmitter is assumed to have a high transmission power (e.g. a TV station), therefore the whole SU network is within the transmission range of the PU transmitter. A network of PU receivers are also collocated in the same area.

See Figure 1 for the system overview. SU nodes are initially uniformly spread throughout the location area and during the time of simulation, they move randomly. The neighbor set of the SU node i, denoted by N_i, consists of all of the SUs that are located within the communication range of SU node i. Obviously, the neighbor sets are always changing due to the mobility of the nodes; however, we assume the SU network topology remains unchanged during one sensing period. When node i broadcasts a message, all of its one-hop neighbors will receive that message. Here we assume perfect communication between the SUs via a common control channel. The detection of a PU is modeled as a binary hypothesis testing problem as follows: H_0 if PU is absent and H_1 if PU is present. Each SU node is equipped with a power detector for sensing the received power from the PU. The received signal by an SU can be modeled as follows:

$$y(m) = \begin{cases} w(m) & H_0 \\ s(m) + w(m) & H_1 \end{cases} \quad (1)$$

where $s(m)$ is the signal component with power P_S and $w(m)$ is the zero-mean additive white Gaussian noise with noise power P_N. When the PU is inactive, the sensed power at an SU will essentially be equal to the received noise power. On the other hand, when the PU is active, the signal component power P_S can be modeled as $P_T - PL(d)$, where P_T is the PU transmission power and $PL(d)$ is the distant-dependent path loss from the PU to the SU. Details on the path loss and fading model is given in subsection III-A. If the power detector takes M samples, the test statistic is given by: $\Gamma = \frac{1}{M}\sum_{m=1}^{M} y(m)y(m)^*$. Using the central limit theorem, it can be shown that for large enough M [27] [28], the test statistic for a detector follows a normal distribution [29]:

$$\Gamma \sim \mathcal{N}(P_S + P_N, \frac{2(P_S + P_N)^2}{M}) \quad (2)$$

In a non-cooperative scenario, an SU node decides on the PU activity by comparing its own received power test statistic, Γ, with a detection threshold, γ. The spectrum sensing performance is characterized by the probability of false alarm (P_{FA}) and missed-detection (P_{MD}):

$$P_{FA} = Pr(\Gamma > \gamma | H_0) \quad and \quad P_{MD} = Pr(\Gamma < \gamma | H_1) \quad (3)$$

Therefore, setting the right detection threshold is important in the performance of a cognitive radio. Obviously, the sensitivity of a device puts a minimum bound on the detection threshold. The IEEE 802.22 working group for spectrum sensing modeling recommends setting $P_{FA} = 1\%$ or 10% for an independent SU and deriving the detection threshold based on that [29] [30]. False alarms occur when H_0 is true; therefore, in order to derive the threshold for a desired P_{FA}, we set $P_S = 0$ in (2) and based on (3) we will have:

$$\gamma = P_N(1 + \sqrt{\frac{2}{M}} Q^{-1}(P_{FA})) \quad (4)$$

where $Q^{-1}(.)$ is the inverse Gaussian Q-function. In our cooperative spectrum sensing model, the SU nodes first sense and measure the received power and then share their power measurements with each-other to estimate the average received power. After a number of broadcast and update iterations, each SU compares its own estimate of the average power with a threshold to make its final binary decision about the PU presence. In our cooperative model, we also consider a set of collusive malicious nodes that perform ISSDF attack (See Section VII) to evaluate their impact on the performance.

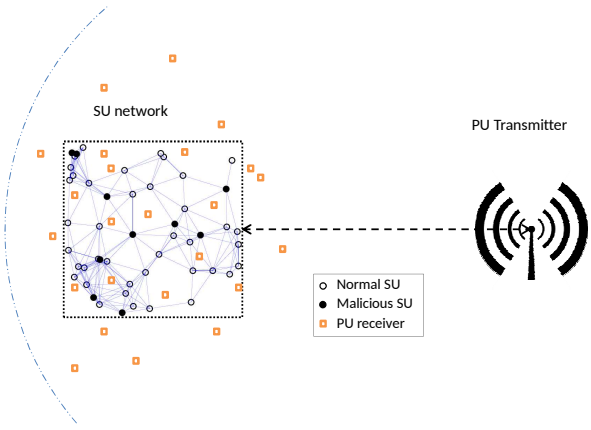


Fig. 1. System Overview

A. Path loss and shadow fading model

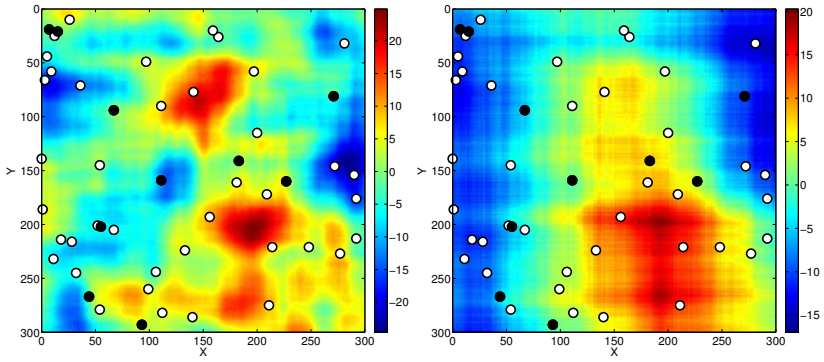
A radio propagation model (analytical or empirical), provides an average path loss for a given transmitter-receiver distance. In our model, we apply Hata path loss model (suburban areas variant) [31]. The IEEE 802.22 working group recommends the Hata model for spectrum sensing in wireless regional area networks operating in TV whitespace.

In addition, a signal transmitted through a wireless link naturally experiences random variations due to obstacles in the path. As a result, two receivers at two different locations with equal distance from a transmitter will not be affected by the same path loss despite the fact that the average path loss is the same at both locations. The random variation about the average path loss due to blockage of objects in the signal path such as buildings and trees is called shadow fading.

It is safe to assume that shadow fading remains constant at a single location over time since normally there is no significant change in the terrain such as the surrounding buildings or trees (a space-time correlated shadow fading model [32] may be used if the shadows are not constant over time.) Obviously, the reception of the mobile radio nodes changes when they move in and out of shadows over time. The loss due to shadow fading is commonly modeled by a random variable with log-normal distribution [31]. That is the shadow fading loss in dB (denoted by ψ_{dB}) is a Gaussian random variable in dB with zero mean and a standard deviation of $\sigma_{\psi dB}$ in dB:

$$\psi_{dB} \sim \mathcal{N}(0, \sigma_{\psi dB}) \quad (5)$$

The log-normal shadowing loss is added to the average path loss to derive the total dB loss at a location with distance d from the transmitter: $PL(d) = \overline{PL}(d) + \psi_{dB} \quad [dB]$ where $\overline{PL}(d)$ is the average path loss as a function of transmitter-receiver distance d based on the Hata model. Therefore the total dB loss is characterized by a Gaussian distribution with



(a) Decorrelation distance = 25 m (b) Decorrelation distance = 100 m

Fig. 2. Correlated shadow fading examples in a square area 300 m × 300 m, $\sigma_{\psi dB} = 8dB$, circle markers are SU nodes (white: normal, black: malicious)

mean $\overline{PL}(d)$ and standard deviation $\sigma_{\psi dB}$. The correlation between shadow fading at two locations separated by distance δ can be characterized by [31]:

$$A(\delta) = \sigma^2_{\psi dB} e^{-\delta/X_c} \tag{6}$$

where X_c is the decorrelation distance and is usually in the order of the size of the obstacles in the environment. For example for outdoor environments, the decorrelation distance is typically between 50 m and 100 m [31]. As it is inferred from Equation 6, closely located receivers (with smaller δ) experience highly correlated shadowing. This is intuitive because the two receivers that are close to each-other are likely affected by the same obstacles. Figure 2 shows heat-map plots of two examples of simulated correlated shadow fading with different decorrelation distances. For our Monte Carlo simulation (as we will describe in Section VI-A), we generate random shadow maps based on the above model.

IV. Distributed average consensus-based cooperative spectrum sensing

In an average consensus-based DCSS scheme, the SUs aim at estimating the average of the received power by all of the SUs. At each sensing round, each SU first measures its own received power as its initial value; then it participates in broadcast and update iterations. In each iteration, the SUs broadcast their current values and update their average estimates based on the received values from neighbors. Finally, each node independently compares its estimate of the average power with a threshold and makes its final decision about the PU presence. In this section, we briefly describe two categories of average-consensus algorithms: gossip-based and weighted linear iteration-based, that are used for DCSS application. We will compare our DCSS scheme against these schemes.

A. Weighted linear iteration-based

In the weighted linear iteration scheme, the nodes in the network follow a weighted linear combination update strategy at each iteration in order to converge to a consensus about the global average [6]. The value of node i at consensus iteration c is denoted by $v_i(c)$. At each sensing round, each node i is initialized with $v_i(0) =$ *received power at node* i. In order to converge to the global average, at each consensus iteration c, each node i updates its value with a weighted linear combination of its own value and the received values from its neighbors [6]:

$$v_i(c+1) = W_{ii} v_i(c) + \sum_{j \in N_i} W_{ij} v_j(c), \qquad i = 1, ..., n, \tag{7}$$

where W_{ij} is the weight for $v_j(c)$ at node i for neighbor j. If j is not a neighbor $W_{ij} = 0$. Also, W_{ii} is the self-weight of node i. If we consider the vector of the values at all nodes at iteration c, $v(c) = (v_1(c), ..., v_n(c))$ and the $n \times n$ matrix W consisting of all of the mutual weights, the above linear iteration can be written in vector form as follows: $v(c+1) = Wv(c)$. For asymptotic convergence to the global average, matrix W must satisfy the following [10]:

$$\lim_{c \to \infty} W^c = (\frac{1}{n}) \mathbf{1}\mathbf{1}^T. \tag{8}$$

Obviously, for the distributed linear iterations to asymptotically converge to the global average, the graph must be connected; otherwise, the convergence can only be reached for each isolated subgraph. Optimal and heuristic approaches have been proposed to realize the weight matrix that satisfies the convergence condition as described above. The optimal solution [10] is not a distributed solution and therefore is not practical for our purpose. Two heuristic weight choices that satisfy the convergence condition and therefore guarantee asymptotic convergence to the global average are [6]:

- Metropolis: Weights are calculated based on the larger number of neighbors in each pair of nodes:
  $W_{ij} = \frac{1}{1+max\{|N_i|,|N_j|\}}, \ j \in N_i$
- Maximum-degree: Weights are calculated based on the maximum degree (number of neighbors) in the network [13]:
  $W_{ij} = \frac{1}{1+\text{max degree}}, \ j \in N_i$

In both of the above schemes, $W_{ij} = 0$, if j is not a neighbor and the self-weight is set such that the sum of weights is 1: $W_{ii} = 1 - \sum_{j \in N_i} W_{ij}$.

B. Gossip-based

We also compare our proposed scheme against DCSS schemes based on Push-Sum protocol [7] which is a gossip-based solution for the average consensus problem. In Push-Sum algorithm, each node maintains a sum, which is initialized

to be the received PU power at this node; it also maintains a gossip weight which is initialized to 1. At each consensus iteration, each node sends a fraction of its sum and weight to one or more randomly chosen neighbor(s). We will compare our proposed DCSS scheme against the following two variants of the Push-Sum scheme: 1) One-neighbor gossip, where at each iteration, each node picks one of its neighbors at random and sends half of its sum and weight to it [12]. 2) Flooding gossip, where at each iteration, each node distributes its sum and weight values uniformly among all of its neighbors [26]. See [7] for details of the Push-Sum algorithm.

C. Neighbor discovery overhead

The existing average consensus-based DCSS schemes that are described above impose overhead related to neighbor discovery at each sensing round. In the Metropolis linear iteration-based scheme, the weights are calculated based on the larger degree in each pair of nodes. Therefore, the nodes must first discover their neighborhood sizes (degrees) and then broadcast their degrees to others. As a result, this scheme requires the nodes to perform neighbor discovery that needs to be updated every sensing round which imposes significant overhead. In addition to that overhead, each node also needs to broadcast its degree to the other nodes at each sensing round. Note that in a mobile network the neighborhoods are changing all the time and therefore the number of neighbors of a node is different from one sensing round to the next. As a result using the perceived number of neighbors based on the broadcasts received in the immediately previous sensing round introduces error in convergence. Similarly, for the maximum-degree variant, determining the maximum degree is not trivial in a distributed ad hoc network where nodes only have local views of the network.

In the gossip-based DCSS scheme, each node needs to know the number of its active neighbors in advance to calculate the fraction to broadcast in the current sensing round (or to pick one random neighbor in the case of one-neighbor gossip). As mentioned above, using the perceived number of neighbors based on the previous round introduces error (leakage of some fractions of values in this case). Therefore, a neighbor discovery phase is necessary at each sensing round.

Neighbor discovery in mobile ad hoc networks is a non-trivial task and an active area of research. The determination of the direct neighboring nodes is generally done using hello protocols where each node periodically broadcasts a hello message. Each node considers another node as a direct one-hop neighbor only if it receives at least one hello message from it [33] [34]. The random access discovery schemes require the nodes to be randomly in a "listen" or "transmit" mode in each time slot so that each node receives the hello message from every neighbor at least once in a predefined time period. These algorithms generally require a large number of time slots to reliably discover all neighbors [35] [36]. Therefore, neighbor discovery imposes a significant time overhead in particular for mobile networks with changing topologies.

The communication overhead of neighbor-discovery is similar to the cost of one consensus iteration. If the neighbor discovery is needed at each consensus iteration, the overhead is 100%. In the case where neighbor discovery is performed only once per sensing round, the overhead is less than 100% but still considerably high because generally the number of consensus iterations per sensing round is small (e.g. 4) to limit the cost. As we will show next, our proposed equally-weighted DCSS scheme is considerably more efficient than the existing schemes because it does not require the neighbor discovery phase and thus completely eliminates the associated overhead.

V. Proposed equally-weighted linear iteration-based DCSS

We introduce a novel DCSS scheme based on iterative linear combinations with equal weight assignment. At each iteration, each node simply broadcasts its value and then updates its value as an equally-weighted average of its own value and all of the received values in this iteration as follows:

$$v_i(c+1) = \frac{v_i(c) + \sum_{j \in R_i} v_j(c)}{1 + |R_i|},\ i = 1, ..., n, \qquad (9)$$

where R_i denotes the set of nodes from which node i received a value in this iteration. Thus, the nodes do not need to know the number of their active neighbors in advance; instead they only listen and count the number of received messages and average them to update their current estimates. As a result, there is no overhead associated with any sort of neighbor discovery. As discussed in Section IV-C, the neighbor discovery overhead of the existing DCSS schemes makes them costly and unattractive. Our proposed equally-wighted approach offers significantly lower overhead compared with the existing schemes due to the elimination of neighbor discovery. Note that if every neighbor broadcasts its value to node i, then R_i will be essentially equal to N_i (the neighbor set), therefore translating the proposed scheme back to Equation (7), the equal weights that node i assigns to any neighbor j and to itself will be equal to $\frac{1}{1+|N_i|}$ and a weight of zero is assigned to the other nodes.

While the average consensus algorithms are originally designed to asymptoticly converge to the exact global average for sufficient number of iterations (e.g. in sensor fusion applications), in DCSS applications the nodes do not need to converge to the exact average as the estimated average is solely used for comparison against a detection threshold. Therefore, the accuracy of estimation can be relaxed. Obviously with more consensus iterations the accuracy of the estimated average improves, however the consensus overhead also increases. Thus, as for the cost-performance trade-off, the number of iterations must be as few as possible for the desired performance.

The corresponding weight matrix of this approximate approach does not necessarily satisfy the condition for asymptotic convergence to the exact global average as described in Equation (8); however, we show with Monte Carlo experiments, that this scheme results in an approximate convergence with a small error offset and it converges faster compared to the Metropolis and maximum-degree heuristics (See Figure 3). In addition, we will show in the next sections that the small convergence error does not degrade the performance of DCSS. This is because in the DCSS applications, an exact

TABLE I
SIMULATION PARAMETERS

Path Loss and Shadow Fading		Random Way Point Mobility Model		Noise and Threshold		Monte Carlo Simulation	
PU Distance from CRAHN	15 km	CRAHN Area	300 m×300 m	Noise Figure	11 dB	# SU Nodes	50
PU Antenna Height	30 m	Min Velocity	1 m/s	Channel Bandwidth	6 MHz	SU Node Range	80 m
SU Antenna Height	1 m	Max Velocity	2 m/s	Noise Power	-95.22 dBm	Simulation Time	8000 s
Center Frequency	615 MHz	Min Pause	60 s	Threshold Range	[-96 , -82] dBm	Sense Interval	2 s
Log-normal Shadowing Standard Deviation	{4, 8, 12} dB	Max Pause	120 s				
Transmit Power (P_t)	54 dBm						

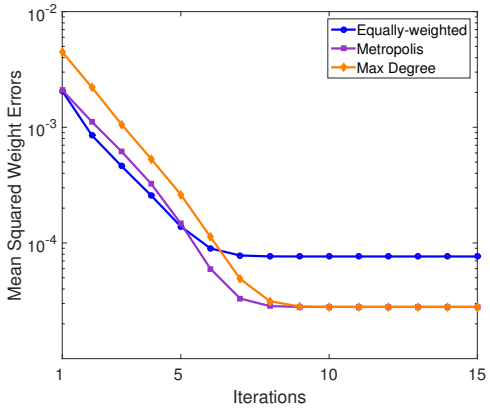


Fig. 3. Weight convergence in different schemes

convergence is not necessary; instead, a practical solution is desired where the nodes estimate the average power within only a few iterations to quickly arrive at binary decisions about the PU presence.

Since for asymptotic convergence weights must satisfy Equation (8), we define the $n \times n$ weight error matrix Υ as a metric to evaluate the convergence for any number of iterations, c:

$$\Upsilon = W^c - (\frac{1}{n})\mathbf{1}\mathbf{1}^T \tag{10}$$

We evaluate the convergence of W^c in different schemes through Monte Carlo simulations. In each Monte Carlo run, we consider a different random network topology which corresponds to a different weight matrix W for each of the schemes. For each of these different weight matrices, we derive Υ and then compute the mean of the square of its elements: $\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\Upsilon_{ij}^2$. Then we average that over the many random network topologies in the Monte Carlo simulations. Figure 3 compares the convergence of the three different schemes in terms of weight error convergence. All of the random topologies include a graph of 50 nodes in a 300 m $\times$ 300 m area. All of the three schemes almost converge within around 6 iterations.

As expected, for the proposed equally-weighted scheme, there is an associated error offset in the convergence; however, the weight errors drop faster than the other two schemes. As a result, the convergence errors of the equally-weighted scheme are smaller in the first few iterations. This faster convergence of weights in the equally-weighted scheme directly translates to a faster convergence of nodes' values towards the global average. Note that fast convergence in a few iterations is vital for a practical DCSS scheme whereas higher number of iterations might not be affordable anyway. In Section VI, we present our performance results in terms of PU detection error rates which confirm that our proposed equal-weighting scheme performs as well as other more complex schemes. In addition, as discussed above, it is the most practical choice for DCSS due to its simplicity.

VI. PERFORMANCE ANALYSIS OF CONSENSUS-BASED DCSS SCHEMES

A. Simulation Setup

We study a cognitive radio ad hoc network with 50 SU mobile nodes spread and moving in a 300 $m \times 300$ m square, according to a random way point model [37], operating in a TV whitespace channel of 6 MHz bandwidth with 615 MHz center frequency. SU network is at a 15 km distance from the PU transmitter (TV station). We assume a 54 dBm transmit power for PU transmitter. Using the Hata path loss model, the nominal loss only due to distance is about 138 dB. In addition to path loss, we consider log-normal shadow fading with dB spread of $4, 8, 12$ dB. We analyze the performance of different schemes in different scenarios through Monte Carlo simulations, where at each run, the network is randomly initialized. P_{FA} and P_{MD} of the network are derived as the average fraction of the honest nodes in the network that make a false alarm and missed-detection error in a sensing round, respectively. The simulation parameters listed in Table I, will be used for the experiments presented in the rest of the paper.

B. Comparison results

In this section, we evaluate and compare the performance of the distributed consensus schemes that were described earlier using complementary Receiver Operating Characteristics (ROC) curves plotting missed-detection rate versus false alarm rate for various values of detection threshold. We have picked a wide range of detection thresholds ranging from -96 dBm to -82 dBm that result in very high to very low missed-detection and false alarm error rates. Figure 4 shows ROC curves for all of the schemes that were described in the previous sections, for 8 consensus iterations. The results show that the proposed equally-weighted linear scheme performs as well as the other consensus more complex schemes.

As discussed in Section V, the estimated average of received power at each SU node is used merely for a binary decision about the PU activity; therefore, the accuracy of the average is of less significance. Our simulation results show that for the DCSS application, the weight error does not degrade the performance for our target number of iterations which must be small (e.g. 8 iterations). In addition, as discussed before, the

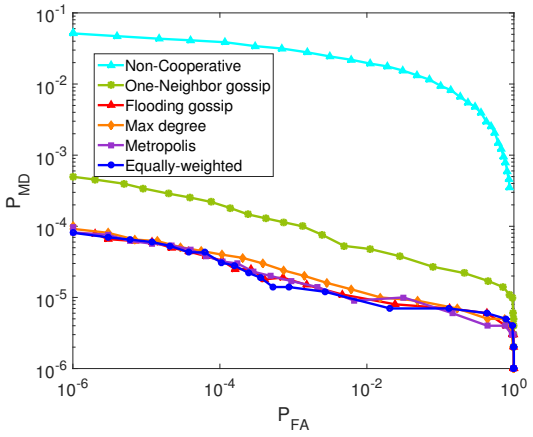


Fig. 4. ROC for consensus-based DCSS schemes. Number of consensus iterations $= 8$, $\sigma_{\psi dB} = 8\ dB$

equally-weighted linear scheme also offers the lowest overhead among all as no information about the network topology and neighborhoods is needed. As a result, the proposed equally-weighted scheme is the most practical choice among all of these schemes. We will further analyze this scheme in the rest of the paper.

C. Performance-complexity trade-offs on the number of consensus iterations

Cooperative spectrum sensing is deployed to overcome the correlated shadow fading by exploiting the spatial diversity among the cooperating nodes with the hope that different nodes at various locations experience different shadow severity. Therefore, the nodes that enjoy a better reception can help the other nodes who may suffer from a deep shadow. As discussed in Section III, the decorrelation distance associated with an environment determines the size of the shadows (see Figure 2 for example). When decorrelation distance is large (shadows are large), in order to better exploit the existing spatial diversity, nodes must cooperate within larger areas (i.e. with nodes that are multiple hops away.) For example, if SUs consult with their direct neighbors only (i.e. only 1 iteration), the cooperation will be ineffective. The reason is that the neighboring nodes are under the effect of the same shadow and their sensing data is highly correlated. As a result, a "local averaging" scheme is not effective particularly for scenarios with large decorrelation distances.

On the other hand, the communication and computational overhead of the consensus-based DCSS schemes is directly related to the number of consensus iterations. If C denotes the number of consensus iterations, the communication overhead of consensus for each node will be C packets per sensing round. In addition, if we denote the average number of neighbors of a node at any given time by B, the computational overhead is of the order of $O(C \times B)$. Therefore, in order to keep the consensus overhead affordable for DCSS application, the number of iterations must be as small as possible. In a nutshell, there is an important complexity-performance trade-off in determining the optimal number of iterations.

Figure 5 compares the missed-detection rates of the equally-weighted scheme with only 1 consensus iteration versus the same scheme with 4 and 8 iterations. On the horizontal axis the decorrelation distance is increased from 25 m up to 100 m. For large decorrelation distances, in particular, a local cooperation (# Iterations = 1) is not sufficient; higher number of iterations is required to better use the spatial diversity. The gap is even more significant for the case of higher dB spread as seen in Figure 5(c) with $\sigma_{\psi dB} = 12$ dB. Figure 6 shows ROC plots for our proposed equally-weighted consensus-based DCSS scheme with 1, 4, and 8 iterations. With only 1 consensus iteration, each node receives information solely from direct neighbors. 4 iterations is significantly better than 1 iteration, however the performance resulting from 8 iterations is very close to 4 iterations. For the rest of the paper we fix the number of consensus iterations to 4.

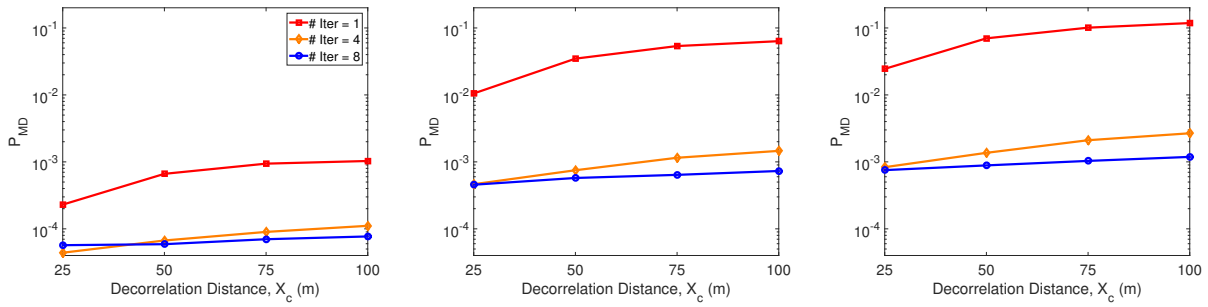
VII. Insistent Spectrum Sensing Data Falsification

In iterative average consensus-based DCSS schemes, in all of the iterations, all of the nodes must follow a predefined update strategy. We study a destructive type of Spectrum Sensing Data Falsification (SSDF) attack [9] aiming at these iterative schemes. This attack is different from the conventional SSDF attack in that, here the attacker not only falsifies its initial sensed value, it also disregards the received values from the other nodes and it never updates its estimate. Thus the attacker broadcasts the same falsified data (possibly with some added noise) in every iteration of the consensus. We call this attack *Insistent* SSDF or ISSDF. Since the falsified data is repeatedly fed into the consensus process, an ISSDF attack is significantly more destructive than the conventional SSDF. We will show that ISSDF attacks make the honest (non-malicious) nodes, diverge from the correct average. In the case of SSDF attack, if the number of attackers is sufficiently small, the malicious effect may be neutralized by the honest nodes in the network by only using simple cooperation. In contrast, as we will show in our experiments, even a very small set of ISSDF attackers can have much larger impact which makes trust management a necessity.

The existing SSDF mitigation schemes are mainly adapted for the non-iterative or centralized cooperative spectrum sensing schemes. While the negative effect of SSDF attacks on PU detection performance is well-known, the effect of ISSDF attack in the context of iteration-based distributed cooperative sensing schemes has not been sufficiently studied. Our proposed trust management technique which will be introduced in the next sections is designed to mitigate both SSDF and ISSDF attacks, however, since the effect of ISSDF attack is much more severe, we focus on this worst case scenario.

In this paper, we study *fabricating* ISSDF attack where the attacker broadcasts a high constant value, when the PU is absent and a low constant value when the PU is present in all of the consensus iterations. Therefore, in our attack model, the attacker falsifies its sensing data as follows in both presence and absence of the PU:

1) If PU is present, a fabricating ISSDF attacker constantly broadcasts a value of zero in all consensus iterations with the



(a) $\sigma_{\psi dB} = 4\ dB$ (b) $\sigma_{\psi dB} = 8\ dB$ (c) $\sigma_{\psi dB} = 12\ dB$

Fig. 5. The proposed equally-weighted DCSS scheme in the presence of correlated shadow fading with different dB-spreads. Detection threshold = -90 dBm.

goal of dropping the honest nodes' estimates of global average below the detection threshold: $v_{Attacker}(c) = 0, \quad c = 1, .., I$. Therefore, the attacker tries to deceive the honest nodes into interfering with the PU by causing missed-detection in the network. This attack can seriously disrupt the spectrum sharing as it directly conflicts with the fundamental requirement of non-interference in a cognitive radio network.

2) If PU is absent, a fabricating ISSDF attacker repeatedly broadcasts a high constant in all of the consensus iterations so that the honest nodes' estimate of the global average is raised above the detection threshold: $v_{Attacker}(c) = D, \quad c = 1, .., I$, where D is a positive high constant. Therefore, the attacker tries to mislead the honest nodes to decide that the PU is present so that the misguided nodes back-off and leave the channel. Thus, one potential motivation behind this attack is selfishness: By increasing the false alarms in the network, the attacker aims at eliminating some of its competitors for using the free channel. This attack can leave the CRAHN completely inoperable as the honest nodes may not find any opportunity to use the free channel.

Note that, in order to prevent detection only based on being constant during all times, an ISSDF attacker may add a random noise to either low or high malicious values that it broadcasts. Our proposed trust scheme, introduced in the next section, is able to mitigate these ISSDF attackers as well. In our attack model, we assume a collusive set of fabricating ISSDF attackers who coordinate to broadcast either low or high values to strengthen the effect of each-other.

Figure 7 shows the effect of a collusive set of 10 ISSDF attackers (out of 50 nodes in total, that is 20% malicious) on our proposed equally-weighted consensus-based DCSS for two example scenarios where PU is present or absent. In both examples, the values of the the 40 honest nodes during consensus iterations of a single sensing round is shown in box plots. Note that the left-most box of each sub-figure corresponds to the initial values of all of the honest nodes, therefore the starting point of both scenarios (honest and under attack) is the same. The true average of the initial values of the honest nodes is shown with a horizontal line for reference. While, in an honest network, the values of the nodes converge to the true average (with a small error offset, as explained before), Figure 7(a) shows that the ISSDF fabricating attackers successfully make the honest nodes' values diverge from the correct average and cause their values to approach zero,

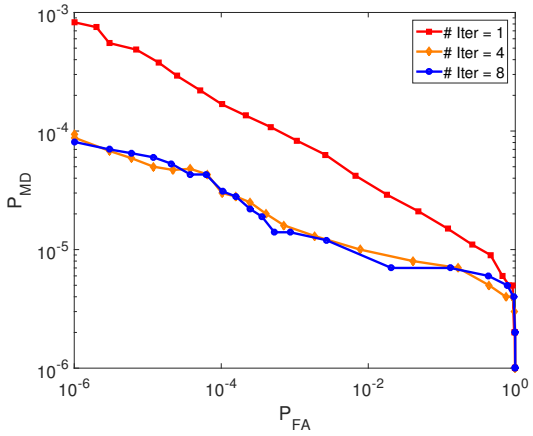


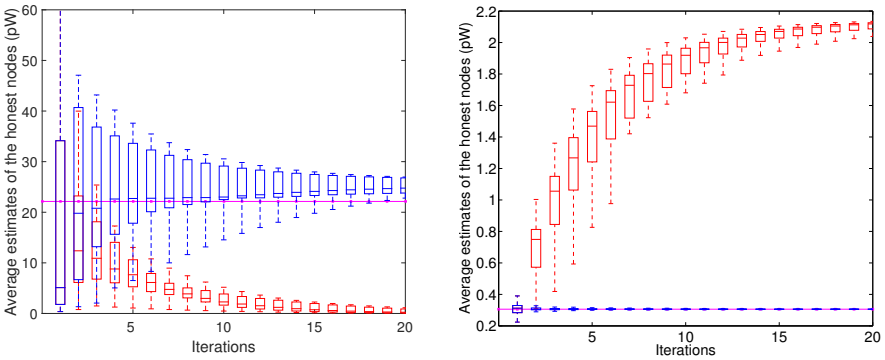
Fig. 6. ROC for the proposed equally-weighted DCSS with different number of consensus iterations, $\sigma_{\psi dB} = 8\ dB$

potentially causing missed-detection. On the other hand, when the PU is absent (Figure 7(b)), the attackers raise the estimated averages of all of the honest nodes significantly higher than the true average, potentially causing false alarms. Note that the effect of ISSDF attackers is amplified with each iteration as the attackers keep broadcasting incorrect values and their falsified data is diffused further throughout the network.

VIII. Proposed Trust Management Scheme

Our trust management works based on trust scores that the nodes assign to each other based on their previous interactions. The trust score that node i assigns to node j at time step (sensing round) t is a value in the interval $[0, 1]$ and is denoted by $\theta_{ij}(t)$. This score can be interpreted as the estimated probability of j being honest from the viewpoint of node i. In order to make the DCSS schemes resilient to data falsifying attacks, each node must be aware of the level of trustworthiness of its neighbors before relying on the received values from them. As we will explain in the next paragraphs, the trust scores are used as weights associated with reports received from different neighbors.

Using our proposed trust system, the nodes do not simply accept neighbors' reports; instead they gradually determine the level of trust of their neighbors through interaction observations. The trust score calculation method that we have devised in this work is inspired by the Beta Reputation System which is previously used in [9] and [20]. When node i compares its



(a) PU is present (b) PU is absent

Fig. 7. Examples showing the impact of collusive fabricating ISSDF attacks in causing divergence from the correct average in one sensing round. The values of the honest nodes are shown with box plots. Blue boxes: nodes in honest network; Red boxes: 20% of nodes are ISSDF malicious. The horizontal solid line shows the correct global average.

final estimate of the average received power with the threshold and makes a final binary decision about the presence of the PU, it also verifies whether the initial values received from its neighbors are consistent with this decision. Therefore, at each sensing round t, the following observation is made by node i from each of its neighbors j:

$$o_{ij}(t) = \begin{cases} 1 & if \quad g_{ij}(t) > \gamma \text{ AND } f_i(t) = \text{PU present} \\ & \text{OR} \quad g_{ij}(t) < \gamma \text{ AND } f_i(t) = \text{PU absent} \\ 0 & \text{Otherwise} \end{cases} \tag{11}$$

where $g_{ij}(t)$ denotes the initial value that node i received from neighbor j in the first consensus iteration of sensing round t and $f_i(t)$ is node i's final decision in this round. If the broadcasted value from j is in agreement with i's final decision, the binary observation is 1 (a positive observation) and otherwise it is 0 (a negative observation). In our ISSDF attack model, the same falsified value is repeated in every iteration, thus, it is sufficient to only observe the received values in the very first iteration of the consensus. It is noteworthy that as SU nodes do not know the "ground truth" about the activity of the PU, the best strategy for them is to rely on their own final decisions when evaluating neighbors' trustworthiness.

We propose that each node i maintains a sliding binary observation vector, O_{ij}, for each neighbor j. The observation vector can grow up to a maximum length, after which the newest observations replace the oldest ones. The trust score can be calculated as follows:

$$\theta_{ij}(t) = \frac{H(O_{ij})}{|O_{ij}|} \tag{12}$$

where $H(.)$ denotes the Hamming weight of the binary vector O_{ij} and $|O_{ij}|$ is the length of it. As described in the previous sections, at each sensing round, a number of consensus iterations (e.g. 4) are needed to finalize the decisions. Note that, at each sensing round, the trust scores are updated only when the final decisions are made (after the final consensus iteration) and not in between the consensus iterations. From one sensing round to the next, the nodes move randomly. As a result, the nodes might have different set of neighbors at each sensing round. When nodes i and j are neighbors (within the range of each other) in a sensing round, they record their observations from each other in their observation vectors. If they move farther from each other, they might not be neighbors any more in the next sensing round, however, the nodes do not drop the existing observations from each other as it is possible that they become neighbors again in the future sensing rounds, when they can use these observations. (Note that we assume that within the iterations, the nodes are static as the consensus is reached quickly relative to the nodes' movement.)

A. Analysis on node agreement probability

The trust score is essentially a quantization of the probability of agreement between the two nodes in the recent interactions. In this section, we analyze the trust score that an honest node assigns to a fabricating ISSDF attacker. A fabricating attacker always reports the opposite of the truth about the PU activity. Therefore, when an honest node i makes an observation from a fabricating node a, node i is able to detect the conflict and tag the observations as negative. There are two conditions where the two nodes agree: 1) if H_0, then a's report indicates the PU is present; therefore, if i makes a false-alarm error, the two node agree, 2) if H_1, then a's report indicates the PU is absent; thus, i agrees with a in case of a missed-detection error. Equation (13) shows the agreement rate which is directly translated to the trust score that a typical honest node assigns to a fabricating attacker.

$$\begin{aligned} Pr(agree_{i,a}) &= Pr(agree_{i,a}|H_0)Pr(H_0) \\ &+ Pr(agree_{i,a}|H_1)Pr(H_1) \\ &= Pr(F_i)Pr(H_0) + Pr(M_i)Pr(H_1) \end{aligned} \tag{13}$$

where F_i and M_i are the events of honest node i making a false alarm and missed-detection error, respectively. In the conditions where the false alarm and missed-detection rate of the honest nodes are not too high, the agreement rate with the fabricating attacker will be sufficiently small as well, thus the assigned trust scores will be small as desired. When honest nodes make too many honest mistakes (even with cooperation), they may increase the associated trust scores. This type of trust error in these extreme conditions is clearly inevitable. As we will show in the results, the integration of trust management

with the proposed DCSS significantly improves the error rate performance in the presence of fabricating attackers.

B. Trust integration

We incorporate the trust management into the linear iterations of our proposed equally-weighted DCSS scheme by using the trust scores as weights associated with received values from different nodes:

$$v_i(c+1) = \theta_{ii}(t)v_i(c) + \frac{\sum_{j\in R_i}\theta_{ij}(t)v_j(c)}{1+|R_i|},\ i = 1, ..., n \tag{14}$$

where R_i denotes the set of nodes from which node i received a value in this iteration and $\theta_{ii}(t) = 1 - \frac{\sum_{j\in R_i}\theta_{ij}(t)}{1+|R_i|}$. The integration of trust scores as weights into our proposed equally-weighted linear iteration-based consensus scheme, makes the weighting biased so that the values from more trustworthy neighbors are more effective than the others. Moreover, our proposed trust system does not introduce any communication overhead. Denoting the number of consensus iterations by C and average number of neighbors by B, the computational overhead of incorporating the trust scores is on the order of $O(C \times B)$. This overhead is reasonably low for realistic scenarios with a bounded number of consensus iterations (e.g. 4 iterations) and typical neighborhood sizes (e.g. 8 to 10 neighbors).

C. Discussion on trust initialization strategy

Our proposed trust assignment strategy is conservative which means each node must perform a minimum number of observations (O_{min}) from a neighbor before it assigns a non-zero trust score to it (i.e. $\theta_{ij} = 0$ if $|O_{ij}| < O_{min}$). As a result, a node builds up a sufficient record of observations from a new neighbor before considering the neighbor's sensing reports in its decisions. This strategy necessitates an initial warm-up period during which the nodes only trust their own sensing values for making decisions, while they observe the values received from their neighbors and also update their trust scores. This conservative strategy does not take any risk in accepting values from unknown nodes and therefore yields better results than a strategy where the nodes start with high trust scores for the other nodes. We observed in our simulation results that the ISSDF attackers, if trusted, can influence their neighbors severely and can cause errors in neighbors' final decisions that result in higher trust in ISSDF attackers and higher impact of the network. The reason is that the malicious value of an ISSDF attacker is broadcasted repeatedly in every consensus iteration, which amplifies its effect and therefore even a few ISSDF attackers in comparison to a majority of honest nodes can cause errors. As a result, we choose the more conservative strategy for trust assignment.

D. Mitigating dynamic attackers

In this section, we consider a more complex attack scenario where a subset of the honest nodes become malicious while the network is in operation. The main complication of this dynamic behavior is that a node which has been honest and therefore has already built up high trust in the viewpoint of the other honest nodes, suddenly starts to broadcast falsified data. This type of attack is harder to mitigate because the dynamic attackers abuse their initial high trust score to influence the final decision of the honest nodes to be in agreement with them which in turn makes the honest nodes continue to trust the dynamic attackers.

In order to mitigate the dynamic attacks, we introduce an outlier detection technique as another layer of defense in our proposed trust management scheme: At the first consensus iteration, $c = 0$, of a sensing round, each node i receives a set of values: $v_j(0), j \in R_i$, where R_i denotes the set of nodes from which node i received a value. Node i identifies both the largest and the smallest values among all of the received values and tags the corresponding nodes as the "high outlier" and the "low outlier", respectively for that sensing round. The received values from both of the outlier nodes are excluded from the updates of node i in the following iterations of that sensing round. As a result, the falsified values received from a currently trusted dynamic attacker is filtered out. Note that, the trust score update is performed without any change as before; therefore, the trust score of a dynamic attacker will be decreased gradually during the following sensing rounds as its reports are repeatedly in conflict with the honest nodes.

At each point of time, an honest node i may have several malicious neighbors that broadcast falsified values. Assume node i has already recognized its malicious neighbors and has assigned low trust scores to them. At this time, one of node i's honest neighbors who has already built up high trust score, starts to broadcast falsified values. The goal of the outlier detection technique is to detect and exclude the dynamic attacker (with currently high trust score) rather than the other already-recognized attackers because the other attackers are already mitigated with low trust scores.

We propose the following strategy for the high outlier detection: Node i needs to identify the new dynamic attacker among all of its malicious neighbors that broadcast falsified high values when PU is absent. In order to do that, i weights all of the values received from its neighbors by their corresponding trust scores. We denote the weighted value from node j by ω_j. With this strategy, the dynamic attacker is likely to have the largest ω_j among all since both its falsified value and its trust score are high. Thus, the dynamic attacker can be correctly identified and excluded.

On the other hand, for the low outlier detection, the new dynamic attacker should be identified among all of the attackers in the neighborhood that broadcast low values when PU is present. In this case, weighting the received values by the corresponding trust scores is not helpful. The reason is that for the already-recognized attackers, weighting their values by their current low trust scores further decreases their weighted values. In addition, for the new dynamic attacker, weighting its value by its current high trust score relatively increases its weighted value compared to the already-recognized attackers. Therefore, by trust-weighting, the smallest weighted value (low outlier) most likely will not be the new dynamic attacker. As a result we propose that for low outlier detection, the values should not be weighted by trust score (or equivalently, they

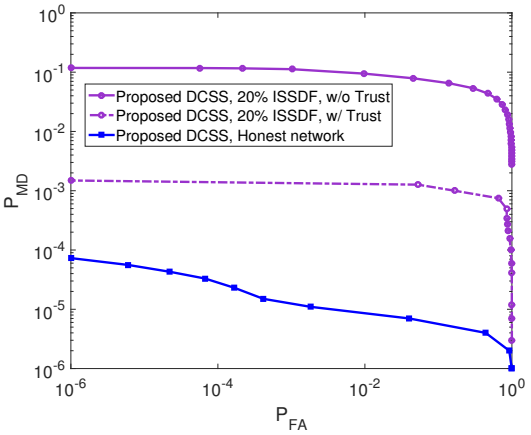


Fig. 8. ISSDF attack and mitigation with trust management for our proposed trust-aware DCSS scheme

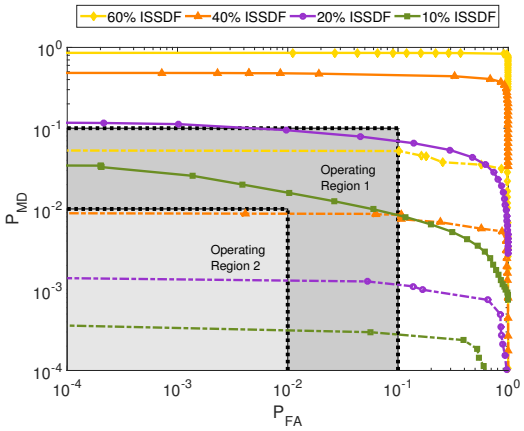


Fig. 9. ISSDF attacks with different percentage of malicious ISSDF nodes with and without trust management, Solid lines: w/o trust, Dashed lines: w/ trust

are weighted by 1).

Therefore, we propose the trust-aided outlier detection strategy as follows: At the first iteration of sensing round t, each node i identifies among the received values the high outlier, j_h^*, as the largest *weighted* received value and identifies the low outlier, j_l^*, as the smallest *unweighted* received value:

$$\begin{aligned} &\text{High outlier detection:} \\ &j_h^* = \arg\max_{j \in R_i} \omega_j, \qquad \omega_j = \theta_{ij}(t) v_j(0) \\ &\text{Low outlier detection:} \\ &j_l^* = \arg\min_{j \in R_i} v_j(0) \end{aligned} \tag{15}$$

Obviously, in both cases of high and low outlier detection, it is possible that the outlier is a correct (non-malicious) value; nevertheless, excluding the outliers significantly reduces the risk of dynamic attacks. In addition, as will be shown with simulations, there is essentially no negative effect due to the exclusion of potentially correct outlier values since a correct final decision can be achieved exploiting the cooperation with the non-outliers.

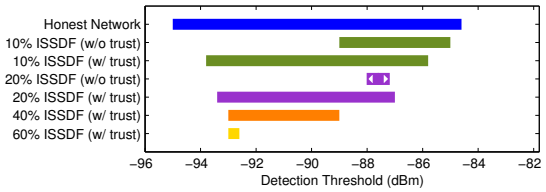
When combined with our proposed trust scheme, the proposed trust-aided outlier detection technique can effectively neutralize dynamic attacks. Unlike the existing high-overhead outlier detection schemes where the nodes need to compute a deviation threshold at each iteration [23], our scheme does not require additional computations by the nodes at each iteration and is only based on comparison among the received values. Note that it is unlikely that while the network is in operation, several neighbors of a node turn malicious at the same time in a non-collusive scenario and therefore our proposed scheme is able to detect and exclude the dynamic attacker. The study of collusive dynamic attacks is a subject for future research.

E. Simulation results

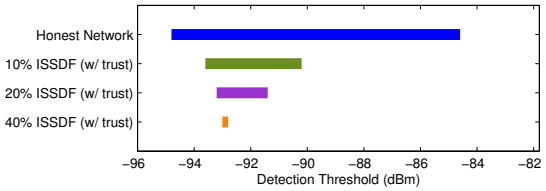
Figure 8 shows the ROC curves for the scenario where 20% of the nodes are ISSDF with and without trust enabled. The ROC curve for an honest network (no attackers) is also shown for comparison. For a fixed false alarm rate, enabling trust management improves the missed-detection rate by as much as two orders of magnitude. From right to left of the plots, the detection threshold is increased. As a result intuitively, the false alarm rate is reduced and the missed-detection rate is increased. As it is clear from the figure, using our proposed trust scheme improves both P_{MD} and P_{FA} significantly. For the simulations, we use $O_{min} = 10$ and we have excluded the initial warm-up sensing rounds in the reported results. We analyze various levels of attack severity and the trust improvements in those scenarios in Figure 9. Our proposed trust-aware scheme outperforms the schemes that are not trust-enabled in all of the scenarios including the case where the majority of the SUs are malicious (See the ROC curve corresponding to 60% ISSDF).

1) Operating regions and sensitivity requirements: The ROC curves are perfect tools for determining the system requirements for a desired operating region in terms of missed-detection and false alarm rates. We have shown two examples of desired operating regions overlaid on top of the plots in Figure 9. The operating region 2 corresponds to both missed-detection and false alarm rates smaller than 10^{-2} and the more relaxed region 1 corresponds to the rates smaller than 10^{-1}. Figure 10 shows the detection threshold subranges in [-96 dBm, -82 dBm] that satisfy the error rate requirements of the operating regions 1 and 2 shown in Figure 9 for different malicious scenarios and also for the honest case.

As can be seen from Figure 10, without trust management, for most of the scenarios, the desired error rates cannot be realized. In contrast, when our proposed trust system is enabled, all of the scenarios except for 60% ISSDF can satisfy the requirements for both regions 1 and 2 for some threshold subrange. Furthermore, the trust management increases the dynamic range of the detection thresholds that support the target performance; therefore, a wider range of radio devices with diverse sensitivity thresholds can be supported while maintaining the desired performance. As a result, using the proposed trust scheme enables us to relax the sensitivity requirements on the cognitive radio devices and potentially reduce the cost. The presented results confirm the significance of integration of the proposed trust system into DCSS.



(a) Operating Region 1: $P_{MD} < 10^{-1}$ AND $P_{FA} < 10^{-1}$ Note: Without trust, in 40% and 60% scenarios, the error rate of 10^{-1} cannot be realized, regardless of the threshold.



(b) Operating Region 2: $P_{MD} < 10^{-2}$ AND $P_{FA} < 10^{-2}$ Note: Without trust, none of the scenarios can achieve the error rate of 10^{-2}, regardless of the threshold.

Fig. 10. Range of detection thresholds to realize the two desired operating regions shown in Figure 9 under different scenarios.

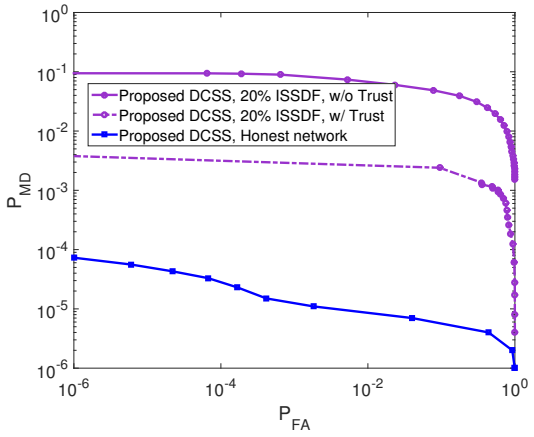


Fig. 11. Dynamic ISSDF attack and mitigation with trust management

2) Dynamic attackers: We analyze the performance of our proposed trust scheme with trust-aided outlier detection in mitigating dynamic attackers. We assume that at the beginning of each Monte Carlo simulation 10% of the nodes are malicious ISSDF attackers and during the time of the simulation another 10% of the nodes become malicious. Therefore, at the end of a Monte Carlo simulation, in total 20% of the nodes are malicious. Figure 11 shows that our trust scheme can effectively mitigate dynamic ISSDF attackers who become malicious when the network is in operation.

IX. Conclusion

In this paper we present a novel trust-aware consensus-inspired scheme for distributed cooperative spectrum sensing that is robust against malicious Insistent Spectrum Sensing Data Falsification (ISSDF) attacks. The proposed equally-weighted linear iteration-based scheme is a practical method for ad hoc networks because it does not require the nodes to have any topology knowledge. We compare the performance of the proposed scheme against other more complex consensus-based methods and show that despite the simplicity, the performance enhancement through cooperation is as effective as the other schemes. We evaluate our proposed trust management scheme in the presence of collusive fabricating ISSDF attackers with various severity levels through extensive Monte Carlo simulations. We show that integration of our trust management with the proposed equally-weighted consensus-based scheme improves the performance in terms of missed-detection and false alarm error rates by as much as two orders of magnitude. Furthermore, we present an analysis of the operating characteristic curves and the desired operating regions and we show that adopting the proposed trust scheme increases the dynamic range of the supported sensitivity thresholds of the cognitive radio devices and therefore can reduce the cost and enhance the flexibility of the cooperative system.

Acknowledgment

This work was supported in part by the US National Science Foundation under grants ECCS-1408370, CNS-1265332, and ECCS-1232274. The authors gratefully acknowledge support from the US-Ireland R&D Partnership USI033 'WiFiLoc8' grant involving Rice University (USA), University College Dublin (Ireland) and Queens University Belfast (N. Ireland).

References

[1] S. M. Mishra, A. Sahai, and R. W. Brodersen, "Cooperative sensing among cognitive radios," in *IEEE International Conference on Communications (ICC)*, vol. 4, 2006, pp. 1658–1663.

[2] Q. Wu, G. Ding, Y. Xu, S. Feng, Z. Du, J. Wang, and K. Long, "Cognitive internet of things: A new paradigm beyond connection," *Internet of Things Journal, IEEE*, vol. 1, no. 2, pp. 129–143, 2014.

[3] H. ElSawy and E. Hossain, "Two-tier hetnets with cognitive femtocells: Downlink performance modeling and analysis in a multichannel environment," *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 649–663, March 2014.

[4] D. Cabric, S. Mishra, and R. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *The Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, vol. 1, Nov 2004, pp. 772–776 Vol.1.

[5] "Cognitive wireless RAN medium access control (MAC) and physical layer (PHY) specifications: Policies and procedures for operation in the TV bands," IEEE Standard 802.22, 2011.

[6] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*. IEEE Press, 2005.

[7] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *44th Annual IEEE Symposium on Foundations of Computer Science Proceedings*, Oct 2003, pp. 482–491.

[8] 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.

[9] R. Chen, J.-M. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *INFOCOM. The 27th Conference on Computer Communications*, April 2008, pp. 31–35.

[10] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *42nd IEEE Conference on Decision and Control*, vol. 5, Dec 2003, pp. 4997–5002.

[11] N. Ahmed, D. Hadaller, and S. Keshav, "Guess: Gossiping updates for efficient spectrum sensing," in *Proceedings of the 1st International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking*. New York, NY, USA: ACM, 2006, pp. 12–17.

[12] A. Vosoughi, J. Cavallaro, and A. Marshall, "A cooperative spectrum sensing scheme for cognitive radio ad hoc networks based on gossip and trust," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec 2014, pp. 1175–1179.

[13] Z. Li, F. R. Yu, and M. Huang, "A distributed consensus-based cooperative spectrum-sensing scheme in cognitive radios," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 383–393, 2010.

[14] W. Zhang, Y. Guo, H. Liu, Y. Chen, Z. Wang, and J. Mitola, "Distributed consensus-based weight design for cooperative spectrum sensing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 54–64, Jan 2015.

[15] S. Sundaram and C. Hadjicostis, "Distributed function calculation via linear iterations in the presence of malicious agents - part I: Attacking the network," in *American Control Conference*, June 2008, pp. 1350–1355.

[16] H. Zhang and S. Sundaram, "Robustness of complex networks with implications for consensus and contagion," in *IEEE 51st Annual Conference on Decision and Control (CDC)*, Dec 2012, pp. 3426–3432.

[17] E. Yildiz, D. Acemoglu, A. E. Ozdaglar, A. Saberi, and A. Scaglione, "Discrete opinion dynamics with stubborn agents," *Available at SSRN: http://ssrn.com/abstract=1744113*, Jan 2011.

[18] M. Pirani and S. Sundaram, "Spectral properties of the grounded laplacian matrix with applications to consensus in the presence of stubborn agents," in *American Control Conference (ACC), 2014*, June 2014, pp. 2160–2165.

[19] J. Ghaderi and R. Srikant, "Opinion dynamics in social networks: A local interaction game with stubborn agents," in *American Control Conference (ACC)*, June 2013, pp. 1982–1987.

[20] T. Qin, H. Yu, C. Leung, Z. Shen, and C. Miao, "Towards a trust aware cognitive radio architecture," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 13, no. 2, pp. 86–95, Sep. 2009.

[21] R. Zhang, J. Zhang, Y. Zhang, and C. Zhang, "Secure crowdsourcing-based cooperative spectrum sensing," in *INFOCOM, 2013 Proceedings IEEE*, April 2013, pp. 2526–2534.

[22] S. Kalamkar, P. Singh, and A. Banerjee, "Block outlier methods for malicious user detection in cooperative spectrum sensing," in *IEEE 79th Vehicular Technology Conference*, May 2014, pp. 1–5.

[23] 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.

[24] 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*, March 2012, pp. 900–908.

[25] X. Liu and J. Baras, "Using trust in distributed consensus with adversaries in sensor and other networks," in *17th International Conference on Information Fusion (FUSION)*, July 2014, pp. 1–7.

[26] A. Vosoughi, J. R. Cavallaro, and A. Marshall, "Robust consensus-based cooperative spectrum sensing under insistent spectrum sensing data falsification attacks," in *IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–6.

[27] S. Atapattu, C. Tellambura, and H. Jiang, *Energy Detection for Spectrum Sensing in Cognitive Radio*. Springer, 2014.

[28] D. Cabric, A. Tkachenko, and R. W. Brodersen, "Experimental study of spectrum sensing based on energy detection and network cooperation," in *Proceedings of the First International Workshop on Technology and Policy for Accessing Spectrum*, ser. TAPAS '06. New York, NY, USA: ACM, 2006.

[29] S. J. Shellhammer, "Spectrum sensing in IEEE 802.22," *IAPR Wksp. Cognitive Info. Processing*, pp. 9–10, 2008.

[30] "Spectrum sensing simulation model, IEEE 802.22-06/0028r10."

[31] A. Goldsmith, *Wireless communications*. Cambridge University Press, 2005.

[32] D. Zordan, G. Quer, M. Zorzi, and M. Rossi, "Modeling and generation of space-time correlated signals for sensor network fields," in *IEEE Global Telecommunications Conference (GLOBECOM)*, Dec 2011, pp. 1–6.

[33] A. Cornejo, S. Viqar, and J. L. Welch, "Reliable neighbor discovery for mobile ad hoc networks," *Ad Hoc Netw.*, vol. 12, pp. 259–277, Jan. 2014.

[34] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, ser. MobiCom '98. New York, NY, USA: ACM, 1998, pp. 85–97.

[35] J. Luo and D. Guo, "Neighbor discovery in wireless ad hoc networks based on group testing," in *46th Annual Allerton Conference on Communication, Control, and Computing*, Sept 2008, pp. 791–797.

[36] S. A. Borbash, A. Ephremides, and M. J. McGlynn, "An asynchronous neighbor discovery algorithm for wireless sensor networks," *Ad Hoc Netw.*, vol. 5, no. 7, pp. 998–1016, Sep. 2007.

[37] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile computing*. Springer, 1996, pp. 153–181.

Aida Vosoughi (S'09) received her B.S. and M.S. degrees in computer engineering from Amirkabir University of Technology, Tehran, Iran in 2006 and 2008, respectively. She received her M.S. in electrical engineering from North Dakota State University, Fargo, ND in 2011. From 2011 to 2016 she was a research and teaching assistant in the VLSI signal processing lab in the Electrical and Computer Engineering Department at Rice University, Houston, TX, where she received her PhD degree in electrical engineering in 2016. Her research interests include security and trust management for cognitive radio ad hoc networks, VLSI design for wireless applications, data encryption/compression and hardware/software co-design.

Joseph R. Cavallaro (S'78, M'82, SM'05, F'15) received the B.S. degree from the University of Pennsylvania, Philadelphia, Pa, in 1981, the M.S. degree from Princeton University, Princeton, NJ, in 1982, and the Ph.D. degree from Cornell University, Ithaca, NY, in 1988, all in electrical engineering. From 1981 to 1983, he was with AT&T Bell Laboratories, Holmdel, NJ. In 1988, he joined the faculty of Rice University, Houston, TX, where he is currently a Professor of electrical and computer engineering. His research interests include computer arithmetic, and DSP, GPU, FPGA, and VLSI architectures for applications in wireless communications. During the 19961997 academic year, he served at the National Science Foundation as Director of the Prototyping Tools and Methodology Program. He was a Nokia Foundation Fellow and a Visiting Professor at the University of Oulu, Finland in 2005 and continues his affiliation there as an Adjunct Professor. He is currently the Director of the Center for Multimedia Communication at Rice University. He is a Fellow of the IEEE and a Member of the IEEE SPS TC on Design and Implementation of Signal Processing Systems and the Chair-Elect of the IEEE CAS TC on Circuits and Systems for Communications. He is currently an Associate Editor of the IEEE Transactions on Signal Processing, the IEEE Signal Processing Letters, and the Journal of Signal Processing Systems. He was Co-chair of the 2004 Signal Processing for Communications Symposium at the IEEE Global Communications Conference and General/Program Co-chair of the 2003, 2004, and 2011 IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP), General/Program Co-chair for the 2012, 2014 ACM/IEEE GLSVLSI, Finance Chair for the 2013 IEEE GlobalSIP conference, and TPC Co-Chair of the 2016 IEEE SiPS workshop. He was a member of the IEEE CAS Society Board of Governors during 2014.

Alan Marshall (M'88-SM'00) received the B.Sc. (Hons.) degree in Microelectronic Systems fro the University of Ulster in 1985 and the PhD from the University of Aberdeen in 1991. He has spent over 24 years working in the Telecommunications and Defense Industries. He has been visiting professor in network security at the University of Nice/CNRS, France, and Adjunct Professor for Research at Sunway University Malaysia. He is currently the Chair in Communications Networks with the University of Liverpool, U.K., where he is also Director of the Advanced Networks Group. He has formed a successful spin-out company, i.e. Traffic Observation & Management (TOM) Ltd specializing in intrusion detection & prevention for wireless networks. He has authored over 200 scientific papers and is the holder of a number of joint patents in the areas of communications and network security. His research interests include Network architectures and protocols, mobile and wireless networks, network security; high-speed packet switching and quality of service and experience architectures; and distributed haptics.

Prof. Marshall is a Fellow of The Institution of Engineering and Technology. He is on the program committees of a number of IEEE conferences. He is a Section Editor (Section B: Computer and Communications Networks and Systems) of the Computer Journal of the British Computer Society and a Member of the Editorial Board of the Journal of Networks (JNW).