applied sciences

MDPI

Article

Learning Frameworks for Cooperative Spectrum Sensing and Energy-Efficient Data Protection in Cognitive Radio Networks

Vinh Quang Do and Insoo Koo *

School of Electrical Engineering, University of Ulsan, Ulsan 44610, Korea; vquang.do@gmail.com
* Correspondence: iskoo@ulsan.ac.kr; Tel.: +82-52-259-1249

Received: 27 March 2018; Accepted: 25 April 2018; Published: 4 May 2018

Abstract: This paper studies learning frameworks for energy-efficient data communications in an energy-harvesting cognitive radio network in which secondary users (SUs) harvest energy from solar power while opportunistically accessing a licensed channel for data transmission. The SUs perform spectrum sensing individually, and send local decisions about the presence of the primary user (PU) on the channel to a fusion center (FC). We first design a new cooperative spectrum-sensing technique based on a convolutional neural network in which the FC uses historical sensing data to train the network for classification problem. The system is assumed to operate in a time-slotted manner. At the beginning of each time slot, the FC uses the current local decisions as input for the trained network to decide whether the PU is active or not in that time slot. In addition, legitimate transmissions can be vulnerable to a hidden eavesdropper, which always passively listens to the communication. Therefore, we further propose a transfer learning actor–critic algorithm for an SU to decide its operation mode to increase the security level under the constraint of limited energy. In this approach, the SU directly interacts with the environment to learn its dynamics (i.e., an arrival of harvested energy); then, the SU can either stay idle to save energy or transmit to the FC secured data that are encrypted using a suitable private-key encryption method to maximize the long-term effective security level of the network. We finally present numerical simulation results under various configurations to evaluate our proposed schemes.

Keywords: actor–critic; cognitive radio network; data encryption; energy efficiency; spectrum sensing

1. Introduction

Cognitive radio is one of the effective solutions to the problem of spectrum scarcity in wireless communications networks. Secondary users (SUs) with cognitive capability can utilize the spectrum bands licensed to primary users (PUs) for reliable and effective data transmission [1]. To achieve this, the SU modifies its parameters to adapt to the time-slotted operation of the PU on the channel of interest, and then senses the presence of the PU on that channel in every time slot. When the PU is sensed as inactive in a particular time slot, the SU can use the licensed channel during that time slot to transmit data. In this paper, the SU uses its limited-capacity battery, powered by a non-radio frequency (non-RF) energy harvester, for spectrum sensing, data encryption, and data transmission.

1.1. Motivation

Many studies concerning energy management problems for energy-harvesting nodes have been conducted, primarily to maximize a system's throughput [2–6]. For example, Park and Hong [2] proposed a joint design of a spectrum sensing policy and a detection threshold to maximize total expected throughput under energy constraints. Pappas et al. [4] examined the two-dimensional

maximum stable throughput region for a simple cognitive system comprising two source-destination pairs. Razaque and Elleithy [6] designed an intelligent decision-making (IDM) model for wireless sensor networks, which allows the sensor node to obtain energy from the Sun, and thus preserves its battery energy in an outdoor environment. Liang et al. [7] studied the optimal sensing duration to maximize achievable throughput for a secondary network while sufficiently protecting primary users. There is research that analyzes optimal transmission power and density of secondary transmitters to maximize secondary network throughput under the constraints of a given outage-probability [8]. In addition, the work in [9] explored a multiple-input multiple-output (MIMO) technique for collaborative spectrum sensing for the distributed detection framework in cognitive-radio scenarios; this paper focuses on the reporting channel in a spectrum-sensing context and exploits the results from decision fusion to improve probability of detection.

In addition, cognitive radio networks (CRNs), like any modern communications system, should guarantee the privacy of the data traveling through the network [10]. However, due to its open and random access nature, wireless communications in CRNs is susceptive to security threats targeting the physical or media access control layers (e.g., passive eavesdropping or radio frequency (RF) jamming). For that reason, a remarkable number of contributions focus mainly on security technologies for CRNs [11]. In particular, Wen et al. [12] presented physical layer approaches to defend against security threats in CRNs. The authors first introduced a MIMO technique that guarantees a low probability of interception, and that enhances the confidentiality of the network; then, they proposed an identified scheme based on channel responses to defend against primary user-emulation attacks. Ciuonzo et al. [13] studied channel-aware decision fusion rules to classify the presence of a (either distributed or co-located) multi-antenna jamming device in wireless sensor networks.

Moreover, physical layer security in CRNs has been widely studied to secure wireless transmissions, especially in the presence of a hidden eavesdropper [14,15]. Besides this, keeping the data classified from prying eyes by using encryption techniques is one of the most feasible solutions to maintain security; but, in reality, it is not easy to implement conventional encryption techniques in CRNs, since the networks have constrained resources (e.g., limited energy or memory). As a consequence, encryption techniques such as symmetric and asymmetric key algorithms are not preferred for data protection in CRNs. Nevertheless, in modern CRNs, wireless energy harvesting technology can ensure the energy autonomy of the network by using a small rechargeable battery integrated with an energy harvester, thus providing the SUs with redundant energy to improve data security. Therefore, protecting data using encryption methods still attracts a lot of interest in the research community [16–18]. To illustrate, Sen [19] identified numerous security threats to cognitive wireless sensor networks and the defense mechanisms against these vulnerabilities by selecting the most appropriate cryptography algorithm for each class of attack.

Recent work proposes an energy-efficient data encryption scheme for an SU powered by an energy harvester to decide its operation mode (e.g., stay silent or transmit encrypted data) in the current time slot [20]. This scheme aims to find an optimal policy for the data encryption decision to maximize the long-term security level of the system. More specifically, the scheme uses a well-known symmetric encryption method called the Advanced Encryption Standard (AES) [21] for the same data block length with different key sizes (AES-128, AES-192, AES-256). The SU can encrypt data using an algorithm with longer key lengths to enhance security, and then transmits the encrypted data on an idle licensed channel. Furthermore, the SU determines the encryption key length based on the impact of spectrum sensing error, the energy causality constraint, and the effect of the current decision on future time slots. The problem is first formulated as the framework of a partially observable Markov decision process (POMDP), and is then solved by using value iteration-based dynamic programming to find the optimal policy. However, this solution is rarely directly useful in reality. It is akin to an exhaustive search, looking ahead at all possibilities, computing the probabilities of occurrence and their desirability in terms of expected rewards (i.e., security levels) [22]. The solution relies on the assumption that we know in advance the dynamics of the environment (i.e., an arrival of harvested energy), which is rarely

true in practice. Consequently, this paper is going to investigate the problem from a different point of view in which the solution does not require prior information about the environment's dynamics.

1.2. Contributions

Our focus in this paper is to solve the problem of reaching a data encryption decision that aims to maximize the security of data transmissions in CRNs by using model-free reinforcement learning [22], namely, an actor–critic algorithm. The main advantage of the actor–critic solution over the POMDP-based approach is that it does not require complex computations or information about the arrival of harvested energy. In this work, we model the arrival of harvested energy and the primary traffic as a Poisson point process and a time-homogeneous discrete Markov process, respectively. At the beginning of a time slot, the SU does not have the exact information about the energy harvesting model and the spectrum occupancy status of the PU, except for the average value of harvested energy and the transition probabilities for the PU state. Thus, the SU needs to carry out spectrum sensing to identify whether the primary channel is busy or not; then, it either stays idle or transmits data on the free channel. Accordingly, to increase the chances for the SU to transmit data on the primary channel and to reduce the probability of collision with the primary user, we propose a new cooperative spectrum sensing technique using a convolutional neural network (CNN) and historical sensing data.

More than that, the primary purpose of this paper is to find an optimal data encryption decision policy that fits into the framework of a Markov decision process (MDP). During this process, we employ an actor–critic sequential learning model so the SU can interact with the environment in a stochastic way to acquire information on the environment's dynamics. Based on this method, the SU can learn the energy harvesting model and the primary traffic variations from the learning practice. Afterwards, it can either stay idle or select an appropriate key length for data encryption (also known as *action* in this paper), and then verify the effect of the decision based on the returned rewards. By repeating this kind of action over time, the SU can establish the policy to make determinations in the future. However, it would take time for the actor–critic learning procedure to converge to an optimal policy, especially with the large size of the state space [23]. To deal with such an issue, we employ the idea of transfer learning, which exploits the historical relevance of the harvested energy model and the primary user's activity in order to speed up the learning process of the conventional actor–critic algorithm [24]. In this paper, we call this method a transfer learning actor–critic (TLAC) algorithm. Compared with previous work, the main contributions of this paper are summarized as follows:

- We first introduce a new energy harvesting model, which is represented by a transformed Poisson distribution proven to give the nearest fit to the empirical measurements of a solar energy harvesting node for time-slotted operation [25].
- We also introduce a new CNN-based technique for cooperative spectrum sensing to enhance the performance of spectrum sensing by increasing the probability of detection while guaranteeing a low probability of false alarm.
- We then formulate the stochastic problem of the data encryption decision policy as the framework of a constrained MDP, and solve the problem by using the transfer learning actor–critic algorithm.

The rest of this paper is organized as follows. In Section 2, we introduce the system model of the proposed schemes. A new energy harvesting model based on transformed Poisson distribution is introduced in Section 3. Section 4 presents the new CNN-based cooperative spectrum sensing (CBCSS) technique. Section 5 focuses on the transfer learning actor–critic algorithm for data protection in CRNs. In Section 6, we evaluate the performance of the proposed schemes through numerical simulation results. Finally, we present a conclusion in Section 7. To make it clear, the most commonly used notations in this paper are listed in Table 1.

Table 1. The list of the most used notations.

Symbol	Description
$s(t)$	Primary user (PU) signal at time step t
$x_i(t), w_i(t)$	The received signal and additive white Gaussian noise at the ith secondary user (SU) at time step t
μ	Mean
σ^2	Variance
γ_i	Average gain of the sensed channel at the ith SU
E_{ca}	Battery capacity of the SU
E_s	The energy consumption for spectrum sensing process
e_r, e_h, e_{tr}	The remaining energy, harvested energy, transmit energy of the SU respectively
Nk	The key length of the encryption method
S_{Nk}	The security level corresponding with the key length Nk
P_d, P_f	The system probabilities of detection and false alarm
$\langle \mathbb{S}, \mathbb{A}, \mathbb{P}, \mathbb{R} \rangle$	Markov decision process (MDP) Tuple: State space $\mathbb{S}$, Action space $\mathbb{A}$, Transition probability function $\mathbb{P}$, and Reward function $\mathbb{R}$
ρ	The belief that the PU is inactive in a time slot
η	Discount factor
$V(s)$	State-value function
$\pi(s)$	Policy function
α_a, α_c	The actor and the critic step-size parameters
δ	Temporal difference (TD) error

2. System Model

The system considered in this paper comprises a pair of licensed primary users, several secondary transmitters (denoted as SUs), a secondary receiver equipped with a fusion center, and an eavesdropper (E), as shown in Figure 1. From now on, we will call the secondary receiver as the fusion center or FC for simplicity.

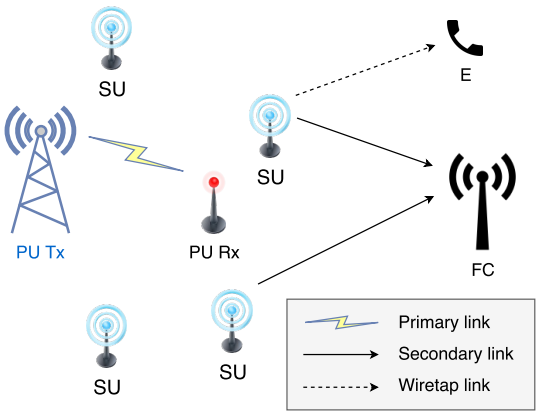

Figure 1. The model of the system considered in this paper. E: eavesdropper; FC: fusion center.

In this work, the SUs are assumed to always have data to transmit to the fusion center. Thus, they would try to access the licensed channel of the PUs for data transmission by carrying out cooperative spectrum sensing.

The primary user's states (active $[A]$ and not active $[\bar{A}]$) are assumed to follow a two-state Markov discrete-time process, in which the transition probabilities between the states are denoted $P_{i,j} : i, j \in \{A, \bar{A}\}$, as illustrated in Figure 2.

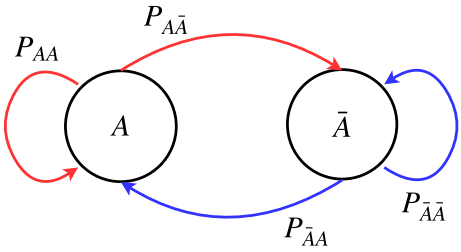

Figure 2. Two-state Markov discrete-time model for the primary user's states. $P_{i,j} : i, j \in \{A, \bar{A}\}$: the transition probabilities between the states.

The performance of the sensing scheme can be evaluated by using the probability of correct detection P_d and the probability of false alarm P_f. The former represents the probability of detecting the *active* state (A) of the PU accurately, whereas the latter indicates the probability that the PU is identified as active, but it is truly not $(\bar{A})$, each of which are given by

$$P_d = \Pr(H = A|A) \tag{1}$$

and

$$P_f = \Pr\left(H = A|\bar{A}\right), \tag{2}$$

respectively, where H denotes the state of the primary user as determined by spectrum sensing. Although the PU state transition probabilities are unknown in practical situations, the historical statistics information of the primary channel can be used to estimate the state transition probabilities based on the Markov model [26]. Therefore, we assume that the SU has a prior information about the PU state transition probabilities based on the historical sensing results; and the global information of the network (e.g., channel state information, probabilities of detection and false alarm) are available for all nodes in the network.

The system's operation proceeds as follows. The system is assumed to operate in a time-slotted manner. At the beginning of each time slot, the SUs perform spectrum sensing separately and send the sensing outcomes to the fusion center, where the data are fused together using a certain rule to decide the state of the primary user. The final sensing result is then broadcast to the SUs. If the channel is free, it is allocated to one of the SUs for data transmission. The SUs take turns using the channel, based on the arrival order of their transmission requests. Each SU can occupy the channel over many time slots until it finishes transmitting data. Meanwhile, the eavesdropper is listening to the communication quietly. Therefore, we are going to investigate learning frameworks for cooperative spectrum sensing and energy-efficient data protection against the hidden eavesdropper for the communication between one SU and the fusion center.

We first present a simple but effective cooperative spectrum sensing method based on a CNN to improve the sensing performance. The CNN is constructed and trained to predict the PU states by using individual sensing data as inputs, which leads to specific target outputs. Hence, the fusion center can make global decisions about the PU state based on the outputs of the neural network. Relying on the final decision, if the channel is free, it is allocated to an SU (denoted as SU1) to transmit data. Furthermore, the SU is assumed to have a finite-capacity battery regularly recharged by a non-RF energy harvester. In addition to that, under energy constraint, the SU encrypts data using the AES algorithm with an appropriate key length to maximize the long-term security level of the system.

Regarding data protection techniques, there are two primary types of cryptography: symmetric (or *private key*) and asymmetric (or *public key*) algorithms . In general, using private-key cryptography for data encryption is not a time-consuming process, and thus expends less energy than public-key cryptography. For example, the experimental results from Kim et al. [27] showed that a public-key algorithm named the elliptic curve integrated encryption scheme (ECIES) consumes a thousand times more energy during the encryption process than the popular AES-128 private-key method. Even though a public-key algorithm can increase the security level by sacrificing a huge amount of energy, it is not a favorable choice for many wireless systems like CRNs. Subsequently, in the paper, we focus on using the AES algorithm to secure the communications between SU1 and the FC. Specifically, the SU can use one of the three key sizes (128, 192, and 256) to encrypt data using the AES algorithm.

In this paper, the security level is defined by the number of repetitions of the transformation rounds that convert the input data into encrypted data [21]. Therefore, the security level S_{Nk} is dependent on the key length Nk of the AES algorithm, as follows:

- $S_{Nk} = 10$ if $Nk = 128$ bits,
- $S_{Nk} = 12$ if $Nk = 192$ bits,
- $S_{Nk} = 14$ if $Nk = 256$ bits.

Using the longer key lengths provides the SU with better data security but consumes more energy [28]. As a result, at the beginning of each time slot, the SU needs to decide its operation mode based on the sensing result and the remaining energy to maximize the long-term security level while efficiently using the limited energy. For example, it can stay silent to save energy for future use; or it can encrypt information by the AES algorithm with a proper key length and transmit the data to the FC. Therefore, in the paper, we additionally design an actor–critic learning framework for SU1 to find the optimal operation mode decision policy. More specifically, when the primary user is determined to be inactive and the remaining energy is sufficient for data transmission, the SU can decide to stay idle to save energy or to transmit data encrypted by the AES algorithm with a suitable key length by calculating the total expected reward in future time slots according to the proposed actor–critic learning algorithm.

3. Energy Harvesting Model

Recent advances in energy harvesting technologies allow small, low-cost devices such as wireless sensor nodes to operate based solely on wireless harvested energy that is stored in a finite-capacity battery. Hence, in designing network protocols, it is essential to obtain a reliable energy-harvesting model to guarantee energy autonomy in the network. In many studies, the arrival of harvested energy is assumed to be identical and independently distributed [29], to follow a deterministic Markov model [30], or to follow a normal Poisson point process [20], all of which are discrete-time models. In [31], the authors considered the problem of decentralized hypothesis testing in energy harvesting wireless sensor networks, where the arrival energy during a time interval is assumed to be drawn from a Bernoulli distribution. The extensive experimental results from Lee et al. [25] showed that the transformed Poisson distribution model produces the nearest fit for most of the empirical datasets.

In this paper, the number of energy packets that an SU can harvest during a particular time slot, e_h, is given as

$$e_h \in \{e_{h,1}, e_{h,2}, \ldots, e_{h,max}\}, \tag{3}$$

where $0 < e_{h,1} < e_{h,2} < \cdots < e_{h,max} < E_{ca}$, and E_{ca} is the maximum battery capacity of the SU. We assume that e_h follows a Poisson point distribution with mean $e_{h,avg}$. Furthermore, the fit with the

Poisson distribution can be improved by using a transformation $x = e_h - e_{h,min}$, where $e_{h,min}$ is the minimum harvested energy. The probability mass function (PMF) of e_h is then given by

$$\Pr(e_h) = \Pr(x = e_h - e_{h,min}) = \frac{e^{-x_{avg}} x_{avg}^{(e_h - e_{h,min})}}{(e_h - e_{h,min})!}, \quad (4)$$

where $x_{avg} = e_{h,avg} - e_{h,min}$ is the sample average of the new variable x. This new distribution is called the transformed Poisson distribution (TPD). This transformation of the original variable can improve the fitting to the empirical datasets, as proven in [25]. In practice, although it is not easy to measure the exact amount of harvested energy in a time-slot interval, we can always estimate the average, the minimum and the maximum values of the harvested energy. Meanwhile, if the normal Poisson point process is used, the minimum harvested energy is assumed to be 0 (or zero) by default , which is rarely true in practical scenarios.

Figure 3 shows the difference in the PMF between the normal Poisson distribution and the transformed Poisson distribution when the average harvested energy is $e_{h,avg} = 8$ packets, with different values of minimum harvested energy: $e_{h,min} \in \{1, 2, 3, 4\}$ packets. As can be seen from the figure, the SU can harvest with a higher probability those energy values located near the mean by using the transformed Poisson model. As a consequence, we can also improve the learning rate of the actor–critic algorithm because the SU can focus on learning the variations of the energy values that are adjacent to the mean.

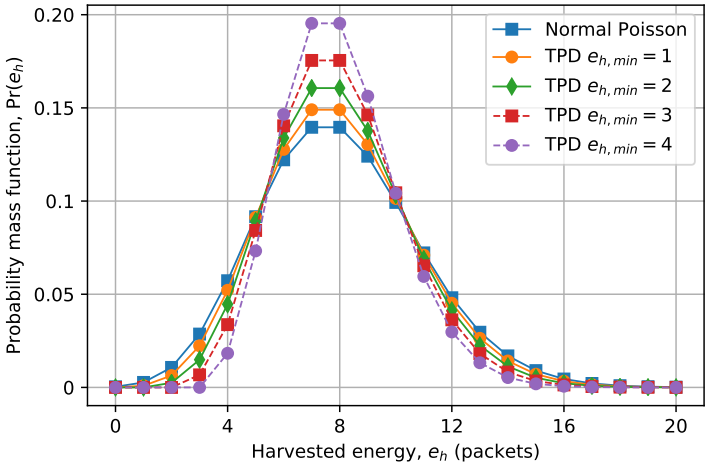

Figure 3. Comparison between the normal Poisson distribution and the transformed Poisson distribution (TPD) with $e_{h,avg} = 8$ and different values of $e_{h,min}$.

4. Convolutional Neural Network-Based Cooperative Spectrum Sensing

In this paper, we exploit the strength of the convolutional neural network, a particular type of deep neural network, to design a new cooperative spectrum sensing solution for the FC to determine the state of the PU on the primary channel. The process of cooperative spectrum sensing is illustrated with the following steps:

1. The FC trains the CNN using historical sensing data represented by the local spectrum decisions provided by the SUs.
2. At the beginning of each time slot, all the SUs are required to perform local spectrum sensing by using an energy detection method and reporting their sensing outcomes to the FC via a control channel.
3. The FC uses the new sensing data as input for the trained CNN to make a global decision about the PU state on the channel of interest, and then feeds back the final decision to the SUs.

Accordingly, the problem of neural network-based cooperative spectrum sensing is divided into two important parts: local spectrum sensing by the SUs and global decision making by the FC using the trained CNN.

4.1. Local Spectrum Sensing

The considered CRN is assumed to be composed of K SUs. Each of them performs spectrum sensing independently using an energy detection algorithm, and then sends the outcome to the FC. Moreover, we assume that the status of the PU remains unchanged during the time slot. The hypothesis test statistics for local spectrum sensing at SU i can be formulated as follows [32]:

$$\begin{cases} A: & x_i(t) = h_i s(t) + w_i(t), \\ \bar{A}: & x_i(t) = w_i(t), \end{cases} ; \forall i \in \{1, 2, \ldots, K\}, \tag{5}$$

where $x_i(t)$ is the received signal by the ith SU in time slot t, h_i denotes the channel gain of the link between the PU and the ith SU, $s(t)$ denotes the PU signal, and $w_i(t)$ is zero mean and unit variance additive white Gaussian noise (AWGN). Regarding energy detection, the observed energy at the ith SU is expressed as follows [33]:

$$xE_i = \sum_{j=1}^{N_i} |x_i(j)|^2; \ \forall i \in \{1, 2, \ldots, K\}, \tag{6}$$

where $x_i(j)$ is the jth sample of the received PU signal at the ith SU, and N_i is the number of sensing samples during each sensing period. For simplicity, we assume that the number of sensing samples collected by each SU is the same for all the SUs. When N_i is sufficiently large (e.g., $N_i \geq 200$), xE_i can be approximated by a Gaussian random variable under the two hypotheses (A and $\bar{A}$) with mean μ_A, $\mu_{\bar{A}}$ and variance σ_A^2, $\sigma_{\bar{A}}^2$, given as follows [34]:

$$xE_i \sim \begin{cases} \mathcal{N}\left(\mu_A = N_i(1+\gamma_i), \sigma_A^2 = 2N_i(1+2\gamma_i)\right), & A, \\ \mathcal{N}\left(\mu_{\bar{A}} = N_i, \sigma_{\bar{A}}^2 = 2N_i\right), & \bar{A}, \end{cases} \tag{7}$$

where γ_i is the average gain of the sensed channel in terms of signal-to-noise ratio (SNR). In this paper, we assume that γ_i follows a Gaussian distribution with mean μ_i and variance σ_i^2 as $\gamma_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$.

For a single-SU spectrum-sensing scheme, the local decision, D_i, is given by

$$D_i = \begin{cases} 1, & if \ xE_i \geq \lambda_i, \\ 0, & otherwise, \end{cases} \tag{8}$$

where 1 and 0 are single-bit data that represent states A and $\bar{A}$ of the primary user, respectively; and λ_i is a predefined decision threshold.

4.2. Convolutional Neural Network-Based Cooperative Spectrum Sensing

In a deep-learning research, the CNN is widely used in computer vision fields, such as image classification, speech recognition, and handwriting recognition, by making use of spatial characteristics. In this section, we present the process of creating and training a CNN for PU state prediction.

4.2.1. Network Configuration

The first step in designing a CNN is to define the network layers that specify the structure of the CNN, as depicted in Figure 4. This network consists of the following layers [35].

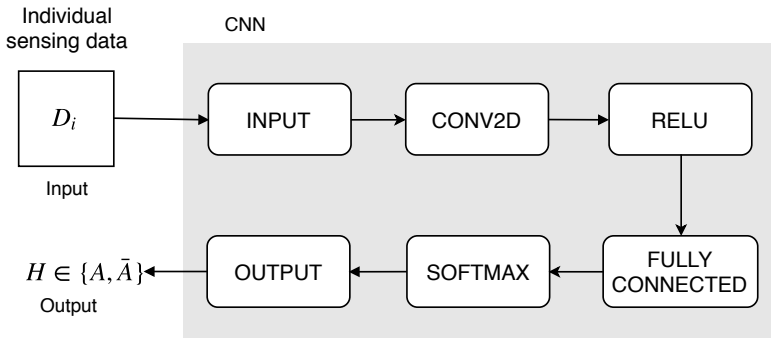

Figure 4. The structure of the convolutional neural network (CNN) for spectrum sensing in this paper.

- The *input* layer stores the input sensing data in the form of a gray scale image with size $1 \times K \times 1$, where K is the number of secondary users.
- The *convolutional (CONV2D)* layer contains K *neurons* (filters) that connect to the local subregions of the input image to learn its features by scanning through it. In this work, each region has a size of 1×2.
- The *rectified linear unit (ReLU)* layer uses the ReLU function to introduce nonlinearity to the CNN by performing a threshold operation on each input element, simply defined as

$$f(x) = \begin{cases} x, & x \geq 0, \\ 0, & x < 0. \end{cases} \tag{9}$$

- The *fully connected* layer combines all the local information from the original image (e.g., the results of feature extraction) determined in the previous layers to classify the status of the PU, which is active (A) or inactive $(\bar{A})$. Consequently, the size of the output data is equal to the number of states of the primary user.
- The *softmax* and *output* layers follow right after the fully connected layer for the classification problem. The softmax layer uses an output unit activation function, also known as a *normalized exponential* function, to create a categorical probability distribution for the two input elements (A and $\bar{A}$), as follows:

$$P(H_i) = \frac{\exp(q(H_i))}{\sum\limits_{H_j \in \{A, \bar{A}\}} \exp(q(H_j))}, \; i = 1, 2, \tag{10}$$

where $P(H_i)$ is the class prior probability; $H_i \in \{A, \bar{A}\}$ is an element class; and $q(H_i)$ is the output value from previous layer of the sample given class H_i. Thereafter, the output (or *classification*) layer takes the values from the softmax function and assigns each input to one of the two classes.

It should be noted that the original image with size $1 \times K \times 1$ is a vector containing the local decisions from K SUs; thus, a one-dimensional (1D) convolution layer can be used in the CNN to solve the problem of PU state classification instead of using a two-dimensional (2D) convolution layer. However, using a 2D CNN is more useful than 1D CNN in image classification. Furthermore, it would be easier to further develop the current approach to deal with three-dimensional data without making many changes in the current architecture of the CNN. For this reason, the size of the input image is generalized as $1 \times K \times 1$; thus, if the number of secondary users cooperating in spectrum sensing is large enough, the image size could be changed to $M \times N \times 1$, where $M \times N = 1 \times K$. Moreover, we can enhance the sensing accuracy by placing other information (e.g., the channel SNRs, the distances between the SUs and the PU) in the second and the third layers of the image, and performing some modifications (e.g., permutation, repetition) to the original data structure to provide the CNN with more features.

4.2.2. Network Training and PU Status Prediction

The local sensing decisions from the SUs, $D_i \ \forall i \in \{1, 2, \ldots, K\}$, are used as input for the CNN. Because a CNN is mostly used for image classification, the local decisions from K secondary users are rearranged to form a grayscale image with the size of $1 \times K \times 1$, where the last figure describes the number of color channels in the image. A stochastic gradient descent (SGD) optimizer with an adaptive learning rate is used in training the network. With this algorithm, the initial learning rate of 0.01 is later reduced based on a pre-defined schedule. For instance, it can be multiplied by a factor of 0.1 after every 10 epochs. The training set is a collection of local decisions from K SUs under different environmental conditions (i.e., a wide range in the sensed channel gain).

The FC uses the historical sensing data to train the CNN for the classification problem in advance. Thereafter, the FC determines the presence of the primary user on the licensed channel in every time slot by using the new individual sensing outcomes received at the beginning of each time slot as input for the trained network.

5. Transfer Learning Actor–Critic Framework for Data Protection in Cognitive Radio Networks

In this section, we present an optimal operation mode-decision policy based on an actor–critic learning framework so the SU can maximize the system's security level and energy utilization. Subsequently, the SU can encrypt data using the AES algorithm with a suitable key length before transmitting the secured information to the FC; or it could stay inactive in a time slot to save energy. In particular, if the SU does not have enough energy to transmit data, or if the sensing result indicates the PU is in state A, the SU will stay silent during the remainder of the time slot. Otherwise, it can decide to transmit the data encrypted by the AES algorithm with one of the three key lengths, $Nk \in \{128, 192, 256\}$, considering the effect of the decision on the long-term security level of the system.

5.1. Markov Decision Process

The problem of the operation mode decision in this paper is first formulated as a framework of a Markov decision process that is defined as a tuple $\langle \mathbb{S}, \mathbb{A}, \mathbb{P}, \mathbb{R} \rangle$, where $\mathbb{S}$ is the state space, $\mathbb{A}$ is the action space, $\mathbb{P} : \mathbb{S} \times \mathbb{A} \mapsto \mathbb{S}$ is a transition probability function, and $\mathbb{R}$ is the reward space. The state of the SU at the tth time slot is defined as $s(t) = (e_r(t), \rho(t))$, where $e_r(t)$ is the remaining energy of the SU, and $\rho(t)$ is the probability (also called *belief*) that the PU is inactive in that time slot. The action state space is defined as $\mathbb{A} = \{ID, TR_{Nk}\}$. At the tth time slot, the SU can choose to stay idle (action $a(t) = ID$) or it can choose to transmit data encrypted by the AES algorithm with key length $Nk \in \{128, 192, 256\}$ (action $a(t) = TR_{Nk}$). This action provides an immediate reward, and causes the SU to transit into a new state, s', with the following transition probability:

$$P(s'|s(t), a(t)) = \begin{cases} 1, & if\ s' = s(t+1), \\ 0, & otherwise. \end{cases} \tag{11}$$

We denote as $R(s(t), a(t))$ the reward (i.e., security level) achieved at the tth time slot when the SU is in state $s(t)$ and taking action $a(t) \in \mathbb{A}$, which is defined as

$$R(s(t), a(t)) \in \{0, S_{Nk}\}, \tag{12}$$

where

- R = 0 if the SU stays idle, or the transmission is not successful.
- R = 10 if the transmission is successful, and the data are encrypted by AES-128.
- R = 12 if the transmission is successful, and the data are encrypted by AES-192.
- R = 14 if the transmission is successful, and the data are encrypted by AES-256.

The value function is defined as the total discounted reward from the tth time slot, when the SU's state is $s(t) = s$, which is given as follows [22]:

$$V(s) = \sum_{k=t}^{\infty} \eta^k R(s(k), a(k)) | (s(t) = s), \tag{13}$$

where η is the discount factor. The objective of this paper is to find an optimal action for the SU in the tth time slot to maximize the value function as

$$a(t) = \arg\max_{a(k) \in \mathbb{A}} \left\{ \sum_{k=t}^{\infty} \eta^k R(s(k), a(k)) | (s(t) = s) \right\}. \tag{14}$$

The solution to the problem of the operation mode decision can be found by solving this equation.

5.2. Transfer Learning Actor–Critic Algorithm

Previous work proposed a POMDP-based approach to solving the problem in Equation (14) on the assumption that the SU already has information about the harvested energy model. In this paper, we introduce a new solution to the problem based on the actor–critic learning framework, which does not require the SU to already know the dynamics of energy harvesting. Instead, the SU determines those dynamics by directly interacting with the environment. A regular actor–critic model comprises three main elements: an actor (related to a learning policy), a critic (related to a learning value function), and the environment, as shown in Figure 5.

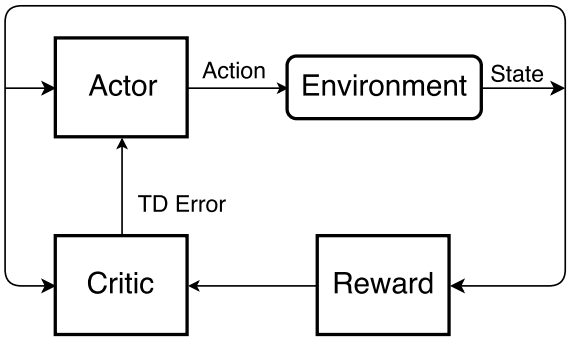

Figure 5. A regular actor–critic model. TD: temporal difference.

At time step t, the actor selects action $a(t)$ based on the current state, $s(t)$, and the policy, $\pi(s(t))$, which is defined by using a Gibbs softmax function as follows [22]:

$$\pi(s, a) = P(a(t) = a | s(t) = s) = \frac{e^{\theta(s,a)}}{\sum_{a' \in \mathbb{A}} e^{\theta(s,a')}}, \tag{15}$$

where $\theta(s, a)$ is the tendency to select action a when the SU is in state s. The final objective of this paper is now to find an optimal mode decision policy for the SU at the tth time slot, and the problem in Equation (14) can be rewritten as

$$\pi^*(s) = \underset{a \in \mathbb{A}}{\operatorname{argmax}} \left\{ R(s, a) + \eta \sum_{s' \in \mathbb{S}} P(s' | s, a) V^*(s') \right\}, \tag{16}$$

where $P(s' | s, a)$ is the transition probability from state s to state s' after taking action a.

After that, the SU transits into a new state, $s(t+1)$, and receives an instant reward $R(s(t), a(t))$. The critic evaluates the new state and computes a temporal difference (TD) error as

$$\delta(t) = R(s(t), a(t)) + \eta V(s(t+1)) - V(s(t)). \tag{17}$$

The critic uses the TD error to improve the estimate of the value function as well as the policy. The value function is updated as

$$V(s(t)) \leftarrow V(s(t)) + \alpha_c \cdot \delta(t), \tag{18}$$

where α_c is a positive parameter of the critic. The action resulting in a positive TD error is favorable, since the state value is better than expected. Hence, the probability of selecting action $a(t) = a$ in state $s(t) = s$ in the future should increase, and vice versa. Following that, the tendency to select this action is updated as

$$\theta(s(t), a(t)) \leftarrow \theta(s(t), a(t)) + \alpha_a \cdot \delta(t), \tag{19}$$

where α_a is a positive parameter of the actor.

Furthermore, we exploit the idea of transfer learning to increase the convergence speed to the optimal solution by making use of historical learning data, as depicted in Figure 6.

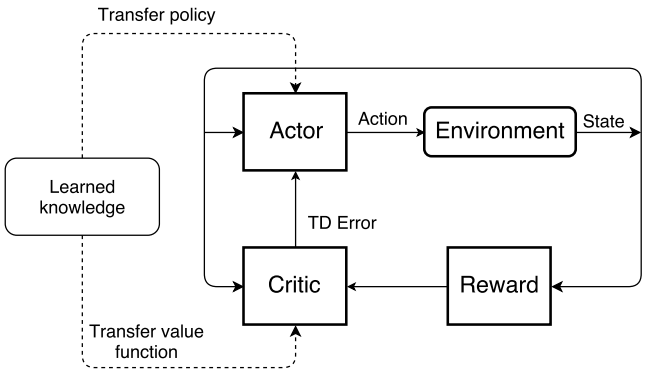

Figure 6. The transfer learning actor–critic model.

The obtained information is transferred to the new actor–critic algorithm for real-time training in which the initialized value function is the same as the transferred function while the overall policy, $\theta_o(s(t), a(t))$, for choosing an action at time step t is given as

$$\theta_o(s(t), a(t)) = \varepsilon(t)\theta_l(s(t), a(t)) + (1 - \varepsilon(t))\theta_n(s(t), a(t)), \tag{20}$$

where $\theta_l(s(t), a(t))$ is the transferred policy; $\theta_n(s(t), a(t))$ is the new policy, which will be updated in every time slot by using Equation (19); and $\varepsilon(t)$ is the transfer rate, which will be reduced after each time step to gradually remove the effect of the transferred policy on the new one.

The training process of the actor–critic learning framework for the SU to decide its operation mode is illustrated as follows. At the beginning of the tth time slot, the SU chooses an action according to policy π considering the sensing result and the remaining energy in its battery. The SU can decide to stay idle, $a(t) = ID$, to save energy, or it can transmit the encrypted data, $a(t) = TR_{Nk}$, to the FC. The immediate reward, $R(s(t), a(t))$, and the next state, $s(t+1)$, are updated at the end of the time slot based on the following cases.

5.2.1. Case 1

The sensing result shows that the PU is in state A on the primary channel, so the SU has to stay idle. Thus, no reward is achieved: $R(s(t), ID) = 0$. The belief that the PU is inactive in the current time slot is updated using Bayes' rule [36] as follows:

$$\rho^*(t) = \frac{\rho(t)P_f}{\rho(t)P_f + (1-\rho(t))P_d}. \tag{21}$$

The belief for the next time slot is given as

$$\rho(t+1) = \rho^*(t)P_{\bar{A}\bar{A}} + (1-\rho^*(t))\,P_{A\bar{A}}, \tag{22}$$

and the remaining energy that the SU can use for the next time slot is

$$e_r(t+1) = \min(e_r(t) + e_h(t) - E_s, E_{ca}), \tag{23}$$

where E_s is the total energy consumption for spectrum sensing, including the energy consumption from local spectrum sensing and that from sending the sensing outcomes to the fusion center.

5.2.2. Case 2

The sensing result indicates that the PU is absent from the primary channel. There are two possible occurrences:

(1) The SU decides to stay idle to save energy for the next time slot.
(2) The SU transmits encrypted information to the fusion center.

In the first occurrence, there is no reward: $R(s(t), ID) = 0$. The probability that the PU is truly inactive in the current time slot is updated using Bayes' rule as follows:

$$\rho^*(t) = \frac{\rho(t)(1-P_f)}{\rho(t)(1-P_f) + (1-\rho(t))(1-P_d)}. \tag{24}$$

The belief and the remaining energy for the next time slot are calculated by using Equations (22) and (23), respectively.

Regarding the second occurrence, the SU uses $e_{tr}(t)$ packets of energy to transmit the encrypted data to the FC. The remaining energy of the SU for the next time slot is calculated as follows:

$$e_r(t+1) = \min(e_r(t) + e_h(t) - e_{tr}(t) - E_s - E_{Nk}, E_{ca}), \tag{25}$$

where E_{Nk} is the energy consumption for the encryption process, which is dependent on the key length Nk of the encryption algorithm. If the SU does not receive an acknowledgement (ACK) from the FC, which means the transmission was unsuccessful, there is no reward: $R(s(t), TR_{Nk}|\overline{ACK}) = 0$. The probability that the channel will be free of the PU signal in the next time slot is given as

$$\rho(t+1) = P_{A\bar{A}}. \tag{26}$$

On the other hand, if the SU receives ACK from the FC, indicating that transmission was successful, the reward is

- $R(s(t), TR_{Nk}|ACK) = 10$ if $Nk = 128$,
- $R(s(t), TR_{Nk}|ACK) = 12$ if $Nk = 192$,
- $R(s(t), TR_{Nk}|ACK) = 14$ if $Nk = 256$.

The belief that the PU will be absent from the channel in the next time slot is given by

$$\rho(t+1) = P_{\bar{A}\bar{A}}. \tag{27}$$

Thereafter, the value function and the new policy are updated based on the received reward and the new state. This process repeats until it converges into the optimal solution that maximizes the long-term reward of the system, which means that value function $V(s)$ and policy $\pi(s)$ will finally converge to $V^*(s)$ and $\pi^*(s)$ as $k \to \infty$ [37].

6. Results and Discussion

In this section, we present simulation results to demonstrate the efficiency of the proposed CBCSS and TLAC algorithms for energy-efficient data protection in CRNs. We first present simulation results to evaluate the performance of the proposed CBCSS technique compared with other fusion techniques, such as a half-voting rule [38], an energy detection (ED) method performed by a secondary user, and the Chair–Varshney rule [39]. We then investigate the potential of the TLAC solution for establishing an operation mode decision policy by comparing it with the POMDP-based solution from earlier work [20], the myopic scheme, and the fixed encryption methods, which will be described in detail later.

6.1. Convolutional Neural Network-Based Cooperative Spectrum Sensing

The proposed CBCSS for the two-state classification problem was implemented using the Neural Network Toolbox in Matlab (R2017a, The MathWorks Inc., Natick, MA, USA, 2017). Unless presented otherwise, the simulation parameters were as listed in Table 2. The average SNR of the sensed channel, γ_i, that was used for training the CNN ranged from -16 dB to -6 dB. Furthermore, the number of training samples for each SNR was 2000.

Table 2. Simulation parameters for the convolutional neural network (CNN) -based cooperative spectrum sensing (CBCSS) scheme.

Symbol	Description	Value
K	The number of secondary users	10
N_i	The number of sensing samples collected by each secondary user	300
γ_i	Average signal-to-noise ratio (SNR) of the sensed channel that was used for training the CNN (dB)	-16 to -6
$P_{A\bar{A}}$, $P_{\bar{A}A}$	Probability of the primary user's state transition from A to $\bar{A}$, and vice versa	0.2

In this work, we consider three different performance metrics: probability of detection P_d, probability of false alarm P_f, and sensing error P_e. The total number of time slots for testing the performance of the proposed CBCSS was 10,000. Furthermore, the process was performed 10 times to get average values for P_d, P_f, and P_e. The first two parameters are calculated by using Equations (1) and (2), whereas sensing error is defined as the sum of the probability of false alarm (P_f) and the probability of missed detection ($1 - P_d$), as follows:

$$P_e = P_f + (1 - P_d). \tag{28}$$

In Figures 7 and 8, we compare the performance of the proposed CBCSS with those of the conventional half-voting fusion rule for cooperative spectrum sensing, the local sensing result based on the energy detection method from one of the $K = 10$ secondary users, and the Chair–Varshney fusion rule.

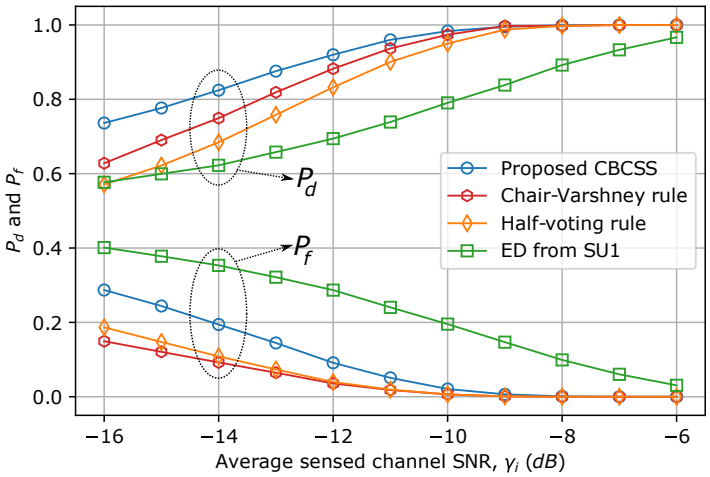

Figure 7. Probabilities of detection and false alarm according to average SNRs for different sensing schemes. ED: energy detection.

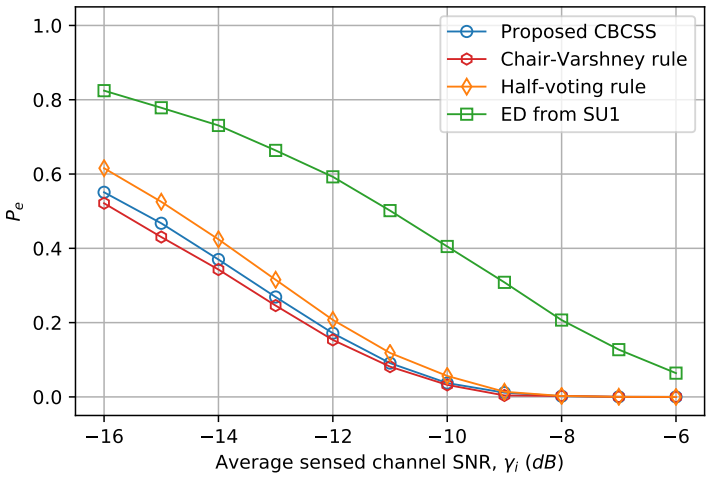

Figure 8. Sensing error according to average SNRs for different sensing schemes.

Regarding the half-voting rule, the fusion center makes a global decision based on the local sensing data. Specifically, the FC decides that the PU is active (A) if at least half of K SUs report the decision $D_i = 1$. With respect to the energy detection method, the local decision from SU1 was obtained for comparison. Under the Chair–Varshney rule, the detection statistics are expressed as the weighted sum of the local decisions; and the weights are functions of detection probability and false alarm [40]. The Chair–Varshney rule is the optimal decision fusion rule but requires a prior knowledge of the PU's activities and the local sensing performance of the secondary users. From the figures, we can confirm that the proposed CBCSS outperforms other conventional methods, except for the Chair–Varshney optimal fusion rule, in terms of detection probability and sensing error. We can also see that with an increment in the average SNR, the probability of detection increases while the probability of false alarm and the sensing error decrease. This is because the effect of AWGN on the local decisions, and thus the training accuracy, decreases as SNR increases. Accordingly, larger sensed channel SNRs at the SUs provide better detection performance and fewer false alarms. Although the probability of false alarm with the proposed scheme is a little higher than with the half-voting and the Chair–Varshney rules, the total sensing error of the proposed CBCSS almost reaches to that of the Chair–Varshney optimal fusion rule and is lower than those of conventional methods.

In Figures 9 and 10, we examine the effect of the number of secondary users, K, on the performance of the proposed CBCSS. To verify this, we evaluated the output results from three distinct CNNs

that were trained with $K \in \{5, 10, 20\}$, while keeping the number of sensing samples unchanged at $N_i = 300$. For each value of K, the performance metrics were calculated again for comparison purposes. As can be seen from the figures, the increases in the number of SUs that cooperate in spectrum sensing can significantly improve the performance of the CBCSS. This is caused by the increase in spatial diversity when using more SUs, which can help the CNN to extract more information from the sensing data. Moreover, in Figure 10, there is almost no sensing error at $SNR = -10$ dB with $K = 20$ sensing nodes.

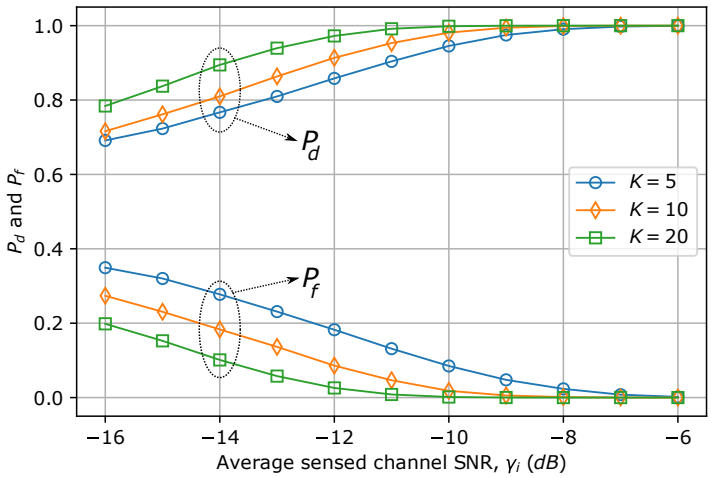

Figure 9. Probabilities of detection and false alarm with the proposed CBCSS according to average SNRs when the number of SUs, K, changes.

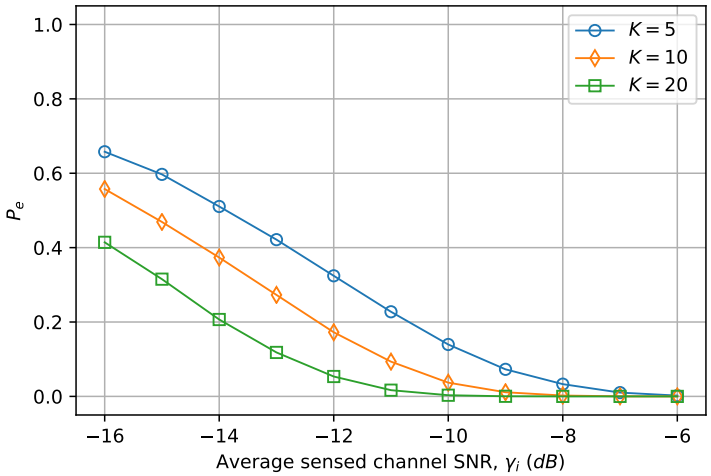

Figure 10. Sensing error with the proposed CBCSS according to average SNRs when the number of SUs, K, changes.

Finally, we measured the performance of the CBCSS by varying the number of sensing samples, N_i, as shown in Figures 11 and 12, for $K = 10$ secondary users. The training process is the same as with the changing K, but now the number of sensing samples is varied instead of K: $N_i \in \{200, 300, 400\}$. We assert that the effectiveness of the new cooperative spectrum sensing system can be improved by increasing the number of sensing samples that are collected by the SUs for individual spectrum sensing using the energy detection method. Again, the larger value of γ_i provides better detection accuracy as well as a lower sensing error.

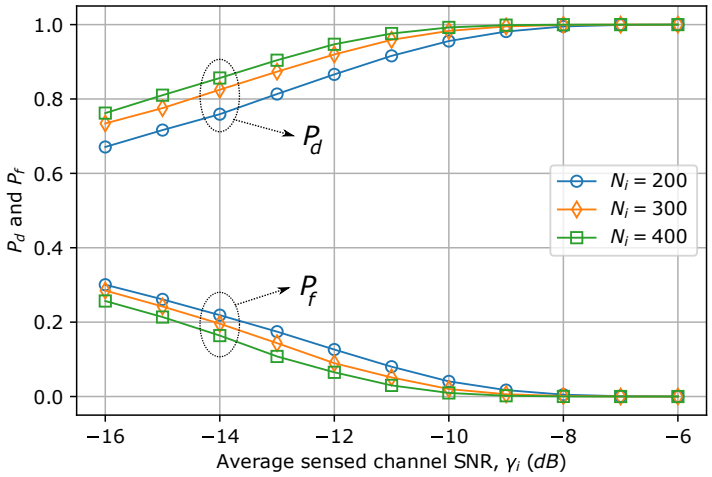

Figure 11. Probabilities of detection and false alarm according to average SNRs when the number of sensing samples for each SU changes.

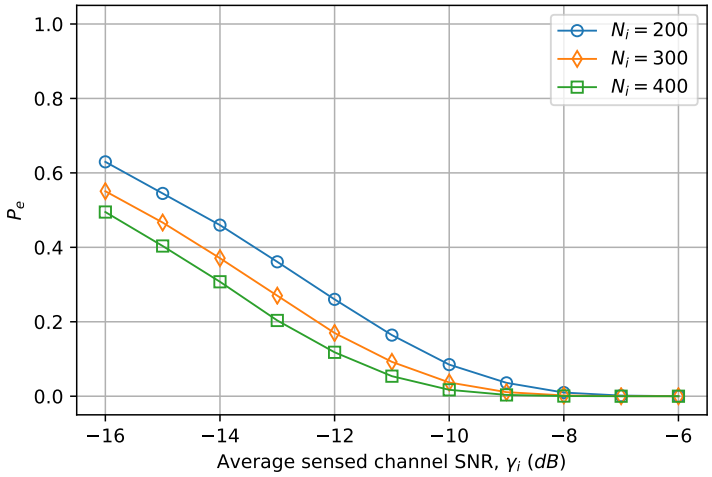

Figure 12. Sensing error according to average SNRs when the number of sensing samples for each SU, N, changes.

Since in the paper we focus on developing a new CNN-based cooperative spectrum sensing technique, for the sake of simplicity, we use a simple energy detection method for local spectrum sensing. However, the sensing efficiency can be further enhanced by improving the local spectrum sensing. That is, if the local sensing outcomes provide more accurate sensing data, the CNN can learn the features of the data with higher accuracy, which will produce more precise classification results. From the simulation results, we can observe that larger values of the channel SNR can ensure the better local sensing results, which leads to better overall sensing performance of the system.

6.2. Transfer Learning Actor–Critic Solution for Energy-Efficient Data Protection Scheme

This section verifies the performance of the proposed actor–critic framework in comparison to the myopic solution and the POMDP-based solution from earlier work. With regard to the myopic scheme, if the PU is found absent from the channel, the SU will sacrifice its energy to maximize data security [41]. Previous work proposed an optimal decision policy for a CRN to maximize the security level based on a POMDP framework, which requires complex numerical computations as well as prior information about the arrival of harvested energy [20]. The complexity of the problem depends on the required amount of the computation *space* (e.g., the sizes of the input states, actions, transition probabilities, and observations). In a POMDP, an agent controls the process by choosing the action at each time

step based on the observation history to maximize the expected long-term reward. The optimal policy for the agent to choose an action can be found by solving the Bellman's equation using value iteration-based dynamic programming. Each iteration requires $O(|\mathbb{A}||\mathbb{S}|^2)$ operations to compute all the probabilities of transitioning from one state, $s \in \mathbb{S}$, to another state, $s' \in \mathbb{S}$, after taking an action, $a \in \mathbb{A}$. The actor–critic method, on the other hand, does not require the agent to compute all the occurrence probabilities to find the solution in advance. In addition to that, the agent learns the optimal policy from actual experienced transitions by directly interacting with the stochastic environment.

The basic simulation parameters for this exercise are shown in Table 3. For analytic convenience, we fixed the SNR value of the sensed channel at -10 dB, and thus the probabilities of detection and false alarm are approximated as $P_d \approx 0.9$ and $P_f \approx 0.1$, respectively (based on the results of the proposed CBCSS method). We assume that the SU transmits a packet of 16-byte data in every time slot, which is equivalent to the minimum encryption block length in the AES cryptography; and the transmission channel gain is unchanged during a time slot. It is worth noting that one packet of energy is equivalent to 25 μJ, and each simulation was run over a thousand of time slots for several iterations to obtain average values.

Table 3. Simulation parameters for transfer learning actor–critic (TLAC) [20].

Symbol	Description	Value
γ_i	Average SNR of the sensed channel (dB)	-10
$P_{A\bar{A}}$, $P_{\bar{A}A}$	Transition probabilities between states (A and $\bar{A}$) of the primary user	0.2
E_{ca}	Battery capacity (packets)	160
E_s	Energy consumption for the whole spectrum sensing process (packets)	1
E_{Nk}	Energy consumption for data encryption using the Advanced Encryption Standard (AES) algorithm with key length $Nk \in \{128, 192, 256\}$ (packets)	$\{4, 6, 8\}$
$e_{h,avg}$	Average harvested energy (packets)	$\{2, 4, 6, 8, 10\}$
e_{tr}	Average energy consumption for data transmission (packets)	40
η	Discount factor	0.9
α_c	Critic learning rate	0.2
α_a	Actor learning rate	0.1
$\epsilon(0)$	Initial transfer rate	0.5

We first examined the convergence speed of the TLAC algorithm during the training process by calculating the average reward received after every 1000 time slots. The average harvested energy was fixed at $e_{h,avg} = 4$ packets. As can be seen from Figure 13, there is a significant rise in the convergence rate of the algorithm during the first 10,000 time slots of the training process; after that, the reward keeps increasing, but at a slower speed. Finally, the algorithm converges to an optimal policy for the SU to determine operation mode after 20,000 time slots when the reward is about 0.91.

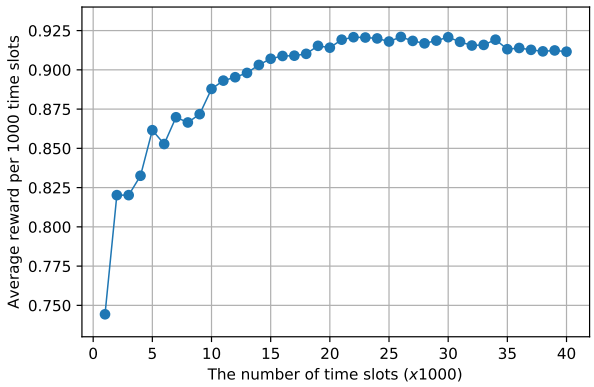

Figure 13. Actor–critic training convergence rate, $e_{h,avg} = 4$.

In Figure 14, we show the efficiency of the proposed scheme compared with the POMDP-based and myopic schemes under the effect of harvested energy. As can be seen from the figure, a larger harvested energy yields a higher reward, indicating that data are protected better. The reason is that, if the SU can harvest more energy, it has a greater chance to operate in transmission mode, and can transmit more data to the FC. Furthermore, the result of the proposed TLAC algorithm is better than the myopic one and a little lower than the POMDP method. To explain this, in the myopic scheme, the SU makes a decision on its working mode without considering the effect of this action on the future reward. In particular, if the primary channel is found free via spectrum sensing, the SU uses too much energy for data encryption to enhance data protection, which causes the SU to stay in idle mode over many time slots due to limited remaining energy. Regarding the POMDP-based solution, the SU is assumed to already have information about the harvested energy model, which is hardly ever true in practice. As a result, by using value iteration-based programming, we can compute all possible happening states and the corresponding occurrence probabilities to find the optimal policy beforehand. Consequently, the SU can predict the next state of the primary user and the upcoming harvested energy before effectively distributing the energy over future time slots. Meanwhile, employing the TLAC algorithm requires the SU to frequently interact with the environment to determine the dynamics of the arrival of harvested energy, which can result in a locally optimal policy [22]. In particular, the SU makes decisions based on a predefined policy (i.e., local or immediate consideration), which is updated at the end of every time slot, to improve future behavior without needing to have any information about the environment's dynamics.

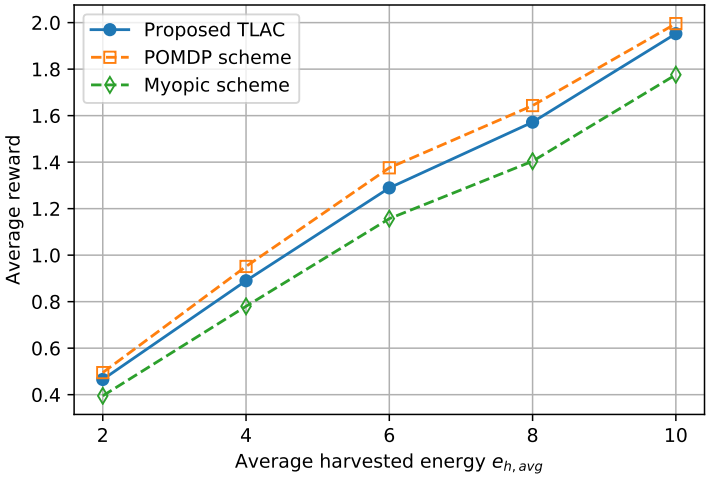

Figure 14. Average reward according to harvested energy for different data protection schemes. POMDP: partially observable Markov decision process.

Figure 15 illustrates the channel utilization by the SU for its data communications, computed as the ratio of the total number of successful data transmissions to the total time slots in which the primary user is sensed as inactive. From the figure, we can see that the primary channel is utilized more effectively when harvested energy $e_{h,avg}$ increases. In addition, the proposed TLAC algorithm utilizes the free channel better than the myopic scheme about 2% of the total successful transmissions. We can also see that the POMDP technique provides an optimal solution to the problem of the operation mode decision. However, the TLAC solution without requiring too much effort in mathematical computation or prior information about the environment's dynamics can provide the SU with a locally optimal policy that almost reaches the result of the POMDP scheme, especially when the amount of harvested energy is large. This is because the SU can encrypt data with a longer key size (e.g., $Nk = 256$) by utilizing extra energy in the battery when the average harvested energy increases. Therefore, the policy would be updated to favor the action that gives a better reward in the future.

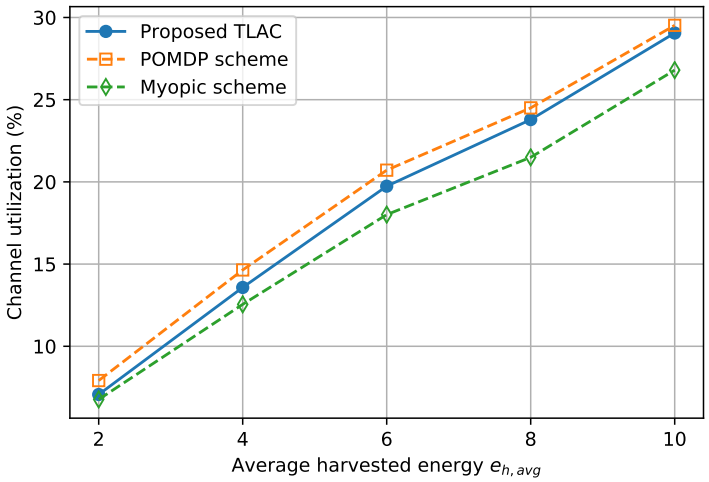

Figure 15. Channel utilization according to the harvested energy for different data protection schemes.

Figure 16 depicts the total number of data packets transmitted from the SU to the fusion center based on harvested energy under three different data protection schemes. As can be seen from the figure, the SU can transmit more packets of data when using the TLAC algorithm, compared to the myopic scheme. The reason is that the proposed learning scheme can allocate the harvested energy more efficiently than the myopic one. Consequently, the SU can operate in transmission mode in more time slots, and thus, can transmit more encrypted data packets to the FC. Meanwhile, using the myopic scheme can cause the SU to be inactive due to lack of energy for future use. For that reason, the proposed TLAC framework can guarantee the security level, and can effectively utilize the limited energy resource.

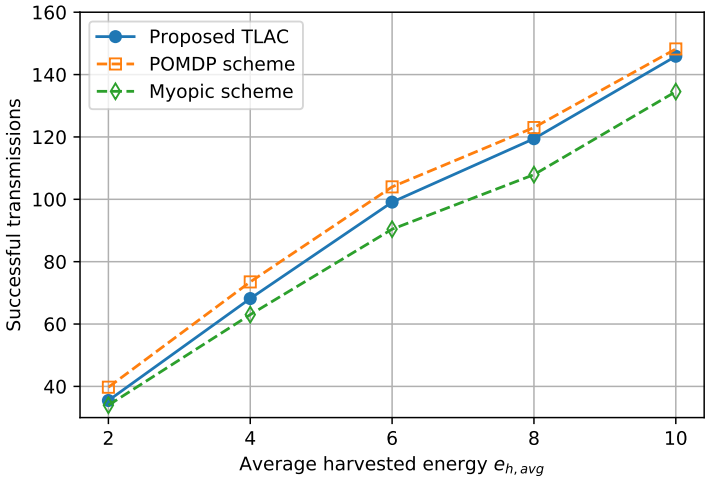

Figure 16. The number of successfully transmitted data packets according to the harvested energy for different data protection schemes.

More specifically, in Figure 17, we present the detailed number of successfully transmitted data packets that are encrypted using the AES algorithm with different key lengths. We can see from the figure that the total number of data packets delivered under the TLAC algorithm is 10% higher than when using the myopic scheme. In particular, more packets are encrypted with longer key sizes (i.e., AES-192 or AES-256) with a rise in the arrival of harvested energy.

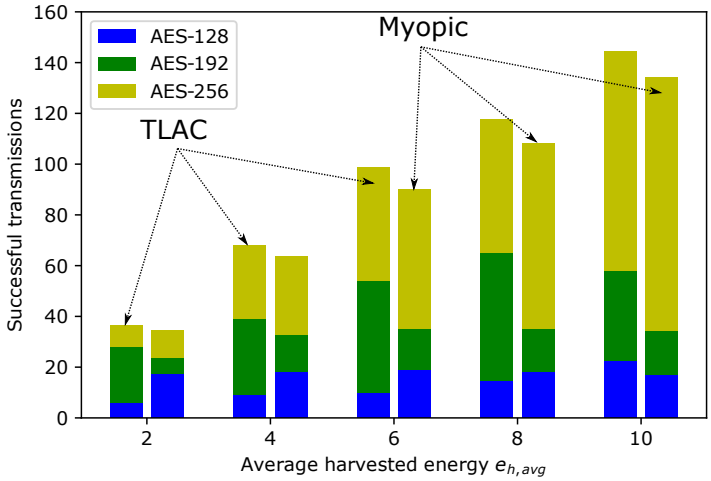

Figure 17. Comparison between the proposed TLAC and the myopic schemes, based on harvested energy.

Finally, we examine the performance of the proposed TLAC by comparing it with that of AES algorithms with fixed key length. In the fixed key length schemes, the SU uses only one key size to encrypt data at each time step even when it has enough energy. In Figure 18, the rewards under the proposed TLAC and other schemes grow persistently with the increment in the harvested energy. While the proposed solution provides the highest average reward, the fixed encryption method with the shortest cipher key (AES-128) shows the lowest security level. The reason is that the proposed method can efficiently allocate the energy to every time slot by estimating expected reward in the future time slots. Meanwhile, the AES-128 algorithm always uses the lowest amount of energy for data encryption, and thus, does not utilize the redundant energy in the SU's battery to enhance the security level as the arrival speed of the harvested energy increases.

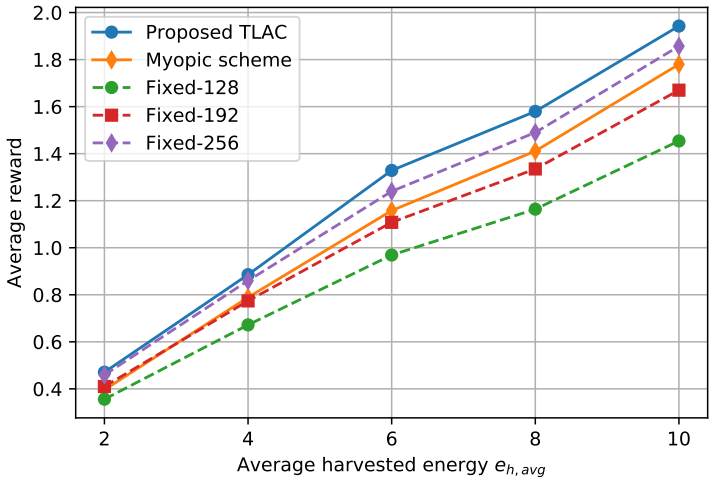

Figure 18. Reward comparison between the proposed TLAC and the fixed key-length schemes according to the harvested energy.

On the other hand, the AES-256 uses maximum energy to encrypt data whenever the energy is sufficient to increase data security. However, this action reduces the chance for the SU to operate in transmission mode, which leads to low successful transmissions, as shown in Figure 19. From the figure, we can see that the proposed TLAC provides the SU with the highest channel utilization since the SU can transmit more data packets in comparison to other methods. This is because the fixed encryption techniques do not utilize the energy effectively for future use. Among those fixed

encryption methods, the AES-128 with lower energy consumption allows the SU to transmit more data packets than the AES-192 and the AES-256, but provides the SU with the lowest reward. Consequently, we can verify that the proposed TLAC algorithm can ensure effective data communications between the SU and the fusion center in terms of security level and channel utilization.

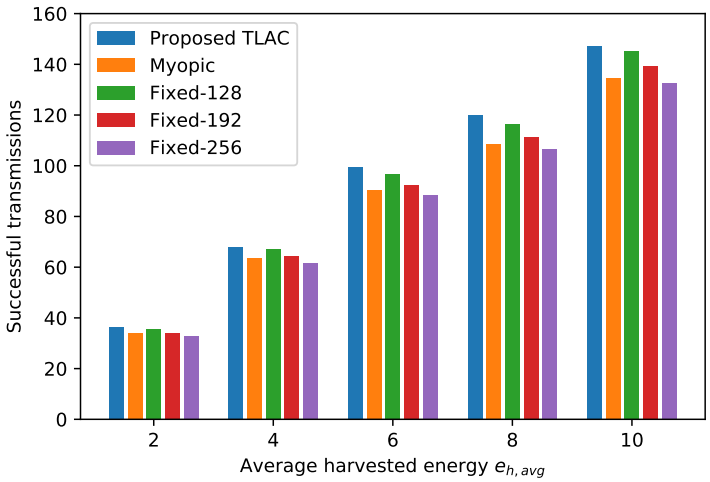

Figure 19. The number of successfully transmitted data packets according to the harvested energy, compared with the fixed key-length encryption methods.

7. Conclusions

In this paper, we propose learning-based techniques for cooperative spectrum sensing and energy-efficient data protection in CRNs, by which the SUs can effectively utilize the primary channel under the constraint of limited harvested energy. We first design a new CNN-based cooperative spectrum sensing method. In this approach, the CNN is trained by using historical sensing data collected from secondary users under various environmental conditions. At the beginning of each time slot, the SUs individually perform spectrum sensing using an energy detection method, and then send the local decisions to a fusion center to make a global decision about the state of the primary user. The proposed CBCSS can increase the detection probability and remarkably reduce the sensing error, which can also contribute to effective communications between the SUs and the fusion center. Regarding the proposed TLAC scheme, the SU determines its operation mode based on the remaining energy and the sensing result considering the effect of this decision on future time slots. By calculating the expected accumulated reward from the current time slot, the SU can decide to stay in idle mode to save energy for future use, or operate in transmission mode and transmit cypher data that are protected by using the AES algorithm with an appropriate key length. We then present simulation results to evaluate the performance of the proposed solutions, which show that the proposed schemes can guarantee energy-efficient data communications in cognitive radio networks.

However, there are still some areas of the proposed learning frameworks that can be of interest for future research. First, it is possible to improve the performance of the current CNN-based cooperative spectrum sensing technique by modifying the structure of the input data. In addition, the local sensing results from K secondary users, other information such as the sensed channel SNRs, sensing duration and even the distances between the SUs and the PU can be used as the input data for the CNN to predict the state of the primary user. Those information sources could provide the CNN with more useful features for reducing the negative effect of the noise on the local sensing outcomes. Secondly, with respect to the TLAC framework, it is essential to choose good learning-rate parameters that can balance the convergence speed and the computational resource. Furthermore, the current actor–critic framework would be further extended to apply to the problems with large or continuous domains instead of discrete-time state space and action space.

Author Contributions: All authors conceived and proposed the research idea. Vinh Quang Do designed and performed the experiments; Insoo Koo analyzed the experimental results; Vinh Quang Do wrote the paper under the supervision of Insoo Koo.

Acknowledgments: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2015R1D1A1A09057077) as well as by the Korea government (MSIT) (2018R1A2B6001714).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Joshi, G.; Nam, S.; Kim, S. Cognitive Radio Wireless Sensor Networks: Applications, Challenges and Research Trends. *Sensors* **2013**, *13*, 11196–11228. [CrossRef] [PubMed]
2. Park, S.; Hong, D. Optimal Spectrum Access for Energy Harvesting Cognitive Radio Networks. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 6166–6179. [CrossRef]
3. Park, S.; Kim, H.; Hong, D. Cognitive radio networks with energy harvesting. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 1386–1397. [CrossRef]
4. Pappas, N.; Jeon, J.; Ephremides, A.; Traganitis, A. Optimal utilization of a cognitive shared channel with a rechargeable primary source node. *J. Commun. Netw.* **2012**, *14*, 162–168. [CrossRef]
5. Sultan, A. Sensing and Transmit Energy Optimization for an Energy Harvesting Cognitive Radio. *IEEE Wirel. Commun. Lett.* **2012**, *1*, 500–503. [CrossRef]
6. Razaque, A.; Elleithy, K.M. Energy-Efficient Boarder Node Medium Access Control Protocol for Wireless Sensor Networks. *Sensors* **2014**, *14*, 5074–5117. [CrossRef] [PubMed]
7. Liang, Y.-C.; Zeng, Y.; Peh, E.; Huang, A.T. Sensing-Throughput Tradeoff for Cognitive Radio Networks. *IEEE Trans. Wirel. Commun.* **2008**, *7*, 1326–1337. [CrossRef]
8. Lee, S.; Zhang, R.; Huang, K. Opportunistic Wireless Energy Harvesting in Cognitive Radio Networks. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 4788–4799. [CrossRef]
9. Rossi, P.S.; Ciuonzo, D.; Romano, G. Orthogonality and Cooperation in Collaborative Spectrum Sensing through MIMO Decision Fusion. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 5826–5836. [CrossRef]
10. Holcomb, S.; Rawat, D.B. Recent security issues on cognitive radio networks: A survey. In Proceedings of the IEEE SoutheastCon 2016, Norfolk, VA, USA, 30 March–3 April 2016; pp. 1–6.
11. Fragkiadakis, A.G.; Tragos, E.Z.; Askoxylakis, I.G. A Survey on Security Threats and Detection Techniques in Cognitive Radio Networks. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 428–445. [CrossRef]
12. Wen, H.; Li, S.; Zhu, X.; Zhou, L. A framework of the PHY-layer approach to defense against security threats in cognitive radio networks. *IEEE Netw.* **2013**, *27*, 34–39.
13. Ciuonzo, D.; Aubry, A.; Carotenuto, V. Rician MIMO Channel- and Jamming-Aware Decision Fusion. *IEEE Trans. Signal Process.* **2017**, *65*, 3866–3880. [CrossRef]
14. Xu, X.; He, B.; Yang, W.; Zhou, X.; Cai, Y. Secure Transmission Design for Cognitive Radio Networks with Poisson Distributed Eavesdroppers. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 373–387. [CrossRef]
15. Elkashlan, M.; Wang, L.; Duong, T.Q.; Karagiannidis, G.K.; Nallanathan, A. On the Security of Cognitive Radio Networks. *IEEE Trans. Veh. Technol.* **2015**, *64*, 3790–3795. [CrossRef]
16. Wang, B.; Zhan, Y.; Zhang, Z. Cryptanalysis of a Symmetric Fully Homomorphic Encryption Scheme. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1460–1467. [CrossRef]
17. Sultan, A.; Yang, X.; Hajomer, A.A.; Hu, W. Chaotic Constellation Mapping for Physical-Layer Data Encryption in OFDM-PON. *IEEE Photonics Technol. Lett.* **2018**, *30*, 339–342. [CrossRef]
18. Angizi, S.; He, Z.; Bagherzadeh, N.; Fan, D. Design and Evaluation of a Spintronic In-Memory Processing Platform for Non-Volatile Data Encryption. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2017**. [CrossRef]
19. Sen, J. A Survey on Security and Privacy Protocols for Cognitive Wireless Sensor Networks. *J. Netw. Inf. Secur.* **2013**, *1*, 1–43.
20. Do-Vinh, Q.; Hoan, T.N.K.; Koo, I. Energy-Efficient Data Encryption Scheme for Cognitive Radio Networks. *IEEE Sens. J.* **2018**, *18*, 2050–2059. [CrossRef]
21. NIST Standards. *Advanced Encryption Standard (AES)*; NIST Standards: Gaithersburg, MD, USA, 2001.
22. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; The MIT Press: London, UK, 1998; p. 331.

23. Berenji, H.; Vengerov, D. A convergent actor–critic-based FRL algorithm with application to power management of wireless transmitters. *IEEE Trans. Fuzzy Syst.* **2003**, *11*, 478–485. [CrossRef]
24. Taylor, M.E.; Stone, P. Transfer Learning for Reinforcement Learning Domains: A Survey. *J. Mach. Learn. Res.* **2009**, *10*, 1633–1685.
25. Lee, P.; Eu, Z.A.; Han, M.; Tan, H.P. Empirical modeling of a solar-powered energy harvesting wireless sensor node for time-slotted operation. In Proceedings of the 2011 IEEE Wireless Communications and Networking Conference, Cancun, Mexico, 28–31 March 2011; pp. 179–184.
26. Zhang, S.; Wang, H.; Zhang, X. Estimation of channel state transition probabilities based on Markov Chains in cognitive radio. *J. Commun.* **2014**, *9*, 468–474. [CrossRef]
27. Kim, J.M.; Lee, H.S.; Yi, J.; Park, M. Power Adaptive Data Encryption for Energy-Efficient and Secure Communication in Solar-Powered Wireless Sensor Networks. *J. Sens.* **2016**, *2016*, 1–9. [CrossRef]
28. Othman, S.B. Performance evaluation of encryption algorithm for wireless sensor networks. In Proceedings of the 2012 International Conference on Information Technology and e-Services (ICITeS), Sousse, Tunisia, 24–26 March 2012; pp. 1–8.
29. Medepally, B.; Mehta, N.B.; Murthy, C.R. Implications of Energy Profile and Storage on Energy Harvesting Sensor Link Performance. In Proceedings of the GLOBECOM 2009—2009 IEEE Global Telecommunications Conference, Honolulu, HI, USA, 30 November–4 December 2009; pp. 1–6.
30. Ho, C.K.; Zhang, R. Optimal Energy Allocation for Wireless Communications with Energy Harvesting Constraints. *IEEE Trans. Signal Process.* **2012**, *60*, 4808–4818. [CrossRef]
31. Tarighati, A.; Gross, J.; Jalden, J. Decentralized Hypothesis Testing in Energy Harvesting Wireless Sensor Networks. *IEEE Trans. Signal Process.* **2017**, *65*, 4862–4873. [CrossRef]
32. Zhang, W.; Mallik, R.; Letaief, K. Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 5761–5766. [CrossRef]
33. Atapattu, S.; Tellambura, C.; Jiang, H. Conventional Energy Detector. In *Energy Detection for Spectrum Sensing in Cognitive Radio*, 1st ed.; Springer: New York, NY, USA, 2014; Chapter 2, pp. 11–26.
34. Quan, Z.; Cui, S.; Sayed, A.H. Optimal Linear Cooperation for Spectrum Sensing in Cognitive Radio Networks. *IEEE J. Sel. Top. Signal Process.* **2008**, *2*, 28–40. [CrossRef]
35. Gulli, A.; Pal, S. *Deep Learning with Keras*; Packt Publishing Ltd.: Birmingham, UK, 2017; p. 318.
36. Stone, J.V. *Bayes' Rule: A Tutorial Introduction to Bayesian Analysis*, 2013 ed.; Sebtel Press: Sheffield, UK, 2013; p. 174.
37. Singh, S.; Jaakkola, T.; Littman, M.L.; Szepesvári, C. Convergence results for single-step on-policy reinforcement-learning algorithms. *Mach. Learn.* **2000**, *38*, 287–308. [CrossRef]
38. Teguig, D.; Scheers, B.; Nir, V.L. Data fusion schemes for cooperative spectrum sensing in cognitive radio networks. In Proceedings of the 2012 Military Communications and Information Systems Conference (MCC), Gdansk, Poland, 8–9 October 2012; pp. 1–7.
39. Vu-Van, H.; Koo, I. Cooperative spectrum sensing with collaborative users using individual sensing credibility for cognitive radio network. *IEEE Trans. Consum. Electron.* **2011**, *57*, 320–326. [CrossRef]
40. Li, L.C.; Wang, J.; Li, S. An Adaptive Cooperative Spectrum Sensing Scheme Based on the Optimal Data Fusion Rule. In Proceedings of the 4th International Symposium on Wireless Communication Systems, Trondheim, Norway, 17–19 October 2007; pp. 582–586.
41. Zhao, Q.; Krishnamachari, B.; Liu, K. On myopic sensing for multi-channel opportunistic access: Structure, optimality, and performance. *IEEE Trans. Wirel. Commun.* **2008**, *7*, 5431–5440. [CrossRef]