Deep Learning Based Energy Efficiency Optimization for Distributed Cooperative Spectrum Sensing

Haibo He and He Jiang

Abstract

Deep learning has achieved remarkable breakthroughs in the past decade across a wide range of application domains, such as computer games, natural language processing, pattern recognition, and medical diagnosis, to name a few. In this article, we investigate the application of deep learning techniques for wireless communication systems with a focus on energy efficiency optimization for distributed cooperative spectrum sensing. With the continuous development of today's technologies and user demands, wireless communication systems have become larger and more complex than ever, which introduces many critical challenges that the traditional approaches can no longer handle. We envision that deep learning based approaches will play a pivotal role in addressing many such challenges in the next-generation wireless communication systems. In this article, we focus on cognitive radio, a promising technology to improve spectrum efficiency, and develop deep learning techniques to optimize its spectrum sensing process. Specifically, we investigate the energy efficiency of distributed cooperative sensing by formulating it as a combinatorial optimization problem. Based on this formulation, we develop a deep learning framework by integrating graph neural network and reinforcement learning to improve the overall system energy efficiency. Simulation studies under different network scales demonstrate the effectiveness of our proposed approach.

Introduction

Over the past decade, we have witnessed tremendous technology development and societal benefits for various wireless devices, such as smartphones, smart wearable devices, and the Internet of Things (IoT), among others. To meet the traffic demands from these devices and provide a higher quality of experience (QoE) for users, wireless communication systems have exploited more and more radio spectrum. Although the capacity of the radio spectrum is fairly large, the shortage problem is imminent due to the dramatic proliferation of wireless devices. Currently, most spectrum resources are statically allocated to licensed users. However, it has been witnessed that a considerable part of the spectrum bands is underutilized, which is a big extravagance of this scarce natural resource. In this case, an efficient spectrum management policy is essential for the next-generation wireless communication network.

Cognitive radio is one of the most promising solutions to boost the radio spectrum efficiency [1]. In a cognitive radio network, the spectrum bands are allowed to be accessed by both licensed and unlicensed users, which are also called primary users (PUs) and secondary users (SUs), respectively. The PUs are empowered with priority to use the spectrum, whereas the SUs are permitted to access the spectrum only if the PUs are inactive. Under this regulation, spectrum sensing plays a crucial role as the detection procedure for the PUs' activities. Inherently, spectrum sensing belongs to the signal detection realm. Various classic detection techniques can be applied to identify the presence of the PUs. However, the performance of a single detector can be severely degraded in the real-world environment because of shadowing, multipath fading, and hidden terminal issues [2]. In this case, *cooperative spectrum sensing* is proposed to improve the detection performance by taking advantage of the spatial diversity of the distributed sensors. According to the communication topology, the cooperative sensing paradigms fall into three categories: *centralized*, *relay-assisted*, and *distributed* [2]. For the first two categories, a fusion center is necessary to collect the sensing data from the sensors and make the detection decision with a fusion rule. In this way, the fusion center can suffer from severe traffic overload. In contrast, distributed cooperative sensing accomplishes the collaboration through a local iterative consensus algorithm, where each sensor only processes the data from its neighbors [3, 4].

Regardless of which cooperative sensing paradigm is used, the signal detection devices are indispensable to collect and process the sensing data. Traditionally, the spectrum sensing module is integrated with every SU, which will increase the manufacturing cost of wireless devices. As a result, several researchers advocate separating this functionality from the SUs and constructing a specialized sensor network to provide the sensing service [5]. The sensor network is supposed to operate continuously to provide a real-time spectrum map for SUs. Consequently, the energy consumption can be considerable over time. This issue could be more severe for a distributed cooperative sensing system since it relies on an

Digital Object Identifier:
10.1109/MWC.2019.1800397

Haibo He (Corresponding author) and He Jiang are with the University of Rhode Island.

1536-1284/19/$25.00 © 2019 IEEE

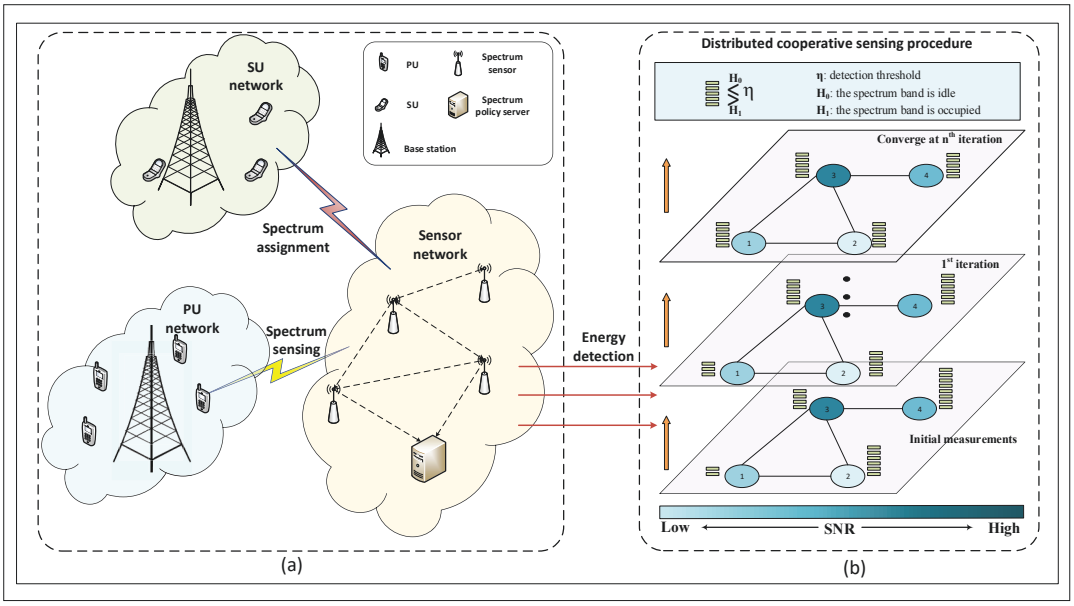

FIGURE 1. A sensor-aided cognitive radio network with energy detection based distributed cooperative sensing: a) sensor-aided cognitive radio network; b) energy detection based distributed cooperative sensing. In the energy detection illustration, the node color indicates the local SNR level of the sensor measurement. Each node recursively communicates with its neighbor to fuse these measurements with a consensus protocol until convergence. The converged result is then compared to a pre-set threshold to infer whether the target spectrum band is idle or occupied.

iterative algorithm. Therefore, energy efficiency is of great importance for the sensor network. The research community has extensively explored the energy efficiency problem for fusion center based cooperative sensing [5], while to the best of our knowledge, limited research has been conducted on distributed paradigms. Moreover, due to the difference of the collaboration mechanism, the methodologies proposed for fusion center based cooperative sensing are not applicable to the distributed counterpart. Under this circumstance, we specifically develop an optimization framework to search the energy-efficient distributed sensing scheme. We consider the case where the cooperative sensors are identical and experience different signal-to-noise ratio (SNR). Our objective is to find a minimum subset of sensors that not only has an effective topology for the distributed sensing algorithm but also satisfies the detection performance requirement.

It is clear that searching the optimal solution is a combinatorial optimization problem, and the exact optimal solution of such a problem, especially for a large-scale network, is almost impossible to obtain as the state space of the solution grows exponentially with the sensor quantity. Fortunately, the recent development of artificial intelligence (AI) methodologies have brought significant breakthroughs for such complex problems in many areas. Several combinatorial optimization problems have been studied through AI-based approaches, achieving state-of-the-art performance [6]. Recently, deep reinforcement learning (DRL) and federated learning techniques have been integrated to manage the resource of the mobile edge computing system [7]. In another related work, an AI embedded cognitive network architecture is proposed to safeguard the stability and efficiency of the communications in a heterogeneous IoT system [8]. We note that there is a significant amount of effort on AI related techniques for the next-generation wireless communication system, which we are not able to elaborate due to limited space. Interested readers can refer to [9] for a comprehensive survey on this topic.

Inspired by the latest research and technology development in the community, in this article we propose a deep learning framework to address the critical challenges for energy-efficient distributed cooperative spectrum sensing application. Our architecture is built based on two learning techniques: graph neural network and reinforcement learning. The graph neural network is employed to embed the graph-structured data of the sensor network to a feature vector of a fixed length. The neural network weights of our architecture are randomly initialized, and then trained through an iterative learning process to improve the performance. Because the optimal solutions are not available for training, reinforcement learning is an essential component for our method. As a widely used reinforcement learning methodology, Q-learning has shown its success in training deep architectures including both the traditional convolutional neural network [10] and the graph convolutional neural network [6]. Therefore, we leverage Q-learning as a training tool to instruct our deep architecture to learn a sensor selection policy for improved energy efficiency.

Sensor-Aided Cognitive Raido Network with Distributed Cooperative Spectrum Sensing

The considered sensor-aided cognitive radio network is depicted in Fig. 1. This sort of cognitive radio network consists of PUs, SUs, and spectrum sensors. The sensors collaborate with each other through a consensus algorithm to detect the activities of the PUs. The detection results are reported to the spectrum policy server via nearby nodes. Then the spectrum policy server will build a real-

Inspired by the latest research and technology development in the community, in this article we propose a deep learning framework to address the critical challenges for energy-efficient distributed cooperative spectrum sensing application. Our architecture is built based on two learning techniques: graph neural network and reinforcement learning.

It is essential for the sensor network to perform spectrum sensing continuously because the SUs must vacate the spectrum band if the PU reclaims the band. As a result, the sensor network can consume tremendous energy over time. In this case, energy efficiency is an imperative consideration for the sensor network.

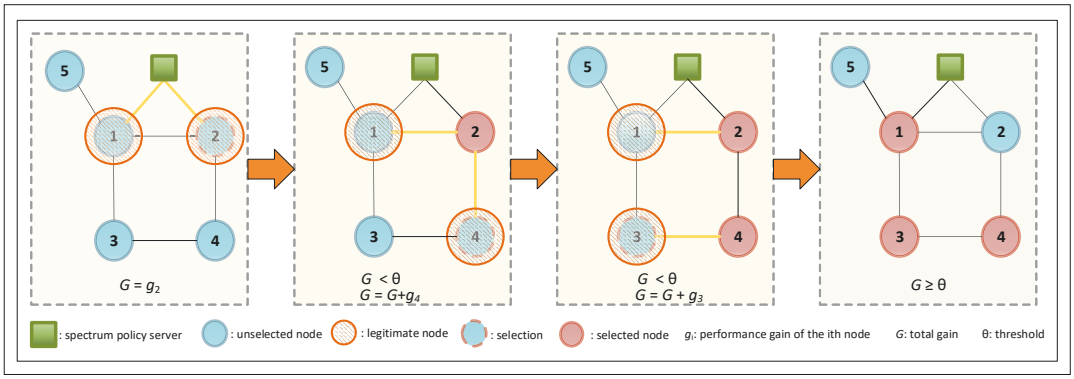

FIGURE 2. Sequential sensor selection process for distributed cooperative sensing. In this process, we pick one node a time until the summation of all the performance gains (G) of the selected nodes surpass a pre-set threshold (θ). In each selection step, the legitimate candidate nodes for selection are those connected to one of the previously selected nodes.

time spectrum map and allocate the available spectrum bands to the SUs [11]. In our work, we assume that the sensor network is built to a regular grid with variations. In this case, each sensor does not have too many neighbors. The variations are introduced to simulate the real-world environment since a sensor's installation position may be affected by other constraints in reality. The cooperative sensors typically use a common detection technique, such as matched filter, cyclostationary feature detection, or energy detection, to detect the spectrum utilization status. Among these detection methods, energy detection is one of the most popular approaches due to its simplicity and independence from the specific form of the PUs' signal. Therefore, in this article, we adopt energy detection as the basic detection technique.

With energy detection, the distributed cooperative sensing procedure can be illustrated by the right part of Fig. 1. The graph topology describes the communication relationship of the sensors, in which the nodes and the edges represent the sensors and the duplex communication link, respectively. Because the sensors are spatially distributed, the channel gains from the PU signal transmitter to different sensors are distinct as well as the local noise level. To manifest this property, we depict the nodes with different colors, which represent their own local SNR levels. In the sensing procedure, all the sensors first measure the signal energy on the target spectrum band. Then they recursively communicate with their neighbors to fuse these measurements with a consensus protocol until convergence. Finally, the converged result is compared to a predefined threshold to assert whether the spectrum band is occupied or not. In this process, we can see that the final converged value is decided by both the initial measurement and the adopted consensus protocol. In general, there are two major consensus protocols in the community: average consensus protocol [3] and weighted average consensus protocol [4]. In the case of average consensus protocol [3], the final converged value will be the average of all the initial energy measurements. In this way, all the nodes are deemed equally important in deciding the final converged result without considering their SNR differences. On the other hand, for weighted average consensus protocol [4], each node is weighted proportionally according to its local SNR, meaning the sensors under high SNR play a more important role for the final converged result. In this article, we adopt the weighted average consensus protocol.

It is essential for the sensor network to perform spectrum sensing continuously because the SUs must vacate the spectrum band if the PU reclaims the band. As a result, the sensor network can consume tremendous energy over time. In this case, energy efficiency is an imperative consideration for the sensor network. While the energy efficiency problem has been substantially investigated for fusion center based sensor networks, limited understanding and techniques are available for distributed cooperative sensor networks given their unique characteristics. To this end, in the following section, we develop an optimization paradigm for the energy efficiency problem of the distributed collaborative sensor network.

Energy-Efficient Distributed Cooperative Spectrum Sensing

A cognitive radio network usually has a target detection performance for a spectrum band. To achieve this desired performance, the cooperative sensing scheme must include sufficient distributed sensors. Traditionally, the cooperative nodes in one sensor network are usually identical, which means the energy consumption of different sensors are the same. Hence, from the energy efficiency perspective, it is wise to use the smallest number of sensors.

With the weighted average consensus protocol, although all sensors can contribute to the improvement of the detection performance, the cooperative gain can be very distinct from different nodes [4]. The nodes experiencing high SNR can substantially enhance the detection performance, while the contributions from the sensors under low SNR can even be neglected. It is easy to infer that for a specific target detection performance, fewer nodes will be needed if their cooperation gains are high. Hence, the desired scheme should pick the sensors with high SNR to perform the collaborative sensing for this spectrum band. Although this objective is intelligible, one cannot simply pick the sensors one by one based on their local SNR. This is because, besides the SNR, the topology of the nodes is also

an important consideration for a distributed cooperative sensing system. To be specific, the communication graph of the selected nodes must be connected to guarantee the topology requirement of the distributed cooperative sensing algorithm [3, 4].

Based on all these discussions, it can be inferred that for a spectrum band, an energy-efficient spectrum sensing scheme is a minimum subset of connected sensors that can achieve the target spectrum sensing performance. Researchers have pointed out that the spectrum sensing performance can be measured by the modified deflection coefficient (MDC) [12], which means for the desired detection performance, there is a corresponding MDC value. With the weight design scheme introduced in [12], the MDC is proportional to the weighted summation of the squared amplitudes of the channel gains. Therefore, if we define a new metric for each sensor by multiplying the original weight with the squared channel gain, the summation of the new metrics will be linearly related to the MDC. We call this metric *performance gain* (g) in this article. Thus, a legitimate set of the sensors that can achieve the desired detection performance can be found by sequentially selecting the sensor until the summation of the performance gains reaches a certain threshold (θ). This sensor selection procedure can be naturally illustrated as in Fig. 2. In this process, we begin with an empty solution and add one node a time until the threshold (θ) is reached. At the first step, we select a node that is connected to the spectrum policy server to ensure the sensing results can be delivered to it. In the following steps, the feasible selections are restricted to the nodes that have connections to the previously selected ones. The optimal solution is characterized by the minimum number of sensors, which can be obtained through an exhaustive search. However, due to the large computational complexity, an exhaustive search is not practical, especially for the large-scale sensor network. Recently, deep learning has been successfully applied to some of the combinatorial optimization problems [6]. Inspired by this prior research, in this article, we develop a deep learning based approach to energy-efficient distributed cooperative sensing.

Deep Reinforcement Learning Based Solution

When solving a sequential decision making problem via deep learning, a deep architecture will be trained to predict the quality or the Q value of all actions at each step and execute the action of the highest quality. Most of the classic deep architectures are designed to handle the data of regular shape and output the Q values of a fixed set of actions. For example, the deep Q network (DQN) for Atari games requires the inputs to be the video images and generates the action values of all buttons on the controller [10]. In contrast, the data of the sensor network is graph-structured, which is irregular, and the legitimate actions vary at different steps. Moreover, the topologies of different sensor networks can be vastly distinct. To overcome these difficulties, we design the following DRL framework, which is depicted in Fig. 3. It takes graph-structured data as input and is trained to make the decision that can benefit the long-term performance at each step. This architecture relies on two learning techniques: the graph neural network and reinforcement learning, which are used for feature extraction and weight tuning, respectively. We elaborate the details of our method centering on these two aspects.

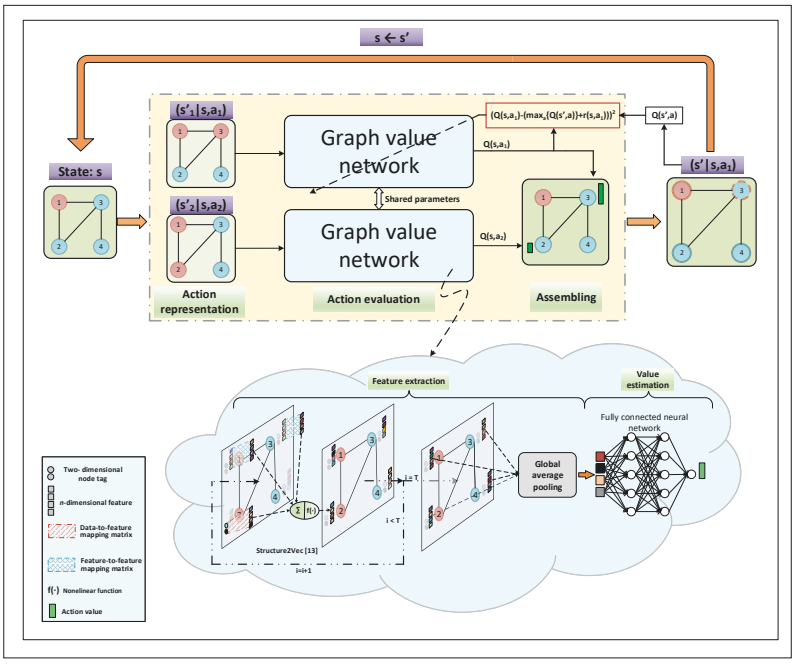

FIGURE 3. The proposed DRL framework. The upper part depicts the decision making and training procedure. The lower part illustrates the specific structure of the graph value network, where Structure2Vec is the graph neural network used to generate the graph feature, and the fully connected neural network is used to calculate the action value.

Sensor Selection via the Proposed Architecture

Our proposed deep learning framework is obligated to make decisions for the sensor selection process presented in the above section. At each step, the input is the communication graph with each node carrying a two-dimensional tag, where the first component is the performance gain of the node normalized by the target threshold, and the second component denotes whether the sensor has been selected or not with 1 or 0. Our method estimates the Q values of all legitimate actions based on this graph-structured data. As the legitimate actions vary at different steps, designing a consistent action representation is not forthright. Considering the fact that the state transition is deterministic in the sensor selection process, we first use the future graph states to represent the corresponding actions. Then these action representations are input to a value network separately to estimate the corresponding Q value. Finally, the selection is made with a greedy policy.

The specific procedure is depicted in the yellow box of Fig. 3. At each step, we first infer the associated future states of all the feasible actions by modifying the second component of the tags for the corresponding nodes. Then these inferred graph-structured data are deemed as the action representations and input to the value network independently. Since this procedure is equivalent to employing multiple graph value networks with shared weights, we draw one graph value network for each action to facilitate visualization. The

Sensor network scale	Sensor coordinate	Spectrum server coordinate	Number of nodes under different performance gain (g)			Threshold (θ)
			$g = 0.1$	$g = 0.3$	$g = 0.9$	
16-node	$\{(x + \sigma_x, y + \sigma y) \mid 0 \le x, y \le 3\}$	(1.5, 1.5)	11	4	1	2
24-node	$\{(x + \sigma_x, y + \sigma_y) \mid 0 \le x, y \le 4, (x, y) \ne (2,2)\}$	(2, 2)	17	6	1	2.5
36-node	$\{(x + \sigma_x, y + \sigma_y) \mid 0 \le x, y \le 5\}$	(2.5, 2.5)	25	9	2	4.0
48-node	$\{(x + \sigma_x, y + \sigma_y) \mid 0 \le x, y \le 6, (x, y) \ne (3,3)\}$	(3, 3)	33	12	3	5.7
64-node	$\{(x + \sigma_x, y + \sigma_y) \mid 0 \le x, y \le 7\}$	(3.5, 3.5)	44	15	4	7.6
80-node	$\{(x + \sigma_x, y + \sigma_y) \mid 0 \le x, y \le 8, (x, y) \ne (4,4)\}$	(4, 4)	56	18	6	9.7

TABLE 1. Parameters of the considered sensor networks.

graph value network estimates the action values through two steps: feature extraction and value estimation. The feature extraction process aims to embed graph-structured data to a distinctive feature utilizing a particular graph neural network, Structure2Vec [13]. Structure2Vec generates the n-dimensional feature of each node via an iterative feature propagation process. At each iteration, the node feature is updated according to the features of all the neighbors and its own tag. To make it clear, we make an illustration with node two. First, the two-dimensional tag of a node is mapped to the feature space though the data-to-feature mapping matrix, while its neighbors' features are mapped through the feature-to-feature mapping matrix. Then the obtained mappings are added together and input to a nonlinear function. The output will be applied to update the feature of node two. The n-dimensional initial feature vectors of all nodes are set to zero. At each iteration, all the node features are updated simultaneously. This process stops after T iterations, and the iteration number T is a hyperparameter that can be set to a small value based on our experience. The obtained features are then average pooled over the whole graph to produce the representation of this graph-structured data. After that, a fully connected neural network estimates the corresponding action value according to this representation. Finally, comparing all actions, the one of the highest quality is executed.

Training with Reinforcement Learning

There is no doubt that the action evaluation accuracy directly affects the quality of the generated solution. Therefore, a training process is essential for our deep structure to learn the correct estimation. It is clear that our problem fulfills the Markov property, which naturally leads us to employ reinforcement learning as the training method. Reinforcement learning is a powerful tool that enables an agent to learn an optimal policy for a Markov decision process through interactions with the environment [14]. In reinforcement learning, the agent receives a reward from the environment at each time step, and the learning algorithm can guide the agent to discover an optimal policy that maximizes the expected rewards accumulated over time. Consequently, when applying reinforcement learning to optimization or planning problems, it is critical to design a proper reward signal at each step that is compatible with the final goal. In our problem, since the optimal solution contains the fewest sensors, we set the reward for each action to –1. In this way, maximizing the accumulated rewards is precisely equivalent to minimizing the number of sensors in the solution.

In a traditional tabular reinforcement leaning approach, the action values at each step are stored in a table, and the learning refers to recursively updating the action values based on the related reward and the estimated future return. With neural network based reinforcement leaning, rather than adjusting the stored action value, the learning process aims to tune the weights of the neural network such that it can accurately estimate the action qualities at all encountered states. Q-learning is a widely used off-policy algorithm for both tabular and neural network based approaches. In this article, we train our deep architecture via Q-learning. Since the precise action value under an optimal policy should equal the maximum expected future return, a Q-learning algorithm adapts the estimated action value or the estimation functions with the maximum action value of the forthcoming state. We present the Q-learning process of our architecture in Fig. 3. At each step, the action values are predicted via our architecture. Then the ε-greedy policy is applied to decide the action. With ε-greedy policy, there is a probability that the action will be randomly selected. This randomness is necessary because the action value can be incorrectly estimated at the beginning, and it is beneficial to implement explorations. With the applied action, we can obtain a reward and a corresponding future state. The actions at this future state will also be evaluated by our architecture. Then the summation of the maximum Q value of these actions and the reward is used as the target of the predicted Q value of the executed action. Finally, to make the prediction close to the target, we utilize the gradient descent algorithm to tune the mapping matrices in Structure2Vec and the weights of the fully connected neural network.

The sensor-aided cognitive radio network may be built at different areas to deal with the local spectrum shortage problem. Hence, we expect that our deep architecture can be trained to learn a generalized policy, such that it is able to find the energy-efficient cooperative scheme for a class of sensor networks. Similar to other reinforcement learning problems [6, 10], numerous training cases are essential for achieving this goal. Obviously, it is not possible to construct sensor network entities for the

purpose of training. Instead, we assume that the different sensor networks are the instances of a common model, and we can generate the training cases with this model. The rationale behind this assumption is that cognitive radio networks are most likely to be built in the area of high user density, such as some big cities. In these areas, the environments are similar, such that it is possible to find a generalized model. In this article, since the modeling strategy is independent of our proposed method, we will use a simplified sensor network model to test the effectiveness of our proposed method.

Performance Evaluations

In this section, we conduct numerical simulations to verify the performance of our proposed approach. We consider sensor networks of different scales, where the sensors are deployed in a two-dimensional grid structure with certain variations. The coordinate of each sensor is set to the form of $(x + \sigma_x, y + \sigma_y)$, where the integers x and y are the coordinates of the vertex in a regular quadrilateral grid; σ_x and σ_y are the Gaussian noise with zero mean and variance of 0.1 to simulate the variations. In each sensor network, the spectrum policy server is located in the central area. The communication range of each sensor is set to 1.35. We assume that the sensors randomly experience three levels of SNR, and accordingly, their performance gains are set to three different values: 0.1, 0.3, and 0.9. In our experiments, the thresholds are set to larger values as the networks scale up. The reason for this setting is that building a large sensor network usually aims to obtain high sensing performance. Based on this model, we generate 1000 training cases and 1000 testing cases for each network scale. The parameters of the sensor networks are summarized in Table 1.

We train our proposed deep architecture with the training cases and apply it to select the nodes for the testing cases. Good performance is indicated by a low averaged number of the selected nodes of all test networks since we aim to use as few nodes as possible to improve the energy efficiency. In our proposed deep architecture, we adopt the Rectifier activation function everywhere, and the fully connected neural network for the action value calculation is designed as a three-layered neural network with 32 hidden neurons. In our architecture, the feature embedding iteration number (T) and the feature dimension (n) can be especially important since they may directly affect the quality of the action representation. To find the values of these two parameters that can lead us to relatively high-quality results, we test different combinations on the 16-node sensor networks. First, we build a group of the proposed deep architectures and set their feature dimensions and embedding iterations to the values listed in Table 2. Then, with randomly initialized weights, we train them with the same training settings and test their performance on the testing cases. Since the initial weights will affect the final performance, we repeat these experiments 20 times. The average number of the nodes in the solutions are reported in Table. 2. Based on these results, we set the embedding iteration number to three and the feature dimension to 32, which can generate the best performance.

Embedding iteration number (T)	Average number of selected nodes under different feature dimension (n)					
	$n = 8$	$n = 16$	$n = 24$	$n = 32$	$n = 40$	$n = 48$
$T = 1$	10.77	10.20	9.66	9.94	9.92	10.29
$T = 2$	10.10	9.77	9.67	9.41	9.26	9.53
$T = 3$	10.08	9.36	8.75	8.28	8.48	9.02
$T = 4$	10.37	10.38	9.25	8.76	8.79	9.14
$T = 5$	9.85	9.84	10.01	9.82	9.24	9.33

TABLE 2. The average number of selected nodes by the proposed deep architectures with different feature dimensions and embedding iteration number. The highlighted underlined numbers represent the best performance under each embedding iteration number.

To better evaluate our approach, we also solve the testing cases with a greedy heuristic (GH)-based approach and a genetic algorithm (GA)-based approach as comparisons. The idea of the applied greedy algorithm is straightforward, which always selects the sensor with maximum performance gain at each step. The GA-based approach is designed based on the priority-based encoding method [15]. In this approach, each sensor is assigned a priority value, and the priority values of all nodes compose the genetic representation of the solution. With this representation, we obtain the solution by selecting the legitimate node with the highest priority at each step. Accordingly, the fitness function is set to the accumulated rewards for this GA- based approach.

Figure 4a shows the performance of these three approaches. Comparing our proposed method to the GH-based approach, it is clear that our proposed method always performs better for each network scale. Furthermore, we find a performance improvement increasing from 15 percent to 28 percent as the network gets larger. On the other hand, the GA-based approach achieves comparable performance. For the 16-node networks, it even performs slightly better than our method. However, when we use these two methods to solve the test cases, there is a great difference in terms of the computation time, which is revealed in Fig. 4b. It should be noted that the computation time of the GH-based approach is not included because it is almost invisible compared to the other two. From Fig. 4b, we can observe that the time consumption of our method is obviously less than that of the GA-based approach. In addition, this advantage enlarges as the network scales up.

The experiments show some interesting results. For the small-scale networks, where the sensor combinatorial problem is relatively simple, the advantage of our DRL-based approach is not very distinct. It is likely that with the GH-based approach, we can find an adequate solution. Moreover, the GA-based approach can provide us an equally good or even better result with acceptable computation time. However, as the sensor network scales up, the solution space experiences an exponential increase. The probability of finding a good solution with the GH-based approach can be low in this case, while the GA-based approach takes a long time to search in the huge solution

Our deep learning architecture leverages a graph neural network and reinforcement learning to estimate the action quality in the sensor selection process. Simulation results under different network scales demonstrate the effectiveness of our proposed approach.

space. On the other hand, our trained deep architecture achieves a good balance between solution quality and computation time. These results clearly demonstrate the superiority of deep learning in solving the highly complex problem.

Conclusion and Future Work

In this article, we develop a deep learning based approach to improve the energy efficiency of distributed cooperative spectrum sensing for sensor-aided cognitive radio network. A sequential sensor selection method has been designed to find a legitimate subset of sensors that can not only fulfill the target sensing performance, but also guarantee the topology requirement of the distributed sensing algorithm. Our deep learning architecture leverages a graph neural network and reinforcement learning to estimate the action quality in the sensor selection process. Simulation results under different network scales demonstrate the effectiveness of our proposed approach.

As a promising research topic, there are many interesting future research directions that can be considered. For instance, in our current study, we consider the sensor network topology to be a two-dimensional grid structure with certain variations. Therefore, it would be interesting to investigate how the proposed method can be generalized to other network topology structures. Also, the proposed energy-efficient spectrum sensing scheme is obtained through a globalized planning style, which requires the related parameters of each sensor, such as channel gain, noise level, and signal strength, to be known. However, these parameters may be uncertain over time with the change of the surrounding environment. Therefore, distributed real-time learning will be critical for this approach to be applicable to practical communication systems in reality.

Acknowledgment

This research was supported by the National Science Foundation (NSF) under grant ECCS 1731672.

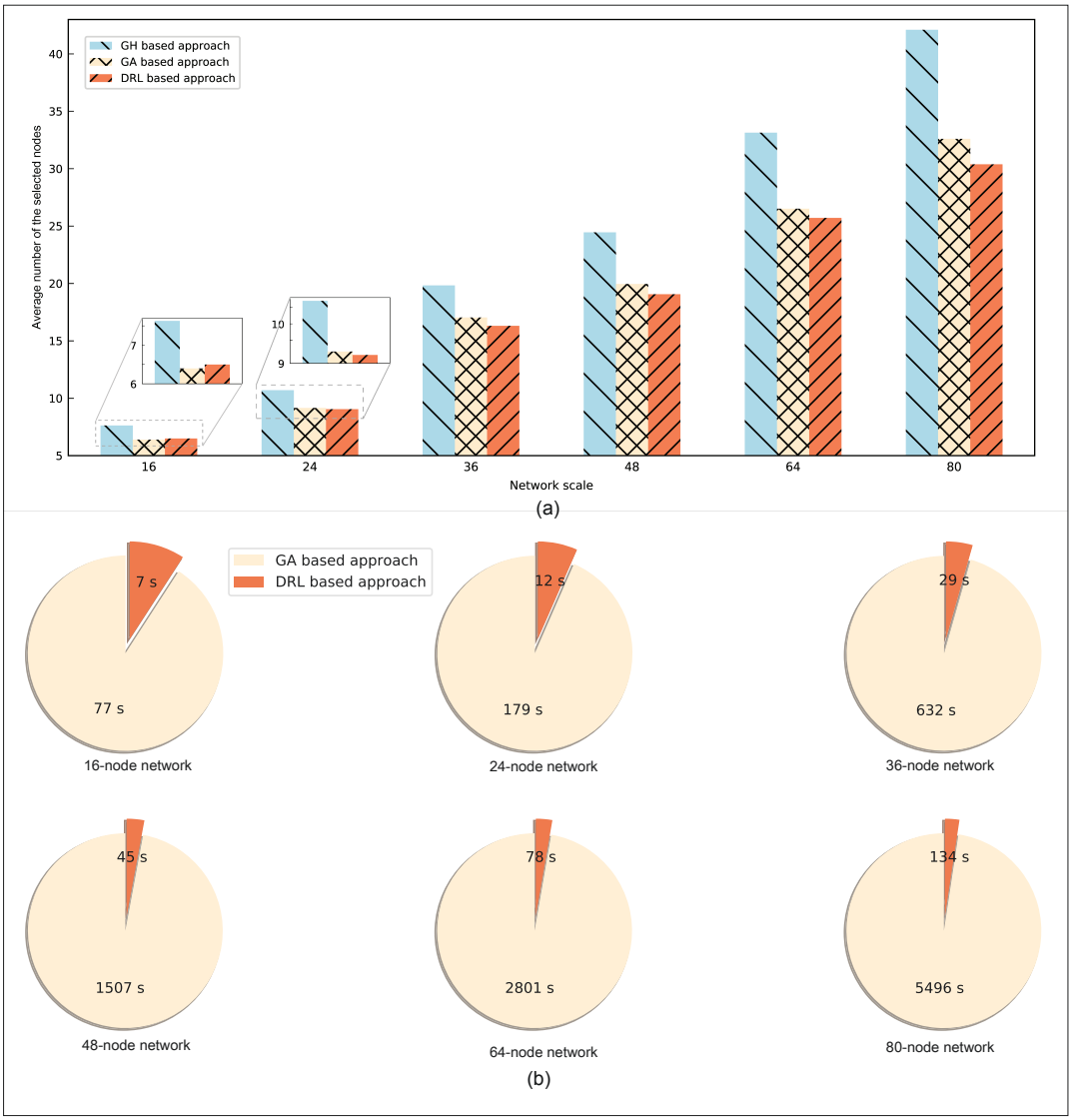

FIGURE 4. Performance comparison under different network scales. In a), the x-axis denotes the number of nodes in the test network, and the y-axis shows the average number of nodes selected by the three comparative approaches: GH, GA, and the proposed DRL. The performance for each network scale is based on the average of 1000 testing cases. In b), the pie charts show the computation time (in seconds) of the GA-based approach and the DRL-based approach.

References

[1] J. Mitola, *Cognitive Radio — An Integrated Agent Architecture for Software Defined Radio*,Ph.D. dissertation, KTH Royal Inst. Tech., 2000.

[2] I. F. Akyildiz, F. L. Brandon, and B. Ravikumar, "Cooperative Spectrum Sensing in Cognitive Radio Networks: A Survey," *Physical Commun.*, vol. 4, no. 1, 2011, pp. 40–62.

[3] Z. Li, F. R. Yu, and M. Huang, "A Distributed Consensus-Based Cooperative Spectrum-Sensing Scheme in Cognitive Radios," *IEEE Trans. Vehic. Tech.*, vol. 59, no.1, 2010, pp. 383–93.

[4] W. Zhang *et al.*, "Distributed Consensus-Based Weight Design for Cooperative Spectrum Sensing," *IEEE Trans. Parallel and Distributed Systems.*, vol. 26, no. 1, 2015, pp. 54–64.

[5] P. Cheng, R. Deng, and J. Chen, "Energy-Efficient Cooperative Spectrum Sensing in Sensor-Aided Cognitive Radio Networks," *IEEE Wireless Commun.*, vol. 19, no. 6, Dec. 2012, pp. 100–05.

[6] E. Khalil *et al.*, "Learning Combinatorial Optimization Algorithms over Graphs," *Advances in Neural Information Processing Systems*, 2017.

[7] X. Wang *et al.*, "In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning," arXiv preprint arXiv:1809.07857, 2018.

[8] M. Chen *et al.*, "Cognitive-LPWAN: Towards Intelligent Wireless Services in Hybrid Low Power Wide Area Networks," *IEEE Trans. Green Commun. and Networking*, vol. 3, no. 2, June 2019, pp. 407–17.

[9] C. Jiang *et al.*, "Machine Learning Paradigms for Next-Generation Wireless Networks," *IEEE Wireless Commun.*, vol. 24, no. 2, Apr. 2017, pp. 98–105.

[10] V. Mnih *et al.*, "Human-Level Control through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, 2015, pp. 529–33.

[11] I. F. Akyildiz *et al.*, "A Survey on Spectrum Management in Cognitive Radio Networks," *IEEE Commun. Mag.*, vol. 46, no. 4, Apr. 2008, pp. 40–48.

[12] Z. Quan, S. Cui, and A. H. Sayed, "Optimal Linear Cooperation for Spectrum Sensing in Cognitive Radio Networks," *IEEE J. Selected Topics in Signal Processing*, vol. 2, no. 1, 2008, pp. 28–40.

[13] H. Dai, B. Dai, and L. Song, "Discriminative Embeddings of Latent Variable Models for Structured Data," *Proc. Int'l. Conf. Machine Learning*, 2016.

[14] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

[15] M. Gen, R. Cheng, and D. Wang, "Genetic Algorithms for Solving Shortest Path Problems," *IEEE Int'l Conf. Evolutionary Computation*, 1997.

Biographies

Haibo He [SM'11, F'18] (haibohe@uri.edu) is the Robert Haas Endowed Chair Professor in Electrical Engineering at the University of Rhode Island. He has published more than 300 journal and conference papers. He received the IEEE International Conference on Communications Best Paper Award (2014), IEEE Computational Intelligence Society Outstanding Early Career Award (2014), and National Science Foundation CAREER Award (2011). He is currently the Editor-in-Chief of *IEEE Transactions on Neural Networks and Learning Systems*.

He Jiang [S'17] (hjiang@ele.uri.edu) received his B.S. degree from North China Electric Power University, Beijing, China, in 2012, and his M.S. degree from the School of Electrical and Electronics Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2015. He is currently working toward a Ph.D. degree in the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston. His research interests include adaptive dynamic programming, and reinforcement learning and its applications in communication systems.