Robust Collaborative Spectrum Sensing Schemes in Cognitive Radio Networks

Hongjuan Li\textsuperscript{1,2}, Xiuzhen Cheng\textsuperscript{1}, Keqiu Li\textsuperscript{2}, Chunqiang Hu\textsuperscript{1}, and Nan Zhang\textsuperscript{1}
\textsuperscript{1}Department of Computer Science, The George Washington University, Washington, DC 20052, USA
\textsuperscript{2}School of Computer Science and Technology, Dalian University of Technology, Dalian, 116024, China
E-mail: \{hongjuan,cheng\}@gwu.edu, keqiu@dlut.edu.cn, \{chu,nzhang10\}@gwu.edu

Abstract

Cognitive radio networking allows the unlicensed secondary users opportunistically access the licensed spectrum as long as the performance of the licensed primary users does not degrade. This dynamic spectrum access strategy is enabled by cognitive radio coupled with spectrum sensing technologies. Due to the imperfection of wireless transmissions, collaborative spectrum sensing (CSS) is proposed to significantly improve the probability of detecting the transmissions of primary users. Nevertheless, current CSS techniques are sensitive to malicious secondary users, leading to a high false alarm rate and low detection accuracy on the presence of the primary users. In this paper, we present several robust collaborative spectrum sensing schemes that can calculate a trust value for each secondary user to reflect its suspicious level and mitigate its harmful effect on cooperative sensing. Our approach explores the spatial and temporal correlations among the reported information of the secondary users to determine the trust values. Extensive simulation study has been performed and our results demonstrate that the proposed schemes can guarantee the accuracy of the cooperative sensing system with a low false alarm rate when a considerable number of secondary users report false information.

I. INTRODUCTION

The unlicensed spectrum bands have become more and more crowded due to the explosively growing wireless applications while a large portion of the licensed spectrum remains unused
or under-utilized for significant periods of time. Cognitive Radio (CR) [1], [2] is deemed as a promising technology to achieve a better spectrum utilization under the current static spectrum allocation policy. Via CR, unlicensed users (secondary users, SUs) are enabled to access the temporarily unused/under-utilized licensed spectrum bands on a non-interference basis, and to evacuate the spectrum immediately when a licensed user (primary user, PU) appears in the same channel. Such an Opportunistic Spectrum Sharing (OSS) method allows the unlicensed users to opportunistically share the licensed spectrum with primary users, thereby increasing the efficiency of spectrum utilization.

One of the most challenging problems in a cognitive radio network (CRN) is to identify the presence of primary users. Such a procedure is termed spectrum sensing. In practice, many uncertainty factors resulted from the channel randomness may significantly compromise the detection performance of spectrum sensing. In a heavily shadowed or fading environment, a sensed low primary energy might be due to the faded primary signal rather than a white space. In such a case, secondary users should not confuse the fade effect with that of the white space, thereby causing interference with primary users. Fortunately, the uncertainty problem can be mitigated by allowing the spatially dispersed secondary users to cooperate and share their observations to collaboratively make a decision regarding the status of the licensed band [3]. This is because, due to spatial diversity, it is unlikely for all spatially distributed secondary users to experience the fading or uncertainty problem. In fact, typically most of the secondary users can observe a strong primary user signal, which can be employed to help to make a more accurate decision, overcoming the deficiency of an individual decision. This collaborative spectrum sensing (CSS) has recently been actively studied in [4], [5], [6], [7] due to its attractive performance.

In this paper, we adopt a collaborative spectrum sensing architecture, in which all the participating secondary users forward their observations regarding the presence or absence of a primary user to a fusion center [8], [9], which makes the final decision about whether the primary user is transmitting or not. Like all other wireless networks, cognitive radio
networks are vulnerable to various security attacks. Malicious secondary users may report false spectrum sensing information with an intention of disrupting the final decision.

We propose several robust collaborative spectrum sensing schemes that can make relatively accurate spectrum sensing decisions in spite of the existence of malicious secondary users. In our approach, a trust value, which is computed from the observations of all secondary users, is assigned to each secondary user. The trust value serves as a weight to indicate the suspicious level of a SU and determine how much the local observation of the SU contributes to the final decision. To improve the PU detection performance, we introduce two approaches that respectively explore the spatial correlation and temporal correlation among the SU reports. Although some existing work also considers the spatial variability of the SUIs, our schemes are more robust and reliable, achieving a higher spectrum sensing accuracy at a lower false alarm rate, as evidenced by the comparison-based simulation results. To the best of our knowledge, this is the first work to consider the temporal correlation of the SU reports for collaborative spectrum sensing.

The rest of the paper is organized as follows: In Section II, we overview some related work in cognitive radio networks. Section III describes our system model. In Section IV and Section V, we detail two methods to evaluate the reputation of each secondary user respectively from spatial and temporal domains. Our simulation based performance evaluations are reported in Section VI. Finally, we summarize our work and present the conclusions in Section VII.

II. RELATED WORK

Spectrum sensing [10] is a key function in cognitive radio networks to detect the presence of primary users and identify the available spectrum bands. Its performance in practical application scenarios is often compromised by many factors such as multipath fading, shadowing, and the receiver uncertainty problem [1]. Collaborative spectrum sensing (CSS) [11], [5], [6] is an effective approach to address these challenges and improve the sensing performance.

CSS has received an extensive attention recently. In [5], Ghasemi et al. investigate spectrum
sharing in a heavily shadowed/fading environment. Their results suggest that significant performance enhancements may be achieved through collaborations among the secondary users. It is argued in [6] that CSS can not only improve the detection performance, but also relieve the sensitivity requirement and hardware limitations of the sensing devices. A spectrum-aware sensor network is considered for spectrum sensing in [11]. All the secondary users could query the sensor network, which determines the availability of the spectrum via collaborative data fusion, to obtain the information regarding the spectrum opportunities. In [12], the authors discuss a bandwidth-efficient combination scheme in which the bandwidth required for reporting is fixed regardless of the number of cooperative users. In [13], the problem of collaborative spectrum sensing is addressed under the bandwidth constraints by employing a censoring method with quantization [14] to decrease the average number of sensing bits sent to the fusion center. This approach makes a local decision based on the sequential probability ratio test (SPRT) and reports only the decision result. Such a censoring procedure decreases the usage of bandwidth by reducing the unnecessary reportings.

Note that most of the current research assumes that the secondary users are honest, which is not always true. In fact, like all other wireless networks, CRNs are also subject to various security threats. Kaligineedi et al. [7] present a technique to identify malicious secondary users and nullify their effect on cooperative sensing by first applying a simple outlier detection mechanism to pre-filter the extreme values in the sensing data, and then evaluating trust factors utilized as the weights in calculating the mean of the received data. This approach only considers simple ‘always yes’ and ‘always no’ attacks without dealing with more sophisticated malicious behaviors. Similar security issues in CSS have been addressed in [15], where the authors model the problem of spectrum sensing as a parallel fusion network, and propose a reputation based fusion scheme to combat malicious attacks. A reputation rating is calculated for each secondary user based on its sensing accuracy. Then the reputation values are converted to the weights to be used in the weighted SPRT for data fusion. This approach is dependent on a priori knowledge of the reporting values regarding the considered spectrum band, i.e., it
requires the true state of the world. Generally speaking, [7] and [15] are the most related to our work as they all employ weight values for mitigating the malicious effect on CSS. But we take completely different approaches to estimate the trust values and obtain significantly better results\(^1\) as evidenced by the extensive simulation study.

In recent research, some features of cognitive radio networks are utilized to improve the robustness of CSS. Min \textit{et al.} [16] propose an attack-tolerant distributed sensing protocol (ADSP) that considers the shadow-fading correlation among nearby sensors. Sensors in close proximity are grouped into a cluster to cooperatively safeguard distributed sensing. The fusion center pre-filters abnormal sensor reports by a novel shadow fading correlation-based filter via cross-validation. However, this work falls short in cases where malicious nodes are more than 1/3 of the sensors in a cluster. A new approach to PU detection based on the estimated transmit-power level is proposed in [17]. This approach was termed IRIS for robust cooperative sensing via iteRatIve State (IRIS) estimation. Its key insight lies in that the sensing results are governed by the network topology and the law of signal propagation. By checking the consistency among sensing reports with the estimated transmit-power and the path-loss exponent, IRIS safeguards the PU detection process. To estimate the system states, IRIS requires the global knowledge of the network topology.

Alternative network architectures are also exploited to address the robust collaborative spectrum sensing problem. In [18], the spectrum sensing data falsification (SSDF) attack is countered in a distributed manner [19]. Each secondary user iteratively exchanges its sensing data with its selective neighbors such that a consensus could be gradually reached. This consensus-based scheme improves the reliability of cooperative spectrum sensing by isolating malicious users in the neighborhood, but it incurs extra communication overhead. In their recent work, Fatemieh \textit{et al.} [20] view the area of interest as a grid of squared cells. The mechanism of identifying outlier measurements is based on a hierarchical structure: (1)

\(^1\)We compared with [7] only in our simulation study due to the lack of a priori channel information, which is required by [15].
intra cell among individual CR measurements and (2) inter-cell among neighbor cells. The limitation of this scheme lies in that the SUs are assumed to be deployed uniformly with a fixed density per square meter, which might not always hold in practice.

In this paper, we tackle a similar problem in which malicious secondary users intend to introduce false information to subvert the entire process of spectrum sensing. Generally, Our approach differs from the previous work in three key aspects. Firstly, we address this problem from both the space domain and the time domain when deriving the weight value to evaluate the trust worthiness of each SU. Secondly, a robust estimate method is applied in the trust value computation that leads to a more accurate and effective primary user detection even with the existence of a large fraction of malicious nodes. Thirdly, we employ affinity propagation [21] in temporal correlation exploitation. This is the first time for affinity propagation to be utilized for robust collaborative spectrum sensing. It can not only enhance the CSS performance, but also significantly save the energy consumption.

III. System Model and The Robust CSS Framework

The process of CSS starts with sensing a primary channel at each individual secondary user. Here a “primary channel” refers to a licensed spectrum band currently being utilized by a primary user. There might exist multiple primary users but we focus on one primary channel for simplicity. This is reasonable as multiple primary users can not operate at the same channel simultaneously in close neighborhood. Such an assumption implies that all the secondary users reside in a “relatively close” neighborhood in order to observe the activity of the same primary user.

Following the general assumptions on collaborative spectrum sensing, we assume that the system is time-slotted. This is reasonable as SUs need to sense the channel periodically in order to vacate in time when a PU comes back. Therefore all SUs need to be time-synchronized. Note that at each slot, time is divided into two intervals, with one for spectrum sensing and one for SU transmission if spectrum is available. To simplify presentation, we
ignore the transmission interval in a time slot, and focus on the sensing interval. We further assume that the length of a time slot is much shorter than the time a PU stays in an activity (silence or transmitting). In other words, a PU stays in a busy or idle state for multiple continuous time slots. This assumption is consistent with IEEE 802.22, which requires a SU to vacate the channel within two seconds when a PU appears, implying that SUs need to sense the channel once every two seconds and PUs operate at much longer time. At each time slot, SUs report to the fusion center their observations, based on which the final decision regarding the channel status will be made. Such a process is termed a round of spectrum sensing.

In this paper, energy detection [22], [23] is chosen as the underlying spectrum sensing technique because of its simple design and low overhead. The goals of local spectrum sensing are twofold: first, secondary users should not cause harmful interference to primary users to ensure PU’s quality of service (QoS); second, a secondary user should efficiently detect the white space or underutilized spectrum and ensure its own QoS.

Our collaborative spectrum sensing model is illustrated in Figure 1. Suppose that there are \( n \) collaborative secondary users \( \{S_1, S_2, \ldots, S_n\} \) that detect the presence or absence of the primary user at the considered channel. These secondary users report their detected power levels and their independent decisions about whether the primary user is transmitting or not to a centralized decision maker at each time slot. The output of the energy detector from each secondary user \( S_i \) is denoted by \( u_i \), and the independent decision is denoted by \( d_i \). In other words, \( u_i \) represents an estimate of the primary power received at node \( S_i \). As introduced in [24], \( u_i \) can be expressed by \( u_i = u_p - (10 \log_{10} r_i^\alpha + H_i + F_i) \) in dB, where \( u_p \) is the transmit-power of the primary signal, \( r_i \) is the distance from \( S_i \) to the primary transmitter, \( 10 \log_{10} r_i^\alpha \) represents the signal attenuation with exponent \( \alpha \) (typically \( 2 < \alpha < 4 \)), and \( H_i \) and \( F_i \) are losses due to shadowing and multipath fading. We also adopt the log-normal shadowing model [25], in which the sum of \( H_i \) and \( F_i \) follows a Gaussian distribution \( H_i + F_i \sim N(\mu_s, \sigma^2) \) on the dB scale [20]. Then we have \( u_i \sim N(\mu'', \sigma^2) \), where \( \mu'' = u_p - (10 \log_{10} r_i^\alpha + \mu_s) \). In this paper, we consider \( \mu_s \) to be 0 for simplicity of the analysis.
Based on the reports from all the secondary users, the decision maker implements some data fusion scheme [26] and makes the final decision $D$. Then the final decision is broadcast to all the secondary users. Such a procedure is performed for each time slot.

![Collaborative spectrum sensing diagram](image)

**Fig. 1: Collaborative spectrum sensing.**

A malicious secondary user may significantly affect the final decision by sending false sensing information in this model. Especially, when the OR rule (which is commonly used because it has the best performance when there is no attacker) is used in collaborative sensing, the existence of a malicious secondary user may completely subvert the spectrum sensing decision. For example, whenever there is a secondary user reporting the existence of the primary user, the decision maker always believes that the spectrum is in use according to the OR rule. Thus malicious secondary users can easily launch false alarm attacks. Then the spectrum can never be opportunistically accessed, causing severe under-utilization of the spectrum. Note that there are other attacks that may cause miss detection of the primary user, which could result in interference to the communications of the primary user.

To resolve the above-mentioned problems, we intend to enhance the robustness of collaborative spectrum sensing by utilizing a weighted approach to detect the presence of a primary user. A trust factor $\beta_i$ is assigned to each user $S_i$ as the measurement of the reliability of $S_i$, and all $\beta_i$’s satisfy $\sum_{u_i \in U} \beta_i = 1$. A higher trust value $\beta_i$ indicates a higher probability that $S_i$ is a normal user. Hence the user that has a high trust factor should contribute more to the final decision, and vice versa. Therefore trust values are used as the weighting factor to
calculate the weighted average of the energy values obtained from the participating SUs. The final decision is made by comparing the weighted sum of the nodes’ energy to a detection threshold:

\[
\begin{cases} 
\mathcal{H}_1, & \sum_{i=1}^{n} \beta_i u_i \geq \eta; \\
\mathcal{H}_0, & \text{otherwise},
\end{cases}
\]

(1)

where \(\mathcal{H}_0\) and \(\mathcal{H}_1\) denote the hypothesis of the absence and presence, respectively, of the primary user in the considered channel.

According to the approach in [24], the detection threshold \(\eta\) can be determined based on the power of the primary transmitter and the surrounding area defined by radius \(r\) that needs to be protected. In our consideration, this threshold should ensure that the probability of the miss detection stays below a threshold (e.g. 0.05). Thus \(\eta\) can be computed as follows [24]. Let \(u_{avg}\) be the average of the received powers by the \(n\) secondary users. If each measurement at distance \(r\) follows a normal distribution with the mean \(u_r = u_p - (10\log_{10} r^\alpha)\) and the standard deviation \(\sigma\), we have \(u_{avg} \sim N(u_r, \frac{\sigma^2}{n})\). Thus \(\eta\) can be derived by:

\[
Pr(u_{avg} \geq \eta) = 0.95 \Rightarrow \eta = -\frac{\sigma}{\sqrt{n}} Q^{-1}(0.95) + u_r,
\]

(2)

where \(Q\) is the tail function of the standard Gaussian distribution and \(Q^{-1}\) is its inverse [24].

Note that the final decision needs to be made at each time slot at the fusion center but the trust value of each PU needs to be computed from history and neighborhood information. In the following two sections, we present the details of our algorithm for trust factor evaluation. We will introduce two schemes to investigate the spatial (neighborhood) correlation and temporal (history) correlation of the reported information for robust collaborative spectrum sensing.
IV. TRUST FACTOR EVALUATION EXPLORING SPATIAL CORRELATION

Since all the secondary users under our consideration reside in a relatively close neighborhood and report their received power level information regarding the same primary channel, they are expected to behave similarly under normal conditions. Thus SUs whose reports deviate significantly from others tend to be malicious. In this section, we explore the spatial correlation among the secondary user reports at each time slot to mitigate the influence of malicious secondary users on the final spectrum sensing decision.

A. Exploring the Spatial Correlation

The centralized decision maker detects if any outlying report exists by studying the data set \( U = \{u_i|i = 1, 2,\ldots,n\} \) for each time slot. The detection is conducted by computing the normalized distance between each reading \( u_i \) to the “center” of the data set \( U \).

As argued in Section III, \( U \) approximately follows a Normal distribution since regular secondary users should report similar results. Thus we employ the mean \( \mu \) of \( U \) as its “center”. Moreover, to compute the normalized distance of each \( u_i \) to the “center” of \( U \), we need to compute the variance of \( U \), denoted by \( \sigma^2 \). Clearly, a simple solution is to estimate \( \mu \) and \( \sigma^2 \) using the sample mean and the sample variance of the data set \( U \). Let \( \hat{\mu} \) and \( \hat{\sigma}^2 \) be the estimate of \( \mu \) and \( \sigma^2 \), respectively. We have

\[
\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} u_i \quad (3)
\]

\[
\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (u_i - \hat{\mu})^2 \quad (4)
\]

Such a method is extremely simple but the sample mean and sample variance may not be reliable, since they are sensitive to the presence of outliers. When the number of outlying reports is large, their values can distort the estimates of \( \mu \) and \( \sigma^2 \), so that the detection via distances may fail to identify true outliers. Therefore, robust estimators \( \hat{\mu} \) and \( \hat{\sigma}^2 \) are required,
which are expected to be less influenced by outlying reports and thus generate estimates close to the true values of $\mu$ and $\sigma^2$. Throughout this section, we employ the Orthogonalized Gnanadesikan-Kettenring (OGK) estimators $\hat{\mu}$ and $\hat{\sigma}^2$ [27], as described below.

Treat $U = \{u_1, u_2, \ldots, u_n\}$ as a single-variate sample set coming from a distribution with the mean $\mu$ and the variance $\sigma^2$. Let $\mu_0$ and $\sigma_0$ be the median and the MAD$^1$ of $U$, respectively. Define a weight function $W(x) = (1 - (x/c_1)^2)I(|x| \leq c_1)$ and a $\rho$-function $\rho(x) = \min(x^2, c_2^2)$, where $c_1 = 4.5$ and $c_2 = 3$. Then $\mu$, $\sigma^2$ can be estimated respectively by [28]:

$$
\hat{\mu} = \frac{\sum_{i=1}^{n} u_i W(v_i)}{\sum_{i=1}^{n} W(v_i)} \text{ for } v_i = \frac{u_i - \mu_0}{\sigma_0}
$$

(5)

$$
\hat{\sigma}^2 = \frac{\sigma_0^2}{n} \sum_{i=1}^{n} \rho\left(\frac{u_i - \hat{\mu}}{\sigma_0}\right)
$$

(6)

OGK provides a robust estimate, i.e. it provides a reliable estimate for the bulk of data when there exists a small fraction of outliers in the sample data set. The OGK estimator employs an intuitive approach followed by a clever correction to ensure that the estimate is positive definite. Besides OGK, there also exist a number of other approaches to generate robust estimates, which are used to help calculate reliable distances in the presence of outliers. However, the application of most of these estimates is restricted either by a low breakdown point$^2$ (e.g. M-estimates), or by a high computational overhead (e.g. MCD, MVD, SDE, P-estimates etc.). We choose the above OGK because it ensures a high breakdown point at the expense of a low computational cost.

Standardizing the data set $U$ we can obtain $y_1, \ldots, y_i, \ldots, y_n$ [29] as follows:

$$
y_1 = \left|\frac{u_1 - \hat{\mu}}{\hat{\sigma}}\right|, \ldots, y_i = \left|\frac{u_i - \hat{\mu}}{\hat{\sigma}}\right|, \ldots, y_n = \left|\frac{u_n - \hat{\mu}}{\hat{\sigma}}\right|
$$

(7)

$^1MAD(Y) = \text{median}(|Y - \text{median}(Y)|)$.

$^2$A breakdown point is defined to be the maximal proportion of outliers that the estimator can tolerate.
The robust estimates introduced above are less influenced by the outlying values, and can provide more accurate assessments of the mean and the variance of $U$.

B. Trust Factor Evaluation

To assess the suspicious level of secondary users, we calculate a trust factor $\beta_i$ from the aspect of history behavior. This is reasonable since a secondary user that always reports false information tends to be malicious.

1) Trust Factor Evaluation using Statistics over a Certain Time Period: In section IV-A, we denote by $y_i$ the distance between each reading $u_i$ to the “center” of the data set. In this subsection, we let $y_i(t_j)$ denote the distance obtained at round $t_j$ (time slot $t_j$). Since all the secondary users should exhibit similar observations regarding the presence or absence of the primary user, $\beta_i$ can be computed based on the degree of $u_i$’s deviation from the “center” of the data set $U$ over a certain time interval consisting of $T$ time slots. This can be done by summing up the distance of $u_i$ to the “center” at each round during $T$ to obtain $L(i, t_j)$:

$$L(i, t_j) = \sum_{j' = j - T + 1}^{j} y_i(t_{j'})$$  \hspace{1cm} (8)

Obviously, the more $u_i$ deviates from the other readings over the given period, the more unreliable $S_i$ is; consequently, the lower $\beta_i$ is. Thus, trust factors (denoted by $\beta'$ before normalization) can be obtained by the following operation:

$$\beta_i' = L(Min, t_j)/L(i, t_j)$$  \hspace{1cm} (9)

where $L(Min, t_j) = \min\{L(i, t_j)|i = 1, 2, ...n\}$. To ensure that the sum of all the newly computed trust factors is equal to 1, we perform the following normalization process:

$$\beta_i = \frac{\beta_i'}{\sum_{u_j \in U} \beta_j'}$$  \hspace{1cm} (10)
2) Trust Factor Evaluation using an Update Mechanism based on Each Decision: Another option is to evaluate the trust factor \( \beta_i \) directly from the decision made by each secondary user at every round. In our model, each secondary user performs local spectrum measurements independently and then makes a binary decision to infer the absence or presence of the primary user in the observed frequency band. We adjust the value of \( \beta_i \) by comparing the local decision of each SU and the final decision made by the centralized fusion center. The value of \( \beta_i \) increases when the report of each secondary user \( d_i \) is consistent with the final decision \( D \), while decreasing otherwise.

Initially, every \( \beta_i \) is set to \( \frac{1}{n} \). After each decision process (within one round) is finished, \( \beta_i \) is updated according to the following two steps:

Step 1: Compute \( \beta'_i \):

\[
\beta'_i = \begin{cases} 
\beta_i (1 - \lambda) & \text{if } (d_i \neq D) \\
\beta_i (2 - \lambda) & \text{o.w.}
\end{cases}
\]

where

\[
\lambda = \frac{y_i}{y_{\text{max}} + 1}.
\]

Step 2: Perform normalization:

\[
\beta_i = \frac{\beta'_i}{\sum_{u_j \in U} \beta'_j}
\]

In step 1, we let the value of \( \beta_i \) change according to each SU’s local decision. Moreover, the update process of \( \beta_i \) is also related to the variation of \( u_i \) in the data set \( U \). When \( d_i \neq D \), the more the \( u_i \) deviates from the other readings, the more unreliable the \( S_i \) is; hence its trust factor decreases more quickly. Conversely, when \( d_i = D \), the more the \( u_i \) deviates from the other readings, the more slowly the \( \beta_i \) increases.

V. Trust Factor Evaluation Exploring Temporal Correlation

Note that in the cognitive radio network under our consideration, all the SUs monitor the same considered frequency band collaboratively; thus their energy outputs should experience
the same change trend. Specifically, when a primary user emerges or disappears at a certain time instant $t_k$, the energy output of each SU should change accordingly. We call this change trend at time instant $t_k$ a change point. Clearly, the change point at a given time instant from all SUs are temporally correlated. Thus, we are motivated to explore the temporal correlation among these change points to enhance the performance of collaborative spectrum sensing.

In this section, the network model varies a little bit as shown in Figure 2. Secondary users monitor the considered frequency band, and do some simple, local computation to find out the change points based on the detected energy values. Instead of transmitting all the energy values to the fusion center, if a secondary user detects a change point $c_i$, it reports the change point to the centralized fusion center, which makes the final decision based on the received information.

![Collaborative spectrum sensing based on change point detection.](image)

Fig. 2: Collaborative spectrum sensing based on change point detection.

### A. Local Change Point Detection using Affinity Propagation

Clustering data by identifying a subset of representative examples is important for processing sensory signals and detecting patterns in the data [21]. In this paper, we aim to find out the change points of each SU’s energy outputs using clustering method. Before we present the details of change point detection, a high level overview of the process is introduced as follows.

Assume that the PU emerges at the time instant $t_k$. We consider a given secondary user that monitors the frequency band of interest. Theoretically, its energy data detected before $t_k$
should form a cluster that has small values, and the energy data detected after \( t_k \) form another cluster that has large values. The time instant \( t_k \) that separates the two individual clusters is the change point in our consideration. Similarly, if a PU disappears at time instant \( t_{k'} \), the energy data after \( t_{k'} \) should belong to a cluster with small values. And \( t_{k'} \) is also a change point. We illustrate this procedure in Figure 3.

![Fig. 3: Change point detection](image)

In this paper we employ Affinity Propagation [21] to cluster the secondary user readings. Affinity propagation is a technique to find clusters more accurately than other methods, and it does so in less time. We apply affinity propagation to cluster each SU’s energy values based on which to identify the change points.

1) **Affinity Propagation:**

A common clustering algorithm is to use data to learn a set of centers such that the sum of the squared errors between the data points and their nearest centers is the minimum. The centers selected from actual data points are called “exemplars”.

Traditionally, most techniques (e.g. the popular \( k \)-center clustering [30]) for identifying exemplars randomly choose a fixed set of candidate exemplars from the initial data points, and iteratively refining them while searching for a better solution. Affinity propagation, on the other hand, simultaneously considers all the data points as potential exemplars. It operates by recursively exchanging real-valued messages between data points until a good subset of data points emerge as exemplars and the corresponding clusters are determined. In contrast
to the k-center clustering technique, which only works well when the initial choice is close to a good solution, affinity propagation achieves a much lower error rate in much less time [21].

The input to affinity propagation is a collection of similarities $s(i,k)$, with each being a real-valued number characterizing the similarity between point $i$ and $k$ and indicating how well the data point $k$ is suited as the exemplar of data point $i$. In our consideration, the secondary user $S_i$ considers the similarity between its energy values at different time points, i.e., the similarity between $v_i(t_x)$ and $v_i(t_y)$, which is denoted by $s_i(t_x, t_y)$, where $v_i(t_x)$ and $v_i(t_y)$ denote the energy values detected by the secondary user $S_i$ at time $t_x$ and $t_y$, respectively.

As shown in Figure 3, it is obvious that the time period is divided into several stages by change points. The goal of affinity propagation is to cluster the energy values in each stage to one group. In other words, we expect a high similarity between two data points only if they have both similar energy values and similar time instants. By taking both energy value difference and time separation into consideration, the similarity between two points can be simply set as follows:

$$s_i(t_x, t_y) = -|v_i(t_x) - v_i(t_y)| * (t_x - t_y)^2$$

Besides, the affinity propagation algorithm takes as inputs a set of real numbers $s(t_k, t_k)$ for each data point $t_k$, where $s(t_k, t_k)$ is a priori preference on which the point $t_k$ can be chosen as an exemplar. The data points with larger values of $s(t_k, t_k)$ are more likely to be chosen as exemplars. And the values of the initial input preference can influence the number of clusters. If a priori, all data points are equally suitable as exemplars, the preferences should be set to a common value. Often, a good choice is to set all preferences to the median of the input similarities (resulting in a moderate number of clusters). Thus we set the initial preferences to the median in this paper.
The data points exchange two kinds of messages, with each considering different aspects of the competition. The “responsibility” \( r(t_x, t_y) \), sent from data point \( t_x \) to candidate exemplar point \( t_y \), indicates the accumulated evidence for how well-suited point \( t_y \) is as the exemplar for point \( t_x \) in contrast to other potential exemplars. The “availability” \( a(t_x, t_y) \), sent from a candidate exemplar point \( t_y \) to point \( t_x \), indicates how much support point \( t_y \) has received from other points for being their exemplar. It also can be comprehended as the accumulated evidence on how appropriate it would be for point \( t_x \) to choose point \( t_y \) as its exemplar, taking into account the support from other points that point \( t_y \) should be an exemplar. As the message-passing procedure proceeds, the messages can be combined at any stage, and the responsibilities and availabilities will become more extreme until a clear set of decisions can be made about which points are exemplars and for every other point, which exemplar it belongs to.

The affinity propagation algorithm is outlined as follows:

**Affinity Propagation (Science, 2007)**

**Initialization:**

\[
r(t_x, t_y) = 0, a(t_x, t_y) = 0 \quad \text{for all } t_x, t_y
\]

**Responsibility updates:**

\[
r(t_x, t_y) \leftarrow s(t_x, t_y) - \max_{y' \neq y, t, t' \neq t} \{a(t_x, t_{y'}) + s(t_x, t_{y'})\}
\]

**Availability updates:**

\[
a(t_k, t_k) \leftarrow \sum_{x' \neq k} \max\{0, r(t_{x'}, t_k)\}
\]

\[
a(t_x, t_y) \leftarrow \min\{0, r(t_y, t_x)\} + \sum_{x' \neq \{x,y\}} \max\{0, r(t_{x'}, t_y)\}
\]
Making cluster assignments:

\[
\text{if } r(t_x, t_y) + a(t_x, t_y) = \max_{y'} \{ r(t_x, t_{y'}) + a(t_x, t_{y'}) \} \\
\text{then } t_y \text{ is the exemplar of } t_x.
\]

For point \( t_i \), the value of \( t_k \) that maximizes \( a(t_i, t_k) + r(t_i, t_k) \) identifies \( t_k \) as the exemplar of point \( t_i \), or if \( i = k \), \( t_i \) is an exemplar. Affinity propagation associates each data point with one exemplar, making the whole data set partitioned into a set of clusters.

As introduced in [21], the message-passing procedure may be terminated after a fixed number of iterations, after changes in the messages fall below a threshold, or after the local decisions stay constant for some numbers of iterations.

2) Change Point Detection

After each secondary user applies affinity propagation, it can detect the change points at which the surrounding energy values (the energy values before and after the change point) belong to different clusters. In other words, the energy values begin to increase or decrease after a change point, indicating that the primary user is emerging or disappearing.

B. Evaluation of the Trust Factors using Change Points

Since all the secondary users monitor the same frequency band, once a primary user emerges or disappears at some time instant, the change points detected by different SUs should be temporally correlated. This phenomenon is explored for trust factor determination.

After all the secondary users report their change points, the centralized decision maker can generate a 1/0 sequence for each SU based on their change points, which is a sequence of the existence/absence of the change point at each time slot. Specifically, at each time slot \( t_i \), each SU has a 1/0 value, with 1 meaning that there is a change point at \( t_i \) and 0 the absence of a change point at \( t_i \).

To determine whether or not a reported change point by a secondary user is a true change point, majority vote is applied for each time slot \( t_i \). If the majority of the SUs reports a
change point at \( t_i \), the centralized decision maker concludes that the change point is a real one. Similarly, if the majority reports no change point at \( t_i \), the decision maker concludes that no change point occurs at \( t_i \). Let \( x \) be a binary variable associated with a secondary user \( S \) and set \( x = 1 \) if and only if \( S \) belongs to the majority group. Then we obtain a new sequence \( \{ x \} \) for each SU, which indicates whether the SU is in the majority group or not at each time slot, from the original sequence labeling the presence/absence of the change points according to the measurements of the SU.

Intuitively, if \( x \) contains more number of 1’s than 0’s, the corresponding SU is more trustworthy; otherwise, it is inclined to be malicious. Then we can use the trust factor evaluation method introduced in section IV-B2 to update the trust factor of each SU and mitigate the malicious influence on the final decision.

C. Discussion

In collaborative spectrum sensing, secondary users consume extra energy since they are involved in both local sensing and data reporting. In particular, if the number of cooperating secondary users or the amount of sensing results for reporting is large, the energy consumption overhead can be significant. The change point detection mechanism stated in the previous subsection can effectively address this issue. In this scheme, secondary users only need to do some simple local computation based on the detected energy values and find out the change point that indicates whether a primary user is emerging or disappearing. Instead of reporting all the energy values, only the change points need to be transmitted when the status of the considered channel changes. By refraining the cooperating secondary users from transmitting unnecessary or uninformative data, the communication overhead can be reduced, and the energy efficiency can be improved.

In cooperative sensing, a common control channel (CCC) [1], [31] is commonly used by secondary users to report local data to the centralized fusion center or share the sensing results with neighboring users. With our approach, the control channel bandwidth requirement can
be lowered due to the reduced number of control messages.

VI. SIMULATION EVALUATION

The performance of spectrum sensing can be primarily described by two basic metrics: the false alarm probability $P_f$ and the detection accuracy $P_d$, with $P_f$ denoting the probability of a secondary user declaring that a PU is present when the spectrum is actually free, and $P_d$ denoting the probability of a secondary user declaring that a PU is present when the spectrum is indeed occupied by the PU. A miss detection causes interference with the primary user, and a false alarm results in severe under-utilization of the spectrum band. It is therefore a challenging problem to find an optimal detection scheme that maximizes the probability of detection while minimizing the probability of false alarms.

A. Simulation Setup

We consider a network topology within a $1000m \times 1000m$ square area in our simulation study. The secondary users are randomly distributed around the primary user located in the center of the network region. The network size $n$ is selected from $\{25, 50\}$, and the the transmit-power of the primary user is chosen from $\{20dBm, 40dBm\}$. The window size $T$ for secondary users’ history behavior considered in Section IV-B1 is selected from $\{5, 10\}$. We assume that a circular area with a radius $707m$ ($\sqrt{2} \times 500m$) around the primary user is the area that needs to be protected. In other words, the whole CR network is within the protection range of the PU. The propagation loss factor $\alpha$ is set to 3, and the standard deviation of the fading and shadowing process, $\sigma$, is 3 (in dB scale). We require that the primary signal be detected with a probability greater than 0.95. From Equation (1), the detection threshold $\eta$ should be set to $-66.2dBm$. Note that under the hypothesis of $H_0$, there is zero mean additive white Gaussian noise (AWGN) with variance $\sigma' = 1$ in the considered channel.

For normal secondary users, the outcome of the energy detector depends on its location and the losses due to multipath fading and shadowing. For malicious secondary users, the reported values are randomly selected from the range $[\eta-100, \eta+100]$dBm.
B. Simulation Results When Exploiting Spatial Correlation

In this subsection, we study the performance of our algorithms that explore the spatial correlation of the sensing results obtained by the secondary users. For the trust factor evaluation, we denote by “trust1” the method using statistics over a certain time period $T$, and “trust2” the method using the updating mechanism at each round. For both methods, we apply OGK as the basic algorithm to generate robust estimates of the mean and the variance.

1) Performance Comparison: As mentioned in our Related Work section (Section II), [7] and [15] are the most related to this work but we decide not to compare our work with [15] as the latter requires a priori spectrum information, which is unavailable in our study. In [7], Kaligineedi et al. present a secure cooperative sensing techniques (SCST) for cognitive radio systems to tackle the same problem. The difference lies in that SCST employs a simple and fast combination scheme to avoid the calculation of the mean and the variance of the sensing data, which might be influenced by the presence of the outlying values. In this study, we compare our schemes with SCST to get a deep understanding about these methods by varying the fraction of the malicious nodes.

![Fig. 4: Comparison of the detection accuracy.](image)

From Figure 4, we observe that our methods trust1 and trust2 can achieve high detection accuracy (100%) when the fraction of the malicious nodes is below 65%. For SCST, the
detection accuracy starts to decrease dramatically when the fraction of the malicious nodes is more than 30%. In general, the detection accuracy shows a downward trend along with the increase of the number of malicious nodes. Obviously, our schemes demonstrate better robustness and effectiveness in detecting the presence of a primary user.

Figure 5 shows the performance of false alarm probability with the increase of the number of malicious nodes for different methods. The false alarms for both trust1 and trust2 remain 0 when the fraction of the malicious nodes is less than 25%. The figure also shows that the false alarm probability of SCST is much higher than those of our schemes. With more malicious nodes, more bogus information are injected, causing the increase of the false alarm probability. However, our schemes are more effective to restrain the false alarms.

The above figures demonstrate the superiority of our methods over SCST. In the following section, we further study the key factors that can affect the performance of our algorithms.

2) Impact of the Network Scale: Figures 6 and 7 depict the impact of the network scale on trust1 and trust2. We observe that a larger network scale results in a better detection accuracy and a lower false alarm probability for both methods. This improvement mainly comes from the increase in the size of the sample data set, since a larger size implies that there is more
information available to estimate the dispersion of the distribution. And the improvement is more obvious for trust2, which means that the network scale makes a greater impact on trust2.

3) Impact of the PU Transmit-Power: Intuitively, the detection accuracy would be better with a higher PU transmit-power. However, Figure 8 and 9 indicate that the detection accuracy varies little for different PU transmit-powers. This is because a higher transmit-power results in a higher energy threshold. The process of primary user detection is not impacted by the value of transmit-power. However, a higher transmit-power leads to a lower false alarm probability.
for both methods. Obviously, to have a higher false alarm, the weighted sum of the energy needs to be larger than the threshold when the primary user is absent. But the threshold is getting larger when the PU transmit-power increases; that’s why the probability of false alarm is smaller.

4) Impact of the Time Period $T$: In trust1, the trust factor of each SU is derived by obtaining its deviation from the “center” over a certain time period $T$. Figure 10 plots the impact of $T$ on trust1. It shows that our scheme can achieve a better performance in terms of
Fig. 10: The impact of $T$ on trust1.

both detection accuracy and false alarm probability with a larger $T$. This is mainly because a larger $T$ provides more available information to assess the trust worthiness of each secondary user.

C. Simulation Results When Exploiting Temporal Correlation

In this subsection, we study the performance of our algorithm that explores the temporal correlation of the SU spectrum sensing results. We denote by “CP” the method based on change points computed from affinity propagation.

We assume that, for the first 10 time slots (from $t_1$ to $t_{10}$), the primary user is absent ($\mathcal{H}_0$). In the next 10 slots (from $t_{11}$ to $t_{20}$), the primary user is present ($\mathcal{H}_1$). The law of this cycle holds in the time domain. And the transmit-power of the PU is set to 20dBm. For this parameter setting, we compute the similarity between two data points as follows:

$$s_i(t_x, t_y) = -|(v_i(t_x) - v_i(t_y)) \cdot (t_x - t_y)^2|$$

1) Performance Comparison: Figure 11 compares the performance of our scheme CP and that of SCST. Since we utilize majority vote in CP, the performance will not be rosy when the fraction of the malicious nodes is more than 50%. We observe that the detection accuracy
begins to decrease when the fraction of the malicious nodes is close to 0.5. Generally speaking, our scheme can achieve a higher detection accuracy while maintaining a nice property of lower false alarm probability.

2) Impact of the Network Scale: As discussed in the previous section, a larger network scale leads to a better detection accuracy and a lower false alarm probability. We observe the same trend in Figure 12. The simulation also shows that our algorithm is robust in that its performance degrades very slowly in terms of detection accuracy while the false alarm is
refrained at a low level.

VII. CONCLUSION

In this paper, we propose a robust spectrum sensing scheme for cognitive radio networks. By exploring the spatial and temporal correlations among the spectrum sensing results of the secondary users, our algorithm can achieve a high detection accuracy and a low false alarm rate as indicated by the extensive simulation study. By applying the OGK estimator, our algorithm can provide a reliable estimate for the bulk of data when there exists outliers in the sample data set, in turn exploring the correlation among SUs more accurately. Besides, we propose a novel idea by conducting the change point detection via affinity propagation. The main idea of exploring the temporal correlation among change points can not only enhance the sensing performance, but also achieve high energy efficiency. The spectrum sensing algorithms developed in this paper are novel, robust, and effective for primary user detection.

ACKNOWLEDGMENT

This work is supported by NSFC under Grants 60973115, 60973117, 61173160, 61173162, and 61173165, the National Science Foundation of the US under grant CNS-0831852, and the Intel-MoE Joint Research Fund.

REFERENCES


