A new algorithm for considering green communication and excellent sensing performance in cognitive radio networks

Tangsen Huang¹, Xiangdong Yin¹ and Qingjiao Cao²

Abstract
Multi-node cooperative sensing can effectively improve the performance of spectrum sensing. Multi-node cooperation will generate a large number of local data, and each node will send its own sensing data to the fusion center. The fusion center will fuse the local sensing results and make a global decision. Therefore, the more nodes, the more data, when the number of nodes is large, the global decision will be delayed. In order to achieve the real-time spectrum sensing, the fusion center needs to quickly fuse the data of each node. In this article, a fast algorithm of big data fusion is proposed to improve the real-time performance of the global decision. The algorithm improves the computing speed by reducing repeated computation. The reinforcement learning mechanism is used to mark the processed data. When the same environment parameter appears, the fusion center can directly call the nodes under the parameter environment, without having to conduct the sensing operation again. This greatly reduces the amount of data processed and improves the data processing efficiency of the fusion center. Experimental results show that the algorithm in this article can reduce the computation time while improving the sensing performance.

Keywords
Cognitive networks, node selection, spectrum sensing, machine learning

Date received: 21 January 2020; accepted: 17 April 2020
Handling Editor: Zheng Chang

Introduction
Developments in wireless communication technology have increased the need for spectrum resources, which are currently limited.¹,² To address this problem, cognitive radio networks have been proposed to improve existing spectrum resources.³,⁴ In this context, spectrum sensing technology is the basic link between cognitive radio networks.⁵

Sensing nodes need to quickly and accurately perform spectrum sensing in order to efficiently utilize the idle frequency band without interfering with the primary user.⁶ This presents another issue; due to the impacts of path loss, shadow fading, and hidden terminals, it is difficult for a single sensor node to accurately detect the primary user’s status.⁷,⁸ Nevertheless, cooperative sensing can effectively overcome these impacts by fusing detection information from multiple nodes in different geographical locations.⁹ In centralized cooperative sensing, there is a special data fusion center in the cognitive network. This collects local perception results from the nodes participating in

¹School of Electronics and Information Engineering, Hunan University of Science and Engineering, Yongzhou, China
²School of Water Conservancy and Hydroelectric Power, Hebei University of Engineering, Handan, China

Corresponding author:
Xiangdong Yin, School of Electronics and Information Engineering, Hunan University of Science and Engineering, Yongzhou 425100, China. Email: rocketop@163.com

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
cooperative perception, judges the current usage of authorized bands, and then broadcasts the decision results in the network or directly controls and schedules the perception nodes. The local sensing result collection process will increase the communication overhead when a large number of nodes are participating. However, too many cooperative cognitive users (sensing nodes) will cause vast communication overhead. A proposed review method addressed this problem by examining the observed values of perceived nodes and only allowing nodes containing sufficient information to send their decision values (0 or 1) to the fusion center. This method reduces communication overhead but also reduces sensing performance. Aiming at the excessive overhead created by equal gain fusion, a double threshold method is used to perform cooperative spectrum sensing in which each node adopts double threshold detection and sends the detected value directly to the fusion center, which then makes the judgment. Combined with the judgments of each node and its own judgment, the fusion center makes two judgments to determine whether the primary user exists. This employs a combination of soft and hard fusion methods, but performs two operations in the fusion center, thus increasing computational power. Furthermore, a hierarchical cooperative spectrum detection method has been proposed to solve the problem of excessive cooperative sensing overhead. Here, when nodal observation values are between two thresholds, the region between these two thresholds is evenly divided into four parts; four different regions are thus quantized by 2 bits. The sensing nodes then send 2 bits of information to the fusion center. When compared with the equal gain fusion method, this reduces communication overhead. However, the sensing performance of hard fusion decreases.

Spectrum sensing performance directly affects the throughput of cognitive users and multi-node cooperative spectrum sensing is a common method to improve the performance of spectrum sensing. However, when multi-node participates in cooperative sensing, the sensing data will increase greatly, and the fusion center cannot process a large number of data in time, which will cause delayed decision, which will affect the security of the main user or the throughput of cognitive users. In order to make a decision in time, a large number of data in the fusion center needs to be processed quickly, which requires the selection of some node data to reduce the number of processed data. Therefore, in order to achieve the real-time spectrum sensing, the fusion center needs to quickly fuse the data of each node. In this article, a fast algorithm of big data fusion is proposed to improve the real-time performance of the global decision. The algorithm improves the computing speed by reducing repeated computation. The reinforcement learning mechanism is used to mark the processed data. When the same environment parameter appears, the fusion center can directly call the nodes under the parameter environment, without having to conduct the sensing operation again. This greatly reduces the amount of data processed and improves the data processing efficiency of the fusion center. Experimental results show that the algorithm in this article can reduce the computation time while improving the perceived performance.

System model

This study designed an analog cognitive radio system consisting of a primary user (PU) and 16 nodes (cognitive users). Each node communicates with the fusion center through a channel, while the fusion center fuses information from each node to determine whether the primary user channel is idle. The system model is illustrated in Figure 1.

Derivation of the optimum local detection threshold

For spectrum sensing, every SU independently performs an energy detection process. The signal received by an SU is determined as follows

\[ x(m) = hs(m) + v(m), m = 0, 1, \ldots, M - 1 \]  

where \( s(m) \) is a PU signal, \( v(m) \) is the additive white Gaussian noise with zero mean and variance \( \sigma_v^2 \), \( m \) represents the serial number of the sampling point, \( M \) is the number of samples, and \( h \) is channel gain. Suppose PU is absent (i.e. \( s(m) = 0 \)); the hypothesis of free channel is then denoted by \( H_0 \), whereas the hypothesis of busy channel is denoted by \( H_1 \), as follows

\[ H_0: x(m) = v(m), m = 0, 1, \ldots, M - 1 \]  
\[ H_1: x(m) = hs(m) + v(m), m = 0, 1, \ldots, M - 1 \]  

Assuming \( E \) is the average collected energy of an SU and it is expressed as follows

\[ E = \frac{1}{M} \sum_{m=0}^{M-1} |x(m)|^2 \]  

If each SU can make its local decision according to single threshold \( \lambda_o \) with probabilities of detection \( P_d \) and false alarm \( P_f \), the equation is as follows

\[ P_d = P\{E \geq \lambda_o|H_1\} = Q \left( \frac{\lambda_o - (\sigma^2_v + \sigma^2_s)}{(\sigma^2_s + \sigma^2_v)/\sqrt{M/2}} \right) \]  
\[ P_f = P\{E \geq \lambda_o|H_0\} = Q \left( \frac{\lambda_o - \sigma^2_v}{\sigma^2_v/\sqrt{M/2}} \right) \]
where $s^2$ is signal power and $v^2$ is noise power; the complementary cumulative distribution function $Q(x)$ will be described as follows:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{-t^2/2} dt$$  \hspace{1cm} (6)$$

Assuming the presence probability of a PU is $P(H_1) = \beta$, $0 < \beta < 1$ and the absence probability of the PU is $P(H_0) = 1 - \beta$, then the probability of error detection ($P_e$) is as follows:

$$P_e = P(H_1)(1 - P_d) + P(H_0)P_f = \beta(1 - P_d) + (1 - \beta)P_f$$  \hspace{1cm} (7)$$

where $P_e$ is the quadratic function of the $\lambda_0$, we can derive $\frac{\partial^2 P_e}{\partial \lambda_0^2} > 0$. The optimal threshold $\lambda_0$ is obtained by $\frac{\partial P_e}{\partial \lambda_0} = 0$ and it is expressed as follows:

$$\lambda_0 = \frac{\sigma^2_v(1 + \gamma)}{2 + \gamma} \left[ 1 + \sqrt{1 + \frac{4(2 + \gamma)}{M\gamma} \ln \left( \frac{1 - \beta(1 + \gamma)}{\beta} \right) } \right]$$  \hspace{1cm} (8)$$

It is easy to obtain optimal threshold $\lambda_0$ if signal power $\sigma^2_s$, noise power $\sigma^2_v$ and the SNR of the SU’s receiving terminal are known.

**Estimating the optimal threshold**

If a sensing cycle sampling point is $M = 2^L$ and $L$ is a positive integer, $M$ can be divided into two following equal sections: (1) the previous $M/2$ sampling points can be expressed with $x_1$; (2) the later $M/2$ sampling points can be expressed with $x_2$. $x_1$ and $x_2$ are given as follows:

$$x_1 = (x(0), x(1), \ldots, x(M_1 - 1))$$  \hspace{1cm} (9)$$

$$x_2 = (x(M_1), x(M_1 + 1), \ldots, x(M - 1))$$  \hspace{1cm} (10)$$

where $M_1 = M/2$, the average energy of each section is expressed as follows:

$$E_1 = \frac{1}{M_1} \sum_{m = 0}^{M_1 - 1} |x(m)|^2$$  \hspace{1cm} (11)$$

$$E_2 = \frac{1}{M_1} \sum_{m = M_1}^{M - 1} |x(m)|^2$$  \hspace{1cm} (12)$$

If $E_1 < E_2$, $x_1$ denotes AWGN, $x_2$ denotes the sum of signal and AWGN (i.e. $E_1 = \sigma^2_v$), $E_2 = \sigma^2_s + \sigma^2_v$. If $E_1 > E_2$, $x_2$ denotes AWGN.

Let $E_1$ denote estimated noise power $\hat{\sigma}^2_v$ and $E_2$ denote the estimated sum of signal and noise powers $\hat{\sigma}^2_s + \hat{\sigma}^2_v$; estimated signal power is then $\hat{\sigma}^2_s = E_2 - E_1$. 

---

**Figure 1.** Simulation scenario for spectrum sensing.
As such, the estimated SNR of received signal $x(m)$ is expressed as follows
\[
\hat{\gamma} = \frac{\hat{\sigma}_v^2}{\hat{\sigma}_y^2} = \frac{E_2 - E_1}{E_1} \tag{13}
\]

The estimation of optimal thresholds $\lambda_o$ is expressed as follows
\[
\hat{\lambda}_o = \frac{\hat{\sigma}_v^2(1 + \hat{\gamma})}{2 + \hat{\gamma}} \left[ 1 + \frac{1 + 4(1 + \hat{\gamma})\ln \left[ \frac{(1 - \beta)(1 + \hat{\gamma})}{\beta} \right]}{M \cdot \gamma} \right] \geq 0 \tag{14}
\]

Here, it should be noted that the conditions for the solution of equation (2) should be satisfied according to the following equation
\[
1 + \frac{4(2 + \gamma)\ln \left[ \frac{(1 - \beta)(1 + \gamma)}{\beta} \right]}{M \cdot \gamma} \geq 0 \tag{15}
\]

Transform equation (15) type to the following
\[
(1 + \gamma) \cdot e^{\frac{\gamma \cdot \beta}{1 - \beta}} \geq \frac{\beta}{1 - \beta} \tag{16}
\]

When the number of sampling points is $M = 1024$ and the SNR $\gamma = -20$ dB, after mathematical derivation, the value of $\beta$ must satisfy $\beta \in (0, 0.731)$, and the equation (16) can be established. The range of $\beta$ values can be extended if the number of sampling points increases. When $M = 10^5$ and $\gamma = -30$ dB, then $\beta \in (0, 0.999)$. This means that during a sensing cycle and when there is a sufficient number of sampling points (even in the case of low SNR and when the probability of the primary user signal is uncertain), equation (15) is established and there are solutions to equation (14).

**Adaptive double energy thresholds**

To avoid error judgments due to SNR variations in a received end, lower threshold $\lambda_l$ and upper threshold $\lambda_h$ are set based on optimal threshold $\lambda_o$ (Figure 2). In Figure 2, $d$ is the distance between $\lambda_o$ and lower threshold $\lambda_l$ or $\lambda_i$ and upper threshold $\lambda_h$. The following is thus obtained
\[
\begin{align*}
\lambda_l &= \lambda_o - d \\
\lambda_h &= \lambda_o + d
\end{align*} \tag{17}
\]

For weak PU signal detection, threshold $\lambda_o$ should be decreased. However, it should not be lower than $\sigma_v^2$. Noise would otherwise be detected as a PU signal. However, $\lambda_o$ should not be larger than $\sigma_v^2 + \sigma_s^2$. The PU signal would otherwise miss detection. To assure a lower probability of false alarm and a higher probability of detection, we thus put a limiting range on threshold $\lambda_o$, as follows
\[
\sigma_v^2 \leq \lambda_o - d \tag{18}
\]
\[
\lambda_o + d \leq \sigma_v^2 + \sigma_s^2 \tag{19}
\]

Here, equations (18) and (19) gives the following
\[
0 \leq d \leq \frac{1}{2} \sigma_v^2 \tag{20}
\]

Different from conventional double threshold settings, we introduced control parameter $\varepsilon$ to accurately fine-tune the double thresholds and define the following
\[
d = \varepsilon \sigma_v^2 \tag{21}
\]

According to equation (20), $\varepsilon$ satisfies the following
\[
0 \leq \varepsilon \leq 0.5 \tag{22}
\]

When $\varepsilon = 0$, it is equivalent to a single threshold case. Thus, $\lambda_l$ and $\lambda_h$ can be rewritten as follows
\[
\begin{align*}
\lambda_l &= \lambda_o - \varepsilon \sigma_v^2 \\
\lambda_h &= \lambda_o + \varepsilon \sigma_v^2
\end{align*} \tag{23}
\]

Parameter $\varepsilon$ is an impact factor for double thresholds.

**Quantization and coding based on adaptive double energy thresholds**

We considered the cognitive radio network as shown in Figure 1. Here, each node communicates with the fusion center. This study assumed that the channel between the node and fusion center was perfect. $\lambda_{l,i}$, $\lambda_{h,i}$, $\lambda_{o,i}$ and $w_i(i = 0, 1, \ldots, N - 1)$ denote the upper threshold, lower threshold, optimal threshold, and weight of node $E_i$, respectively.

**Calculating bode weights**

Assume $E_i$ is the average energy collected by $i$th node. First, if $E_i$ is not lower than upper threshold $\lambda_{h,i}$ and its weight equals 1, then the SU decides that the PU is present. Next, if $E_i$ is not lower than lower threshold $\lambda_{l,i}$ and its weight equals 0, then the node decides that the PU is absent. Finally, if $E_i$ is located between $\lambda_{h,i}$
Calculation is expressed as follows: weight is equal to the normalized result. The weight will change when the two nodal thresholds are altered. As such, it is highly important to select an optimal $\varepsilon$ to establish two proper thresholds, thus improving sensing performance. A grid search is conducted to obtain the best parameters for $\varepsilon$ in part 4. These are memorized through reinforcement learning strategies to obtain better sensing performance and higher sensing efficiency.

**Cooperative spectrum sensing based on quantization and coding**

After an SU obtains weight and encodes weight $w_i$, where $i = 0, 1, \ldots, N - 1$, the coding rules are as follows: First, if $w_i = 1$, is encoded as 1, and expressed as $C_{i, 1} = 1$, then this will be sent to the fusion center, which denotes that an SU transmitted 1 bit of data and consumed 1 unit of energy (e.g. $e_1 = 1$). Next, if $w_i = 0$, is encoded as 0 and expressed as $C_{i, 2} = 0$, then this also denotes that an SU sent 1 bit of data to the fusion center and consumed 1 unit of energy. Finally, if $0 < w_i < 1$, $C_{i, 3} = d_3 d_2 d_1$, where $d_i$ are integers (1 or 0) and $j = 1, 2, 3$, when $2 \times w_i = 1$, $d_1 = 1$; otherwise $d_1 = 0$, when $2 \times (2 \times w_i - d_2) = 1$, $d_2 = 1$; otherwise $d_2 = 0$, and when $2 \times (2 \times w_i - d_1 - d_2) = 1$, $d_1 = 1$; otherwise $d_1 = 0$, which denotes that an SU sent 3 bits of data to the fusion center and consumed 3 units of energy.

The fusion center will decode for $C_i$ after receiving sensing results from an SU. Here, the decoding rules can be described as one of three types. First, if $C_i = 1$, then the decoding data are expressed as $D_i = 1$. Second, if $C_i = 0$, then the decoding data are expressed as $D_i = 0$. Third, if $C_i = d_3 d_2 d_1$, then the decoding data are expressed as

$$D_i = \frac{d_3 \times 2^2 + d_2 \times 2^1 + d_1 \times 2^0}{8}$$

The decoding rules can thus be expressed as follows

$$\begin{cases} 
C_i = 1, & D_i = 1, \\
C_i = 0, & D_i = 0, \\
C_i = d_3 d_2 d_1, & D_i = \frac{d_3 \times 2^2 + d_2 \times 2^1 + d_1 \times 2^0}{8}.
\end{cases}$$

where $0.125 \leq D_{i, 3} \leq 0.875$, and its resolution ratio is 0.125; this can reflect that a node made a contribution for the cooperative spectrum sensing to match its weight.

The fusion center will use majority-rule fusion after completely decoding all data received from all SUs. The fusion expressed is as follows

$$R = \sum_i D_i, \ i = 0, 1, \ldots, N - 1$$

where $R$ is the fusion result, $i$ is the index of the nodes, and $D_i$ is the decoding data for the node. The fusion center compares $R$ and $N/2$ to decide whether there is

![](image.png)

**Figure 3.** Assigned weights and the cooperative spectrum sensing algorithm.
The trained parameters included SNR (to train parameters in order to improve search efficiency. The expression of this decision is as follows

$$\Re = \sum_{i=0}^{N-1} D_i \left\{ \begin{array}{ll} \frac{\omega}{2} & H_1 \\ \frac{\omega}{2} & H_0 \end{array} \right.$$  (27)

The code algorithm summarizes the coding-based cooperative spectrum sensing.

**Coding-based cooperative spectrum sensing algorithm**

1. Initialization
calculate $E_i$, $i = 0, 1, \ldots, N - 1$, estimate $\sigma_i^2$;
set $N, M, \lambda_0, \lambda_1$, calculate $\lambda_i$; and $\lambda_k$, according to (23);
2. Calculate the weights of all SUs
if $E_i \geq \lambda_i$, $w_i = 1$;
else if $E_i \leq \lambda_i, w_i = 0$;
else $w_i = \frac{E_i}{\lambda_i}$
3. Encode the weights for all SUs
if $w_i = 1; C_i = 1$;
else if $w_i = 0; C_i = 0$;
else if $2 \cdot w_i \equiv 1; d_3 = 1$;
else $d_3 = 0$;
if $2 \cdot \left(2 \cdot w_i - d_3\right) \equiv 1; d_2 = 1$;
else $d_2 = 0$;
if $2 \cdot \left(2 \cdot w_i - d_3\right) - d_2 \equiv 1; d_1 = 1$;
else $d_1 = 0$;
$C_i = d_2d_3d_1$;
4. Fusion center decoding for received data from $N$ SUs
if $C_i = 1; D_i = 1$;
else if $C_i = 0; D_i = 0$;
else $D_i = 4d_2^2 + 4d_3^2 + d_1^2$;
5. Fusion center fuses decoding data on the basis of (26).
6. Fusion center makes decision about PU according to (27).
7. Sensing end

**PU: primary user.**

For the note code algorithm, the transmission of all SUs combines 1 bit sent and 3 bits sent. As such, the algorithm can improve sensing performance while reducing communication overhead.

**Reinforcement learning based on the grid search algorithm**

**Grid search algorithm**

A grid search can be used to obtain optimal $\varepsilon$ according to the feedback of global $P_d$. We conducted a grid search to train parameters in order to improve search efficiency. The trained parameters included SNR ($-25$ to $0$ dB with step $1$ dB) and $\varepsilon$ ($0$ to $0.5$ with step $0.05$). These were saved as a prior knowledge to a knowledge base. Spectrum sensing will then directly invoke optimal $\varepsilon$ under an SNR according to the prior knowledge. Figure 4 shows the grid search algorithm used for parameter training.

In Figure 4, $k$ and $j$ stand for SNR and control parameter $\varepsilon$, respectively; $(k_n, f_n)$ is a prior knowledge group, while $j_n$ is optimal $\varepsilon$ under $k_n$, $n = 1, 2, \ldots, 26$. If an SNR is newly appearing, then this algorithm will immediately train new parameters; these will be saved as prior knowledge in the knowledge base.

The grid search algorithm process is described as follows

1. When an SNR $\kappa_i$ appears, the fusion center will conduct a real-time search to find the optimal $\varepsilon_i$ and obtain the highest $P_d$. It will then proceed to step 3, where $\kappa_i$ is the $i$th newly appearing SNR and $\varepsilon_i$ is the corresponding optimal control parameter. These results will be output when $P_d$ and $\varepsilon_i$ are returned. Furthermore, $\varepsilon_i$ and $\kappa_i$ will become a prior knowledge couple that the fusion center will then learn, for example

$$\varphi = f(\varepsilon_i, \kappa_i)$$  (28)

where $i$ is a positive integer, $\varphi$ is a storage library, and $f(\cdot)$ is a learn function.

2. When SNR $\kappa_i$ is not newly appearing, the fusion center will utilize learned knowledge to directly select optimal $\varepsilon_i$; for example

$$\varepsilon_i = f^{-1}(\kappa_i, \varphi)$$  (29)

where $f^{-1}(\cdot)$ is a function reading knowledge from the storage library.

3. Under SNR $\kappa_i$, the range of parameter $\varepsilon_i$ is divided into 10 equal grids by 11 grid points, $\varepsilon_{i,j}$ is the $j$th grid point $j \in \{0, 1, 2, \ldots, 10\}$. $\Delta \epsilon = 0.05$ is the searching step. The searching process is from $\varepsilon_{i,0}$ to $\varepsilon_{i,10}$ with step $\Delta \epsilon$.

4. $P_{d,j}$ is $P_d$ of the $j$th grid point, when a $P_{d,j} = 1$, $j \in \{0, 1, 2, \ldots, 10\}$, stopping search, the $P_{d,j}$ and corresponding $\varepsilon_{i,j}$ are returned to 1. The search otherwise continues.

5. When the real-time search has been finished, highest probability $P_{d,\text{max}}$ and corresponding optimal parameter $\varepsilon_{i,j}$ are returned to step 1, as follows

$$\left\{ P_{d,\text{max}} = \max\{P_{d,0}, P_{d,1}, \ldots, P_{d,10}\} \right\}$$

$$\left\{ \varepsilon_{i,j} = f_i^{-1}(P_{d,\text{max}}) \right\}$$  (30)

where $f_i^{-1}(\cdot)$ is a function seeking $\varepsilon_{i,j}$ through $P_{d,\text{max}}$.

6. End
Reinforcement learning based on the grid search algorithm

The learning process is as follows:

1. After the fusion center finishes executing the grid search, the obtained grid coordinates are represented as follows:

\[
P_A = \begin{bmatrix}
p_{d,0} & p_{d,1} & \cdots & p_{d,40} \\
p_{a,0} & p_{a,1} & \cdots & p_{a,40} \\
\vdots & \vdots & \ddots & \vdots \\
p_{e,0} & p_{e,1} & \cdots & p_{e,40}
\end{bmatrix}
\]

The first column in matrix \( A \) represents the value of control parameter \( e \), while the second column represents the value of signal-to-noise ratio \( a \). Each row represents the optimal control parameters found by executing the grid search algorithm in a specific radio environment.

2. The fusion center sends matrix \( A \) and the matrix description to cognitive users.

3. The cognitive user can memorize the data from matrix \( A \); the local detection threshold is set according to the data and is the best to be preserved. When the radio environment is consistent with the memory, then the optimal threshold in this environment can be directly invoked to perform the next spectrum sensing process.

4. In the case of a new radio environment, the fusion center must alter the range of \( a \) values and re-execute the grid search algorithm (i.e. execute steps 1-3 again).

Experiments and evaluation

This study designed three groups of Monte Carlo simulation experiments to evaluate the performance of the cooperative spectrum sensing method, as follows: (1) a comparison of detection probabilities (\( P_d \)) between traditional fusion methods and the new fusion method presented in this article, (2) under the same conditions, a comparison of probability of error (\( P_e \)) between the used grid search algorithm and those unused in this article, (3) under the same conditions, a comparison of sensing speed between reinforcement learning and non-reinforcement learning, (4) verification of fast fusion algorithm, compare the processing time of using fast fusion algorithm with that of not using fast fusion algorithm. The Monte Carlo simulations were conducted under conditions involving path loss and additive Gaussian white noise.

The simulation experiments set the PU signal to a BPSK, bandwidth to 100 kHz, and sensing duration to 100 ms. The PU was placed in the center of a 1000 x 1000 m square and surrounded by 16 evenly distributed sensing nodes. The simulation scenario is shown in Figure 1. The probability of setting the channel of PU occupancy was \( \beta = 50\% \), while the transmitting power of the PU signal was 100 mW. Each node sampled 20 points; the noise power range was set to between 0 and 2 dB, while the path-loss exponent was 2.7, the standard deviation of the shadow was 5 dB, and the mean of the multipath Rayleigh fading was 1.32–34

Figure 5 illustrates a comparison of detection probabilities (\( P_d \)) between the traditional fusion methods and the new fusion method proposed in this article. The traditional methods used for comparison included AND, OR and Majority fusion methods. As seen in Figure 5, the detection probability (\( P_d \)) obtained by the
new fusion method was higher than that obtained by traditional methods. This is more obvious in cases involving low SNR because each sensing node uses double the threshold energy detection and calculates weights according to the signal energy received by itself. Such nodes can then make appropriate contributions to the spectrum sensing process based on their own weights. As such, the new fusion method more accurately reflects the actual roles of each node when compared to traditional methods.

Figure 6 illustrates the comparison of probability of error ($P_e$) between the grid search algorithm used in this study and other algorithms (i.e. fixed-single and fixed-double threshold algorithms). As seen in Figure 6, the grid search algorithm exhibited the lowest error probability during spectrum sensing. This is because the best detection threshold can be obtained through the grid search algorithm in any radio environment. The other two algorithms have fixed thresholds and can therefore not adapt to noise fluctuations. While the probability of error ($P_e$) increases when SNR decreases in all cases, the grid search algorithm results in the smallest increase.

Figure 7 shows a sensing speed comparison between reinforcement and non-reinforcement learning. As seen, reinforcement learning takes less sensing time than non-reinforcement learning under the same SNR conditions. This is because reinforcement learning can directly invoke detection thresholds in the same environment from the repository. If reinforcement learning is not used, then every spectrum sensing procedure requires a grid search algorithm to find the optimal threshold; this requires more sensing time. Sensing time decreases when SNR increases because the radio environment is simpler; with an increased signal-to-noise ratio, less information is stored and judgments are easier to make.

In order to verification of fast fusion algorithm, compare the processing time of using fast fusion algorithm with that of not using fast fusion algorithm. The experiments are all under the same number of nodes. In order to highlight the advantages of the fast algorithm proposed in this article, observe the data processing time under different node numbers. Table 1 shows the processing time at different nodes. The processing environment is MATLAB 7.0, and the computer configuration is Intel (R) Core (TM) i5-8500 CPU at 3.00 GHz, RAM is 8 GB, and 64-bit operation system.

It can be seen from Table 1 that the fast algorithm used by the fusion center can effectively reduce the data processing time, and the average processing time can be reduced by 18%. When the number of nodes is more, the advantage of fast algorithm in dealing with big data is more obvious. When the number of nodes is more than 30, the time of fast algorithm in dealing with data is less than that of not using fast algorithm in dealing with 25 nodes, which can be the advantage of fast algorithm in this article.

**Conclusion**

This article studies a new perceptual data fusion algorithm, which can process the perceptual data of each node quickly without delay. In the cognitive radio network, different nodes have different perception data due to different geographical location, and the contribution of each node's perception data to cooperative perception is also different. The fusion center uses reinforcement learning mechanism to select cooperation nodes by identifying the sensing performance of node,
which can reduce the processing data to a certain extent, and enable the fusion center to process quickly the data sent by each node will not cause decision delay. This greatly improves the throughput of cognitive users while protecting the primary users. The experimental results show that the big data fast fusion algorithm in this article can effectively reduce the data processing time, The average processing time of using fast algorithm is 18% less than that of not using fast algorithm. When the number of nodes is more than 30, the time of fast algorithm in dealing with data is less than that of not using fast algorithm in dealing with 25 nodes, which

Figure 6. A comparison of probability of error between the grid search algorithm and others.

Figure 7. A sensing speed comparison between reinforcement and non-reinforcement learning.

Table 1. compare the processing time of using fast fusion algorithm with that of not using fast fusion algorithm.

| Number of nodes | Processing time using fast fusion algorithm (ms) | Processing time do not using fast fusion algorithm (ms) |
|----------------|-----------------------------------------------|-------------------------------------------------------|
| 5              | 0.32                                          | 0.38                                                  |
| 10             | 0.56                                          | 0.64                                                  |
| 15             | 0.72                                          | 0.79                                                  |
| 20             | 0.84                                          | 0.92                                                  |
| 25             | 0.95                                          | 1.04                                                  |
| 30             | 1.02                                          | 1.16                                                  |
can be the advantage of fast algorithm in this article. Furthermore, the algorithm in this article can reduce the processing time of node data and improve the sensing performance at the same time and increase the throughput of cognitive users, which is of great significance. However, at present, only the fast algorithm of big data is implemented in the fusion center, but not the energy-saving algorithm in the node itself, which is the follow-up research goal.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This article was supported by the Natural Science Foundation of Hunan Province, China (grant nos 2019JJ40097 and 2019JJ40096), the Key Research and Development Projects of Science and Technology Department of Hunan Province (grant no. 2017NK2390), the Research Foundation of Education of Science and Technology Department of Hunan Province (grant no. 17B107), the Research Foundation of Science and Technology Bureau of Yongzhou City, China (nos 2019YJK308 and 2019YJK10), and the construct program of applied characteristic discipline at the Hunan University of Science and Engineering. The authors would like to thank Editage (www.editage.com) for English language editing.

ORCID iD
Tangsen Huang https://orcid.org/0000-0002-4426-2622

References
1. Jing Z and Huang Z. Blind recognition of binary BCH codes for cognitive radios. Math Prob Eng 2016; 2016: 1–6.
2. Xiong Z, Li Y, Yao D, et al. Random, persistent and adaptive spectrum sensing strategies for multiband spectrum sensing in cognitive radio networks with secondary user hardware limitation. IEEE Access 2017; 5: 14854–14866.
3. Xiong T, Yao Y, Ren Y, et al. Multiband spectrum sensing in cognitive radio networks with secondary user hardware limitation: random and adaptive spectrum sensing strategies. IEEE Tran Wirel Commun 2018; 17: 3018–3029.
4. Jun K, Min J, Gu X, et al. Adaptive weight collaborative complementary learning for robust visual tracking. KSII Tran Inte Info Syst 2019; 13: 305–326.
5. Ni L, Da X, Hu H, et al. Energy efficiency design for secure MISO cognitive radio network based on a non-linear EH model. Math Prob Eng 2018; 2018: 1–7.
6. Liu M, Zhao N, Li J, et al. Spectrum sensing based on maximum generalized correntropy under symmetric alpha stable noise. IEEE Tran Vehic Technol 2019; 68: 10262–10266.
7. Chen H, Zhou M, Xie L, et al. Joint spectrum sensing and resource allocation scheme in cognitive radio networks with spectrum sensing data falsification attack. IEEE Tran Vehic Technol 2016; 65: 9181–9191.
8. Park J, Pawelczak P and Cabric D. Performance of joint spectrum sensing and MAC algorithms for multichannel opportunistic spectrum access Ad Hoc networks. IEEE Tran Mob Comput 2011; 10: 1011–1027.
9. Khoshkholgh MG, Navaie K and Yanikomeroglu H. Optimal design of the spectrum sensing parameters in the overlay spectrum sharing. IEEE Tran Mob Comput 2014; 13: 2071–2085.
10. Zhang X, Ma Y, Gao Y, et al. Autonomous compressive-sensing-augmented spectrum sensing. IEEE Tran Vehic Technol 2018; 67: 6970–6980.
11. Yang P, Tsay S-C, Wei H, et al. Remote sensing of cirrus optical and microphysical properties from ground-based infrared radiometric Measurements-part I: a new retrieval method based on microwindow spectral signature. IEEE Geosci Remo Sens Lett 2005; 2: 128–131.
12. Zhang Z, Wen X, Xu H, et al. Sensing nodes selective fusion scheme of spectrum sensing in spectrum-heterogeneous cognitive wireless sensor networks. IEEE Sens J 2018; 18: 436–445.
13. Sun D, Song T, Gu B, et al. Spectrum sensing and the utilization of spectrum opportunity tradeoff in cognitive radio network. IEEE Commun Lett 2016; 20: 2442–2445.
14. Li B, Li S, Nallanathan A, et al. Deep sensing for next-generation dynamic spectrum sharing: more than detecting the occupancy state of primary spectrum. IEEE Tran Commun 2015; 63: 2442–2457.
15. So J and Srikant R. Improving channel utilization via cooperative spectrum sensing with opportunistic feedback in cognitive radio networks. IEEE Commun Lett 2015; 19: 1065–1068.
16. Sharifi AA and Musevi Niya MJ. Defense against SSDF attack in cognitive radio networks: attack-aware collaborative spectrum sensing approach. IEEE Commun Lett 2016; 20: 93–96.
17. Feng G, Chen W and Cao Z. “A joint PHY-MAC spectrum sensing algorithm exploiting sequential detection. IEEE Sig Proces Lett 2010; 17: 703–706.
18. Sun B, Chen Q, Xu X, et al. Permuted&filtered spectrum compressive sensing. IEEE Sig Proces Lett 2013; 20: 685–688.
19. Xiong T, Li H, Qi P, et al. Predecision for wideband spectrum sensing with sub-Nyquist sampling. IEEE Tran Vehic Technol 2017; 66: 6908–6920.
20. Huang X, Hu F, Wu J, et al. Intelligent cooperative spectrum sensing via hierarchical dirichlet process in cognitive radio networks. IEEE J Selec Areas Commun 2015; 33: 771–787.
21. Ismail N and Mohamad M. Review on energy efficient opportunistic routing protocol for underwater wireless sensor networks. KSII Tran Inte Info Syst 2018; 12: 3064–3094.
22. Chen Y, Zheng Z, Hou Y, et al. Energy efficient design for OFDM-based underlay cognitive radio networks. Math Prob Eng 2014; 2014: 1–8.
23. Chen N, Zhang X and Su S. A joint scheduling and beamforming scheme for RoF-aided MC-SSN. *IEEE Access* 2019; 7: 29245–29252.

24. Zhao Y, Lin H, Zhou C, et al. Cascaded Mach–Zehnder interferometers with Vernier effect for gas pressure sensing. *IEEE Photon Technol Lett* 2019; 31: 591–594.

25. Alibeigi M and Taherpour A. Optimisation of secrecy rate in cooperative device to device communications underlaying cellular networks. *IET Commun* 2019; 13: 512–519.

26. Luo Z, Li C and Zhu L. Full-duplex cognitive radio using guided independent component analysis and cumulant criterion. *IEEE Access* 2019; 7: 27065–27074.

27. Yazıcıgil RT, Haque T, Kinget PR, et al. Taking compressive sensing to the hardware level: breaking fundamental radio-frequency hardware performance tradeoffs. *IEEE Sig Proces Magaz* 2019; 36: 81–100.

28. Li A and Han G. Full-duplex-based control channel establishment for cognitive internet of things. *IEEE Commun Magaz* 2019; 57: 70–75.

29. Huang T and Li J. A weighted cooperative spectrum sensing scheme based on dynamic double energy thresholds in cognitive radio networks. In: *Proceedings of IEEE Global High Tech Congress on Electronics (GHTCE)*-Shenzhen, China, 17–19 November 2013, pp.201-204. New York: IEEE.

30. Lu W, Hu S, Liu X, et al. Incentive mechanism based cooperative spectrum sharing for OFDM cognitive IoT network. *IEEE Tran Net Sci Eng*. Epub ahead of print 16 May 2019. DOI: 10.1109/TNSE.2019.2917071.

31. Salah M, Omer OA and Mohammed US. Spectral efficiency enhancement based on sparsely indexed modulation for green radio communication. *IEEE Access* 2019; 7: 31913–31925.

32. Chen A, Shi Z and Xiong J. Generalized real-valued weighted covariance-based detection methods for cognitive radio networks with correlated multiple antennas. *IEEE Access* 2019; 7: 34373–34382.

33. Bhowmick A, Roy SD and Kundu S. Performance of spectrum sensing scheme using double threshold energy detection in the presence of sensor noise. *Int J Energy Info Commun* 2012; 3: 75–84.

34. Lee H, Noda K, Mizuno Y, et al. Distributed temperature sensing based on slope-assisted Brillouin optical correlation-domain reflectometry with over 10 km measurement range. *Electron Lett* 2019; 55: 276–278.