

Deep Recurrent Neural Network for Multiple Time Slot Frequency Spectrum Predictions of Cognitive Radio

Zhi-ling Tang^{1,2} and Si-min Li¹

¹ Guangxi Key Laboratory of Wireless Broadband Communication and Signal Processing, Guilin University of Electronic Technology
Guilin, GX 541004 - China

[e-mail: tzl888@guet.edu.cn; siminl@guet.edu.cn]

² CETC Key Laboratory of Aerospace Information Applications
Shijiazhuang, HB 050081 – China

* Corresponding author: Si-min Li;

*Received November 23, 2016; revised March 17, 2017; accepted March 22, 2017;
published June 30, 2017*

Abstract

The main processes of a cognitive radio system include spectrum sensing, spectrum decision, spectrum sharing, and spectrum conversion. Experimental results show that these stages introduce a time delay that affects the spectrum sensing accuracy, reducing its efficiency. To reduce the time delay, the frequency spectrum prediction was proposed to alleviate the burden on the spectrum sensing. In this paper, the deep recurrent neural network (DRNN) was proposed to predict the spectrum of multiple time slots, since the existing methods only predict the spectrum of one time slot. The continuous state of a channel is divided into a many time slots, forming a time series of the channel state. Since there are more hidden layers in the DRNN than in the RNN, the DRNN has fading memory in its bottom layer as well as in the past input. In addition, the extended Kalman filter was used to train the DRNN, which overcomes the problem of slow convergence and the vanishing gradient of the gradient descent method. The spectrum prediction based on the DRNN was verified with a WiFi signal, and the error of the prediction was analyzed. The simulation results proved that the multiple slot spectrum prediction improved the spectrum efficiency and reduced the energy consumption of spectrum sensing.

Keywords: cognitive radio, spectrum sensing, recurrent neural network, time series

This research was supported by the national natural science foundation of China (Grant No. 61461013), Program for Innovative Research Team of Guilin University of Electronic Technology (IRTGUET), CETC key laboratory of aerospace information applications (Grant No. EX166290017) and Dean Project of Guangxi Key Laboratory of Wireless Broadband Communication and Signal Processing (Grant No. GXKL06160103).

1. Introduction

The evolution of communication technology has resulted in growing data transmission rates. These increasing rates have impacted the spectrum resources for wireless communication, causing a bottleneck. According to statistics, there are different degrees of idle spectrum resources in time and space [1]. Fig. 1 shows the radio spectrum usage in 14 channels. In some channels, almost all slots are occupied by the primary users (PU), while the other channels have almost no users. The concept of a cognitive radio was proposed in order to use the radio spectrum more efficiently. In a cognitive radio network, when the spectrum assigned to a PU is not used, secondary users (SU) are allowed to use the spectrum, including the processes of spectrum sensing, spectrum decision, spectrum sharing, and spectrum conversion [2].

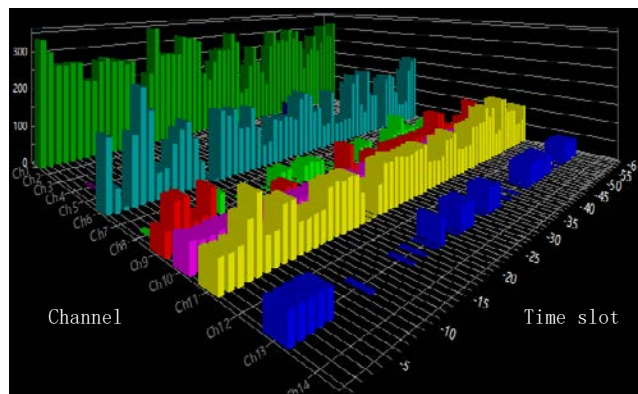


Fig. 1. The variation in the radio frequency spectrum resources over time.

This process is known as perception, and Haykin pointed out that a perception-action cycle is the key to the cognitive radio system [3]. At present, there are two main local spectrum sensing methods. The first is based on energy detection and it compares the average energy and threshold of the signal sampling to determine which spectrum it is using [4]. The second is based on feature detection: it looks for a peak in the cyclic spectrum calculated by the Fourier transform of the cyclic autocorrelation function, which indicates the spectrum is occupied [5-6]. However, the spectrum energy cannot be effectively detected by the first method because of the uncertainty in the noise power when the signal-to-noise ratio (SNR) is lower than a certain level. The second method requires more complex signal processing algorithms. Both of these methods need more time for signal processing. Furthermore, some experiments have shown that the spectrum decision, spectrum sharing, and spectrum conversion introduce time delays that deteriorate the accuracy and utilization of the spectrum sensing [7].

In order to solve this problem, a method for spectrum prediction was proposed. It predicts whether the PU will use the channel or determines the probability of the channel's use as it learns the history of the channel state, which allows SUs to skip sensing the busy channels to reduce the amount of time needed to find a channel [8].

The Neural Network (NN) and Hidden Markov Models (HMM) are the two main methods for spectrum prediction. In the NN methods, Multilayer Perceptron (MLP) can predict idle channels without any prior knowledge of the PU's traffic characteristics; numerical simulations have shown that the spectrum prediction can reduce power consumption and improve the utilization of the spectrum [9-11]. Nicola, Bheemarjuna and Manoj B.S. [12]

noted that the Multilayer Feedforward Neural Network (MFNN) was used to predict the spectrum state under slowly varying channel conditions; this method had a low complexity and the ability to model nonlinear relationships. In a real WiFi network, the network traffic is improved by 13%-50% with this method. In practical application scenarios, neural network training needs more data samples, which is restricted in some fast mobile applications.

In the HMM method, the spectrum predictor is trained by the historic channel state sequence at first, and then the maximum likelihood probability of the channel state at next time the channel is accessed is calculated with the current channel state [13-16]. Cognitive frequency hopping communication [14], high frequency (HF) cognitive communication [15], and time division multiple access (TDMA) communication [16] have all been validated on a WiFi network [13]. The researchers who developed these methods showed that the probability of a false alarm was increased with the length of the time slot. Good prediction results were obtained in good channel conditions while large prediction errors were produced at low SNRs. The shortcoming of HMM is that it needs to be trained repeatedly.

In this paper, we propose a deep recurrent neural network (DRNN) to predict the multiple time slot spectrum. The existing methods only predict the spectrum of one time slot. If a multiple time slot spectrum can be predicted, the SU can choose the idle channel and combine the spectrum resources from multiple channels, which increases the flexibility of the spectrum utilization. In spectrum sensing, the spectrum is divided into channelized, multiple time slots formed as shown in Fig. 1. The results of spectrum sensing form a time series, and the prediction of a multiple time slot spectrum appears as a multi-step prediction of a time series. It assumes that the time length of each time slot is T_s and the maximum number of time slots is N_{\max} . We used the extended Kalman filter to overcome the slow convergence and vanishing gradient of the gradient descent method when the DRNN was trained.

The rest of this paper is organized as follows. In Section 2, we give a description of the problem. Section 3 introduces some related works. Section 4 describes deep recurrent neural networks, the prediction strategy of multiple time slots with the DRNN, and its training method based on the extended Kalman filter. Our experiments and simulation results are given in Section 5, and we draw conclusions in Section 6.

2. Problem description

The recurrent neural network (RNN) is suitable for processing a time series, so the frequency spectrum needs to be channelized and slotted. When a frequency spectrum is channelized, the bandwidth of a channel is fixed to B_s . For a particular channel, spectrum sensing is carried out in the unit form of the time slot, where the time length of each time slot is T_s . The signal strength of a channel at different time slots is expressed as shown in Fig. 2 (a), where $\mu(n)$ indicates the intensity of the signal in n th time slot. By comparing the intensity of the sensing signal with the threshold of λ , we can determine whether the current sensing channel is used by the PU, and the decision rule is

$$\gamma(n) = \begin{cases} 0 & \mu(n) < \lambda \\ 1 & \mu(n) \geq \lambda \end{cases} \quad (1)$$

After the determination, a time series (such as that shown in Fig. 2(b)) formed by $\{0, 1\}$ is produced, where "1" represents the channel that is occupied by the PU, and "0" represents that the channel is idle. For example, in Fig. 2(b), the channel state is represented by

$\gamma(n) = \{0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, \dots\}$. Assuming that the previous N values in Fig. 2(a) are known, then the prediction gives the subsequent K values, $\{\gamma(N+1), \gamma(N+2), \dots, \gamma(N+K)\}$, which is a multi-step prediction for a time series.

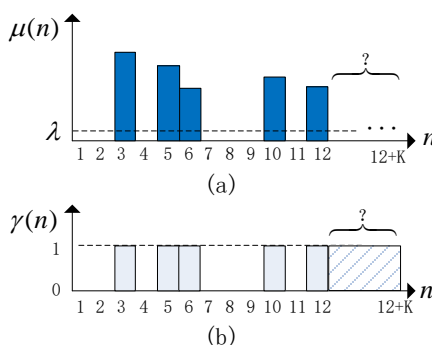


Fig. 2. The spectrum sensing results of a channel expressed in the form of a time series: (a) The signal strength of a channel at different time slots and (b) the channel state represent by $\gamma(n)$.

3. Related works

The multi-step prediction is more difficult than the spectrum prediction for a single time slot and a single-step time series prediction, [17]. First, we needed to build a regression model of the time series that was established with the HMM, NN, autoregressive integrated moving average model (ARIMA), supporting vector machine, nearest neighbor, lazy learning algorithm, and other regression methods [18]. In a single-step prediction, the following projection is implemented:

$$\mu(n+1) = f[\mu(n), \mu(n-1), \dots, \mu(n-p)] + w \quad (2)$$

where f is the regression model and w is the prediction error.

Either NN or HMM is generally used for the one slot spectrum prediction. Compared to HMM, NN has a more powerful representation ability and the parameters of the prediction model only need to be trained once, while the HMM needs repeated training. Any prior statistical distribution knowledge of the PU's channel state, which is one of the difficulties in the cognitive radio network, is not required when the NN is operating as a spectrum predictor. Once NN training is complete, its computing complexity is greatly reduced. The NNs currently used for spectrum prediction include the MLP [10,19] and MFNN [12,20]. In the NN, neurons in different layers are connected by adaptive weights, and these weights need to be trained in the initial stage. When training, the spectrum sensing samples are constructed as vector input into the NN, and the weight values are obtained through the back propagation (BP) algorithm, which uses the channel state of the next time slot as the expected value [21]. In practical applications, this kind of NN lacks the dynamic time characteristics similar to the HMM, and it is difficult to adapt to the change of the channel. However, the predictor based on the RNN has a similar structure to that found in the HMM.

The RNN is a transforming form of the deep neural network (DNN). It inherits the model structure of the DNN's linear transformation and nonlinear activation function and absorbs the idea of the finite series correlation of the HMM model. As shown in Fig. 3(a), the hidden layer has an edge connected with itself, which is the key to realizing the recurrent property. In Fig. 3,

t represents the time of the series. The self-connecting edge of the hidden layer is actually connected to h at the last time t . At each t , h_t is the function of the current input x_t and h_t of the last time slot:

$$h_t = F_\theta(h_{t-1}, x_t) \tag{3}$$

When the self-connection of the hidden layer is unfolded, an architecture similar to that of the HMM is obtained, as shown in Fig. 3(b). The main difference between the two is that the parameters of the RNN are shared across time. That is, for any t , the network parameters from h_{t-1} to h_t and x_t to h_t are the same, which greatly reduces the complexity of the model. Sharing parameters allows the RNN to be adapted to a series of any length and no special adjustment is required for training and prediction. These are the advantages to using the RNN to predict a time series. However, many time series in practical applications have different time scales, so the RNN is not suitable for the prediction of the time series with such characteristics.

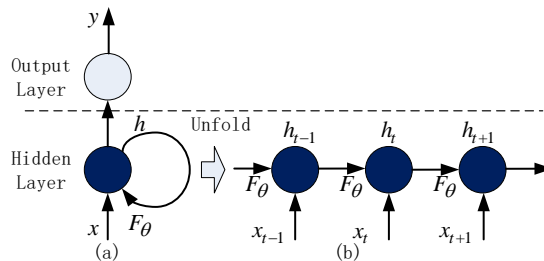


Fig. 3. When the hidden layer is unfolded in time, RNN has a similar structure to that of HMM: (a) the hidden layer has an edge connected with itself and (b) the architecture unfolded in time.

4. Proposed Method

4.1 Deep recurrent neural network (DRNN)

In this paper, we used the DRNN to model the time series of the channel spectrum state. For spectrum sensing, the smaller time slots result in higher time resolutions. Measurements of the PU's activity indicate that the idle or busy duration of the channel is related to the previous state [22, 23]. At different time scales, the spectrum of the PU exhibits different characteristics. Those time series of the different slot lengths have different time scale characteristics. The weakness of the common RNN lies in that the information of the output layer is only through one processing layer, so the input of the current time step lacks hierarchical processing. In addition, the RNN is not able to process a time series on multiple time scales. Hermans and Schrauwen have used the DRNN to provide the time hierarchy information through multiple scales [24]. The differences between the RNN and DRNN are shown in Fig. 4. The RNN has only one hidden layer, while the DRNN has L hidden layers, each layer contains N neurons, and all of the hidden layers are connected in time. According to equation (2), the transfer function of the i th element in the l th hidden layer at t is defined as shown in the following equation:

$$\begin{aligned} h_{i,t}^l &= F_\theta(\mathbf{h}_i^{l-1}, \mathbf{h}_{i-1}^l) \\ &= \Phi_l(\mathbf{U}^l \mathbf{h}_{i-1}^l + \mathbf{W}^l \mathbf{h}_i^{l-1}) \end{aligned} \tag{4}$$

where h_t^l is the hidden state of the l th layer at time t , $\Phi_l(\bullet)$ is a nonlinear function by unit of the l th layer, \mathbf{w}^l is the weight matrix of the l th layer, \mathbf{U}^l is the weight matrix of the recurrent connection of the l th layer, $\mathbf{h}_t^{L-1}=[h_{1,t}^{L-1} h_{2,t}^{L-1} \dots h_{N_{L-1},t}^{L-1}]$, and N_{L-1} is the number of elements in the L th layer. When $l=1$, the hidden activation h_t^1 can be computed by $h_t^0=x_t$. In considering how to choose the activation function in the neural network, \mathbf{w}^l obeyed the empirical rule after initializing; almost half of the neurons using the traditional Sigmoid function were activated simultaneously, which created difficulties in training the deep network. Therefore, we chose to use the rectified linear unit $\Phi_l=\max(0,x)$ [25], which had a better performance than the Sigmoid or tanh function. Fig. 4(b) shows that the network structure inherently has different time scales. The bottom layer of the network has a fading memory for the input, and each additional remaining layer has a fading memory for the underlying hidden state and input from the past.

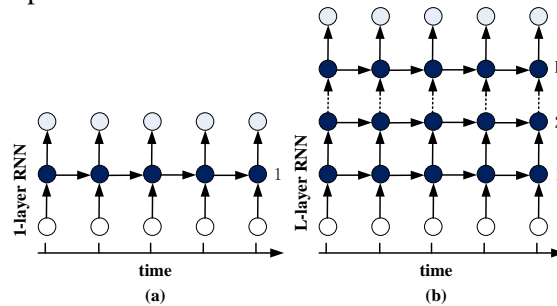


Fig. 4. Structure of the RNN and DRNN: (a)RNN and (b)DRNN.

The computation complexity of the DRNN depends on its structure. Relative to the DNN, the DRNN adds storage and computational power for temporal matrix \mathbf{U}^l . Given the same input during the forward propagation stage, a DRNN with L hidden layers, N hidden units, and a temporal connection at the l th layer requires extra $\theta(N^2)$ buffers to save the floating-point matrix \mathbf{U}^l and extra $\theta(N^2)$ multiply-add operations to compute the hidden activations in equation (4). Back-propagation through time (BPTT) is usually used to update the network parameters during the backward propagation stage. Given a time series with T time steps in length, the DRNN with a l th time connection requires an extra $\theta(TN)$ space to keep the activations in memory, as well as extra $\theta(TN^2)$ operations to update the parameters. It should be noted that the gradients of the DRNN parameters are updated in time and space, unlike the DNN. The input cannot have a time series that is too long, since the length affects the efficiency of the calculation.

4.2 Multiple time slot frequency spectrum prediction

There are 5 kinds of multiple time slot prediction strategies: Iterative, Direct, Direct and the Recursive (DirRec), Multi-Input Multi-Output (MIMO), and Direct and MIMO (DirMO) [26]. Among them, the DirRec strategy integrates the principle and structure of the Direct [27] and Iterative [28] strategies, and the input of each prediction model includes the current state and prediction state. In the multiple time slot prediction, the state of the k -th prediction needs to be estimated after obtaining the $(k-1)$ -th prediction. If we assume that the time for each prediction is T_p , then it is necessary to wait for time KT_p in order to predict K time slots. The time delay of the prediction reduces the time available to the spectrum in order to measure the sensitive radio frequency spectrum utilization. Every time slot prediction needs a module, which

increases the complexity of the system training.

Using the above analysis, we adopted the MIMO strategy based on the structure of the DRNN. It is first necessary to establish a prediction model F which is input to the channel state of the known I time slots and outputs the channel state of the predicted J time slots for the future. Here, we use DRNN to achieve the prediction model. **Fig. 5** is the space and time structure of a DRNN with L hidden layers and $N_l(1 \leq l \leq L)$ units at each layer. The visible layer of the DRNN has I units to receive the known spectrum sending samples, and the number of units in the output layer corresponding to the number of time slots for the output is J . Assuming the network input is $\mathbf{x}_t = [x_{t,1} \ x_{t,2} \ \dots \ x_{t,P}]$, the number of the known channel state slots is P . Equation (4) then gives the output of the i -th unit in the input layer at time t as

$$h_{i,t}^1 = \Phi_1(\mathbf{W}_i^1 \mathbf{x}_t) \tag{5}$$

The result generated by the output layer is the prediction of the next time channel state. The output of the j th unit(s) at time t is

$$\hat{y}_{j,t} = \Phi_o(\mathbf{W}_j^{L+1} \mathbf{h}_t^L) \tag{6}$$

where $\mathbf{h}_t^L = [h_{1,t}^L \ h_{2,t}^L \ \dots \ h_{N_L,t}^L]$ is the output of the L th hidden layer, which can be calculated by equation (4).

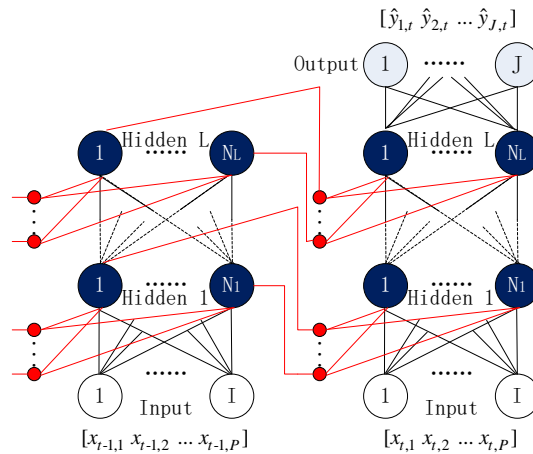


Fig. 5. The space and time structure of a DRNN with L hidden layers and $N_l(1 \leq l \leq L)$ units at each layer, which has I units in the input layer and J units in the output layer.

Fig. 5 shows that the J timeslot channel states predicted in the future are related to the parameters of the network and the internal state at time t . Due to the memory characteristics of the structure, the internal state at time t is dependent on the weight matrix and the internal state of the past time slots. Once the weight is updated at time t , the current J outputs can be calculated. If the relationship between the input and output of the network is constant, the weight matrix at each time is the same. The weight is determined according to the prediction error. The error at time t is partly related to the error of the weight and partly related to the state estimation error at time t , which is also related to the prediction error at time $t-1$. Therefore, the error is actually affected by the weight matrix. Therefore, the prediction model F first need

to use a certain number of samples for training. The method for reducing the errors is to improve the training quality of the DRNN for the weight matrix with \mathbf{w}^l and \mathbf{u}^l .

4.3 Training through space and time based on the extended Kalman filter (EKF)

Back propagation (BP) is a classical method of neural network training. The BP for the DRNN training is not only through space, but also time. Usually, the gradient descent method is used for back-propagation training, but this method suffers from a slow convergence, gradient disappearance, and other issues [26]. The EKF is better able to avoid these problems [27], so we used it for the estimation of the weights in space and time.

The EKF is a state estimation technique for nonlinear systems. Suppose a discrete system with additive input and no observation noise is

$$\begin{cases} \mathbf{x}(n+1) = \mathbf{f}(\mathbf{x}(n+1)) + \mathbf{q}(n) \\ \mathbf{d}(n) = \mathbf{o}_n(\mathbf{x}(n)) \end{cases} \quad (7)$$

where $\mathbf{x}(n)$ is the state vector of the system, \mathbf{f} is a system status update function, $\mathbf{q}(n)$ is the external input to the system, $\mathbf{d}(n)$ is the output of the system, and \mathbf{o}_n is a time-dependent observation function. At time $n=0$, the system state $\mathbf{x}(0)$ is estimated by a multidimensional normal distribution with a mean $\hat{\mathbf{x}}(0)$ and covariance $\mathbf{P}(0)$. The task of the EKF is to give an estimation $\hat{\mathbf{x}}(n+1)$ of the true state $\mathbf{x}(n+1)$, given the initial state and all the previous output values. For this purpose, the following two time update and three measurement update computations were used in this research:

$$\begin{cases} \hat{\mathbf{x}}^*(n) = \mathbf{f}(\hat{\mathbf{x}}(n)) \\ \mathbf{P}^*(n) = \mathbf{F}(n)\mathbf{P}(n-1)\mathbf{F}(n)^t + \mathbf{Q}(n) \end{cases} \quad (8)$$

$$\begin{cases} \mathbf{K}(n) = \mathbf{P}^*(n)\mathbf{O}(n)[\mathbf{O}(n)^t\mathbf{P}^*(n)\mathbf{O}(n)]^{-1} \\ \hat{\mathbf{x}}(n+1) = \hat{\mathbf{x}}^*(n) + \mathbf{K}(n)\xi(n) \\ \mathbf{P}(n+1) = \mathbf{P}^*(n) - \mathbf{K}(n)\mathbf{O}(n)^t\mathbf{P}^*(n) \end{cases} \quad (9)$$

where $\mathbf{F}(n)$ and $\mathbf{O}(n)$ are Jacobians of \mathbf{f} and \mathbf{o}_n with respect to the state variables,

$$\begin{cases} \mathbf{F}(n) = \left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(n)} \\ \mathbf{O}(n) = \left. \frac{\partial \mathbf{o}_n(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(n)} \end{cases} \quad (10)$$

$\xi(n) = \mathbf{d}(n) - \mathbf{o}_n(\hat{\mathbf{x}}(n))$ is the difference between the observed output and the estimated output; $\mathbf{P}(n)$ is an estimate of the conditional error covariance matrix $E[\xi | \mathbf{d}(0), \dots, \mathbf{d}(n)]$; $\mathbf{Q}(n)$ is the diagonal covariance matrix of the process noise, and the time updates $\hat{\mathbf{x}}^*(n)$, $\mathbf{P}^*(n)$ of the state estimate and state error covariance estimate are obtained from extrapolating the previous estimates with the known dynamics \mathbf{f} . The basic idea of the EKF is to update $\hat{\mathbf{x}}(n)$, $\mathbf{P}(n)$ based on equation (8) to generate a preliminary guess for $\hat{\mathbf{x}}^*(n)$, $\mathbf{P}^*(n)$. Then, the information contained in $\mathbf{d}(n)$, which enters the measurement update in the form of $\xi(n)$ and is

accumulated in the Kalman gain $\mathbf{K}(n)$, is integrated to adjust these preliminary estimates. The initialization of the system demands $\mathbf{P}(0)=\delta\mathbf{I}$, $\mathbf{Q}(0)=\lambda\mathbf{I}$, where \mathbf{I} is the unit matrix.

Spectrum predictions were based on the DRNN's estimated future J time-slot channel states $\hat{\mathbf{y}}_t$ through the past P time-slot channel states \mathbf{x}_t . In order to obtain accurate prediction results, it was necessary to have a good estimate of the weight matrix of \mathbf{U} and \mathbf{W} , which are all represented as \mathbf{V} in the following equations. In order to use the EKF for the DRNN to estimate the optimal weights, \mathbf{V} was regarded as the state of a dynamic system. Then, the output $\hat{\mathbf{y}}_t$ of the DRNN can be seen as a function of \mathbf{V} and \mathbf{x}_t :

$$\hat{\mathbf{y}}_t = \mathbf{o}(\mathbf{v}, \mathbf{x}_t) \quad (11)$$

which assumes that the transient effects of the initial state of the network are eliminated. The inputs can be integrated into the output function \mathbf{o} , rendering it the time-dependent function \mathbf{o}_n . Further, we assumed that the network update contained process noises, which were added to the weight in the form of Gauss uncorrelated noise $\mathbf{q}(t)$. The dynamic equations of (7) for the perfect DRNN are

$$\begin{cases} \mathbf{v}(t+1) = \mathbf{v}(t) + \mathbf{q}(t) \\ \hat{\mathbf{y}}_t = \mathbf{o}_n(\mathbf{v}(t)) \end{cases} \quad (12)$$

The training of the DRNN was carried out from the initial guess $\hat{\mathbf{v}}(0)$ to estimate the steady state $\mathbf{v}(t)$ and the output $\hat{\mathbf{y}}_t$. The error covariance matrix was initialized to the large diagonal matrix $\mathbf{P}(0)$ and the diagonal elements were set to large values.

5. Experiment and discussion

5.1 Experiment of the spectrum prediction for the WiFi channel

We carried out the experiment for the WiFi channel in order to validate the spectrum prediction based on the DRNN. Because a sufficient quantity of spectrum sending samples are required for training the DRNN, we used a software defined radio (SDR) system to sense the signal intensity and record it.



Fig. 6. SDR hardware system composed of a notebook computer and the HackRF One.

The SDR system was made up of hardware and software. The hardware included the HackRF One, an open source hardware + software project and a notebook computer. The

HackRF One's main purpose was to provide a cheap SDR program. HackRF One can run on Mac OS X, Linux, Windows, and other platforms. It was connected to the PC through the USB interface. Through the GNU Radio framework or other SDR software, the HackRF One can be configured to different modes of the transmitter or receiver with an operating frequency range of 30 MHz~6 GHz, with a maximum bandwidth of 20 MHz. The notebook and HackRF One are shown in Fig. 6.

The software used was the spectrum sensing and recording system built by GNU Radio [1]. We designed a down converter, filter, and other modules using the GNU Radio framework. These modules were connected to the system as shown in Fig. 7. The modules controlled the central working frequency and the intermediate frequency (IF) of the HackRF One. These samples were stored on the hard disk after the IF signal was filtered and sampled. Finally, we calculated the energy strength of the sampled signal and obtained the channel state.

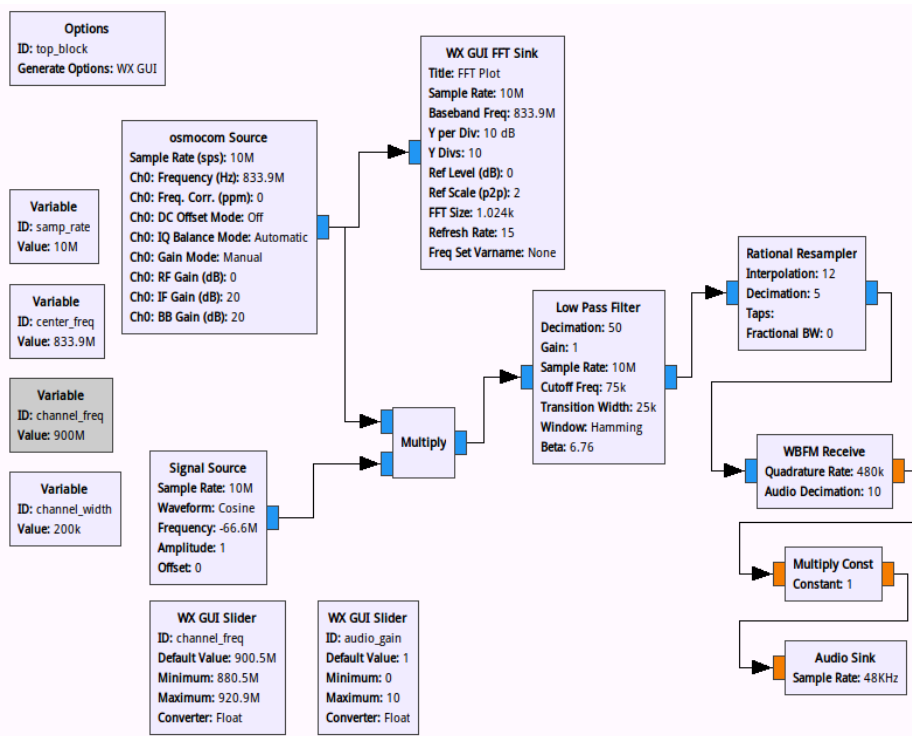


Fig. 7. Spectrum sensing and recording system designed within the framework of Radio GNU.

There are 13 channels available in the 2.412-2.472 GHz range for WiFi. We recorded the WiFi signal between 8 a.m. and 12 a.m., and estimated the state of a single channel. The training for our spectrum predictor used 1,000 time-slot samples, and the rest of the samples were used to validate the prediction results. The comparison between the predicted results and the actual channel state was based on the normalized mean square error (NMSE):

$$\begin{cases} MSE(x_{t,i}, \hat{y}_{t,i}) = \frac{1}{N} \sum_i (x_{t,i} - \hat{y}_{t,i})^2 \\ NMSE(x_{t,i}, \hat{y}_{t,i}) = MSE(x_{t,i}, \hat{y}_{t,i}) / MSE(x_{t,i}, 0) = \frac{\|x_{t,i}, \hat{y}_{t,i}\|_2^2}{\|x_{t,i}\|_2^2} \end{cases} \quad (13)$$

The WiFi spectrum is shown in Fig. 8, and it contains the state of all the channels. Fig. 8 shows that the 1st, 6th, and 11st channels are very crowded, and there are many idle time slots for the 2nd, 3rd, 4th, 5th, 7th, 12th, and 14th channels. We let $\delta=10^{-3}$ and $\lambda=10^{-8}$, and Fig. 9 gives the relationship between the NMSE and training times in the training process. Given the different numbers of the hidden layers and prediction time-slot length separately, the NMSE was obtained; these values are shown in Fig. 10.

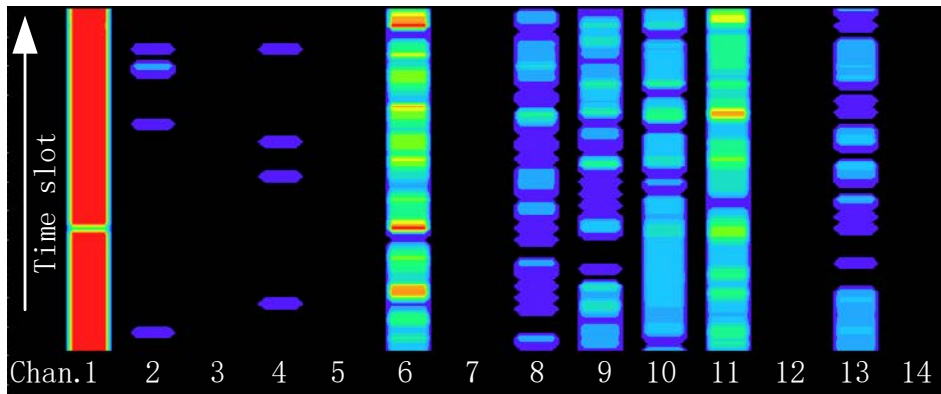


Fig. 8. WiFi spectrum in a certain period of time obtained by our spectrum sensing and recording system.

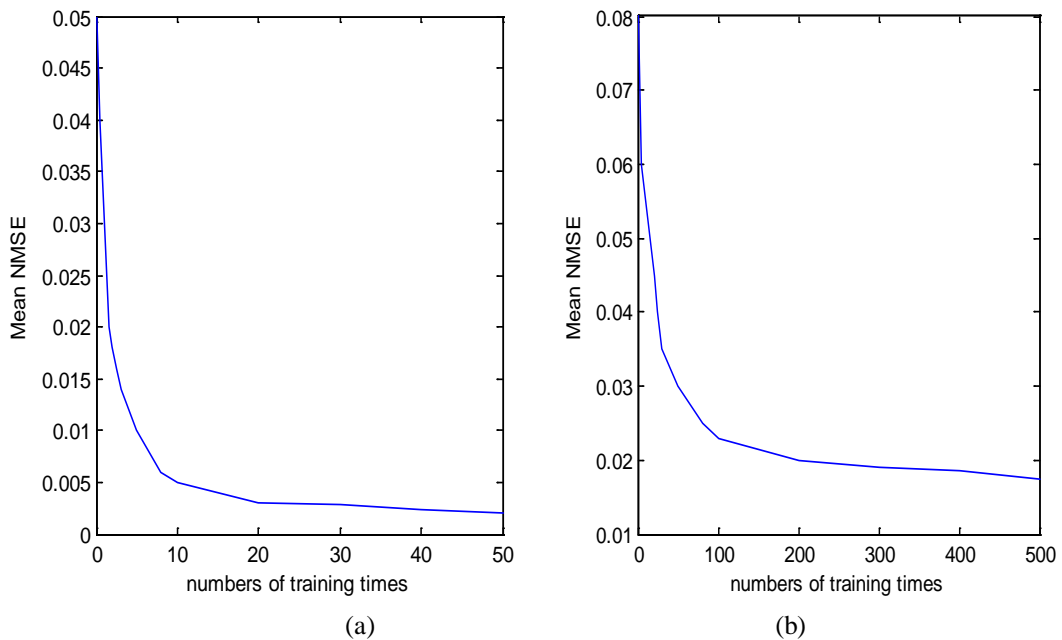


Fig. 9. The relationship between the NMSE and numbers of training times during the process of training the spectrum predictor with the EKF and random gradient descent method, the (a) Extended Kalman Filter and (b) Random Gradient Descent.

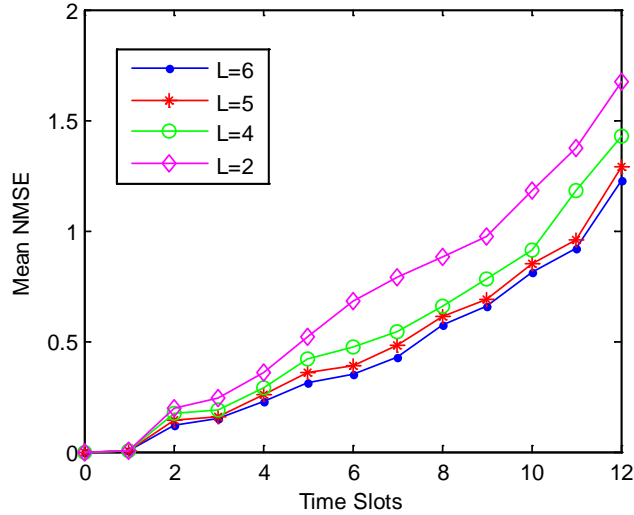


Fig. 10. The variation in the NMSE with different numbers of hidden layers and the prediction of the time-slot length.

5.2 Effects of the spectrum prediction on network performance

The effects of the spectrum prediction on the network performance were verified by simulation. In the simulation of a cognitive radio network, the spectrum occupied by the user's traffic was associated with the assumption that the PU's traffic follows the Poisson process. The simulation scenario was a wireless network with 20 nodes and M_c available channels. The traffic intensity of the network ε was defined as the ratio of t_{traffic} , the average channel holding time, and t_{inter} , the average inter-arrival time of the traffic packet:

$$\varepsilon = \frac{t_{\text{traffic}}}{t_{\text{inter}}} \tag{14}$$

By changing ε , different traffic conditions were obtained. Each channel was divided into slots, and the slot length of all the channels was fixed. We assumed that each SU could only sense one channel at a slot, and stored the spectrum sensing samples for a period of time.

In order to compare the performance of the DRNN's spectrum sensing and spectrum prediction, we used three types of nodes in the simulation; we assumed that these nodes had the same spectrum sensing ability and precision. The first node, SU_{sense} , randomly selected idle channels through spectrum sensing. The second node, $SU_{\text{predictor-1}}$, predicted the next time slot spectrum with the HMM method and randomly selected an idle channel. The third node, $SU_{\text{predictor-2}}$, predicted the multiple time slot spectrum with the DRNN and selected the channel with the largest number of idle slots. We defined an index for evaluating the channel utilization as

$$\eta = \frac{N_S}{N_T} \tag{15}$$

where N_s is the number of idle channels sensed by the nodes and N_t is the total number of idle channels. The performance improvement of $SU_{\text{predictor-1}}$ and $SU_{\text{predictor-2}}$ relative to SU_{sense} is expressed as

$$\begin{aligned}\eta_L &= \frac{\eta_{\text{predictor-x}} - \eta_{\text{sense}}}{\eta_{\text{sense}}} \times 100\% \\ &= \frac{(N_s)_{\text{predictor-x}} - (N_s)_{\text{sense}}}{(N_s)_{\text{sense}}} \times 100\%\end{aligned}\quad (16)$$

where η_{sense} belongs to SU_{sense} and $\eta_{\text{predictor-x}}$ belongs to $SU_{\text{predictor-1}}$ or $SU_{\text{predictor-2}}$. Under the conditions of a certain ε , the performance of the spectrum utilization of both the spectrum prediction and the pure spectrum sensing were found by changing the number of channels M_c for different scenarios, and these results are shown in **Table 1**.

Table 1. The performance of the spectrum utilization of the spectrum prediction and spectrum sensing.

M_c	$(N_s)_{\text{sense}}$	$(N_s)_{\text{predictor-1}}$	$(N_s)_{\text{predictor-2}}$	$(\eta_L)_{\text{predictor-1}}(\%)$	$(\eta_L)_{\text{predictor-2}}(\%)$
3	15825	24524	28425	54.97	79.62
4	15763	24925	29154	58.12	84.95
5	16254	25472	31023	56.71	90.86
6	16342	25869	31872	58.29	95.03
7	15836	26331	32531	66.27	105.42
8	16173	26792	33324	65.65	106.05
9	16481	27538	34056	67.09	106.63
10	16533	28137	34873	70.19	110.93

The other simulation task was to evaluate the savings in the energy consumption using the spectrum prediction. In a single channel scenario, SU_{sense} sensed the spectrum at all the time slots, while $SU_{\text{predictor-1}}$ or $SU_{\text{predictor-2}}$ sensed the spectrum only at the idle time slot. Similarly, we assumed that these nodes had the same spectrum sensing capability and accuracy. Assuming that the energy consumed in a time slot is Δ , the total energy required for SU_{sense} is

$$E_{\text{sense}} = \alpha \Delta \quad (17)$$

where α is the number of time slots for spectrum sensing. The total energy required by $SU_{\text{predictor-1}}$ or $SU_{\text{predictor-2}}$ is

$$E_{\text{predictor-x}} = E_{\text{sense}} - \beta \Delta \quad (18)$$

where β is the number of idle slots predicted by $SU_{\text{predictor-1}}$ or $SU_{\text{predictor-2}}$. Therefore, according to equations (17) and (18), the ratio of the reduced energy required for spectrum sensing is

$$\begin{aligned}\eta_E &= \frac{E_{\text{sense}} - E_{\text{predictor-x}}}{E_{\text{sense}}} \times 100\% \\ &= \frac{\beta}{\alpha} \times 100\%\end{aligned}\quad (19)$$

The simulation results are shown in **Table 2**.

Table 2. The ratio of the reduced energy required for spectrum sensing.

ε	t_{inter}	$\beta_{\text{predictor-1}}$	$\beta_{\text{predictor-2}}$	$(\eta_E)_{\text{predictor-1}}(\%)$	$(\eta_E)_{\text{predictor-2}}(\%)$
0.5	6	15524	17539	44.35	50.25
0.8	6	24537	28923	70.10	82.63
0.5	8	15382	17364	43.95	49.61
0.8	8	23349	28657	66.71	81.88
0.5	10	15396	17278	43.99	49.37
0.8	10	23453	28865	67.01	82.47
0.5	12	15334	17296	43.81	49.41
0.8	12	23388	28792	66.82	82.26

5.3 Discussion

The experiments in section 5.1 showed that the convergence speed of the training spectrum predictor with the EKF is faster than that with the stochastic gradient descent method, and the time consumed by the former is about 1/10 of the latter. After the training was completed, the NMSE with the EKF was smaller than that with the stochastic gradient descent method. This means that in the application of cognitive radio, when the electromagnetic environment changes, the training method using the EKF can quickly adapt to the new environment and achieve a better prediction accuracy. Increasing the number of hidden layers of the DRNN reduces the prediction error. However, the increase of the number of hidden layers increases the complexity, and the training time is longer. In addition, the error increases gradually with the increase of the predicted number of time slots.

The simulation in section 5.2 shows that the improvement of the spectrum utilization and the reduction of the energy consumption in cognitive radio communication can be obtained by using the spectrum prediction method. When compared to the indexes of the channel utilization in Table 1, it shows that the spectrum prediction can improve the channel utilization. The increase in the number of channels improves the spectrum utilization, and the spectrum prediction of the multiple time slot spectrum is higher than that found with one time slot. The energy consumption of the cognitive radio system is reduced because the spectrum prediction avoids the sensing spectrum when the channel is occupied, which is verified by the data in **Table 2**. The energy consumption of $SU_{\text{predictor-1}}$ and $SU_{\text{predictor-2}}$ varied with different traffic intensities, and the energy saving ratio is higher with the increase in the traffic.

6. Conclusion

In this paper, we investigated the structure of the spectrum predictor based on the DRNN, and used it to predict the channel state. We adopted the MIMO strategy to realize the multiple time slot spectrum prediction, which is different from the previous prediction methods. In order to facilitate the multiple slot spectrum prediction, the continuous channel state was divided into multiple time slots, and only the maximum value of the signal energy in each time slot was selected to form the time series of the channel state. We adopted the EKF to improve the training speed of the spectrum predictor. The spectrum sensing and recording system was set up to acquire the historical spectrum sending samples. We carried out the training and spectrum prediction experiments with these samples, and the results show that the EKF has a shorter training time and smaller MNSE than the stochastic gradient descent method. We

analyzed the effects of the multiple slot spectrum prediction on the performance of cognitive radio networks. The simulation results show that the cognitive radio system with our method increases the channel utilization by 79%~110%, and reduces the energy consumption by 50%~80%.

References

- [1] J. Mitola III, "Cognitive radio for flexible mobile multimedia communications," *Journal Mobile Networks and Applications*, vol.6, no.5, pp. 435-441, 2001. [Article \(CrossRef Link\)](#)
- [2] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE JSAC*, vol.23, no.2, pp.201-220, 2005. [Article \(CrossRef Link\)](#)
- [3] S. Haykin, *Cognitive Dynamic Systems: Perception-Action Cycle, Radar and Radio*, Cambridge University Press, London, pp.14-15, 2012. [Article \(CrossRef Link\)](#)
- [4] A. Sahai, N. Hoven, R. Tandra, "Some fundamental limits on cognitive radio," in *Proc. of the Allerton Conference on Communication, Control, and Computing, Monticello, UT, USA*, 29 Sep.–1 October, 2004. [Article \(CrossRef Link\)](#)
- [5] M. Ghozzi, F. Marx, M. Dohler, J. Palicot, "Cyclostationarity-based test for detection of vacant frequency bands," in *Proc. of the 2nd International Conference on Cognitive Radio Oriented Wireless Network and Communication*, Mykonos Island, Greek, 8–10 June, 2006. [Article \(CrossRef Link\)](#)
- [6] P.D. Sutton, K.E. Nolan, L.E. Doyle, "Cyclostationary signature in practical cognitive radio applications," *IEEE JSAC*, vol.26, no.1, pp.13–24, 2008. [Article \(CrossRef Link\)](#)
- [7] Z. Chen, N. Guo, Z. Hu, R. Qiu, "Experiment validation of channel state prediction considering delays in practical cognitive radio," *IEEE Transactions on Vehicular Technology*, vol.60, no.4, pp. 1314-1325, 2011. [Article \(CrossRef Link\)](#)
- [8] P. Huang, C.J. Liu, L. Xiao, J. Chen, "Wireless Spectrum Occupancy Prediction Based on Partial Periodic Pattern Mining," *IEEE Transactions on Parallel & Distributed System*, vol.25, no.7, pp.1925-1934, 2014. [Article \(CrossRef Link\)](#)
- [9] F.R. Huang, W. Wang, H.Y. Luo, G.D. Yu, Z.Y. Zhang, "Prediction-Based Spectrum Aggregation with Hardware Limitation in Cognitive Radio Networks," in *Proc. of IEEE 71st Vehicular Technology Conference*, Taipei, Taiwan, 16-19 May, 2010. [Article \(CrossRef Link\)](#)
- [10] V. Tumuluru, P. Wang, D. Niyato, "Channel status prediction for cognitive radio networks," *Wireless Communications & Mobile Computing*, vol.12, no.10, pp. 862–874, 2012. [Article \(CrossRef Link\)](#)
- [11] V. Tumuluru, P. Wang, D. Niyato, "A neural network based spectrum prediction scheme for cognitive radio," in *Proc. of IEEE International Conference on Communications*, Cape Town, South Africa, 23-27 May, 2010. [Article \(CrossRef Link\)](#)
- [12] B. Nicola, R.T. Bheemarjuna, B.S. Manoj, "A Neural Network based Cognitive Controller for Dynamic Channel Selection," in *Proc. of IEEE International Conference on Communications*, Dresden, Germany, 14-18 June, 2009. [Article \(CrossRef Link\)](#)
- [13] Z. Chen, N. Guo, Z. Hu, R. Qiu, "Channel state prediction in cognitive radio, part ii: Single-user prediction," in *Proc. of IEEE Southeastcon*, Nashville, USA 17-20 March, 2011. [Article \(CrossRef Link\)](#)
- [14] S. H. Shon, S. J. Jang, Kim J. M, "HMM-based adaptive frequency hopping cognitive radio system to reduce interference time and to improve throughput," *KSII transaction on internet. and information system*, pp.475-490, 2010. [Article \(CrossRef Link\)](#)

- [15] L. Melián-Gutiérrez, S. Zazo, J.L. Blanco-Murillo, “Efficiency improvement of HF communications using cognitive radio principles,” in *Proc. of Ionospheric Radio Systems and Techniques Conference*, York, UK, 15-17 May 2012. [Article \(CrossRef Link\)](#)
- [16] G.C. Tiao, R. S. Tsay, “Some advances in nonlinear and adaptive modeling in Time Series,” *Journal of Forecasting*, vol.13, no.2, pp. 109-131, 1994. [Article \(CrossRef Link\)](#)
- [17] J. G. De Gooijer, R. J. Hyndman, “25 years of Time Series Forecasting,” *International Journal of Forecasting*, vol.22, no.3, pp.443-473, 2006. [Article \(CrossRef Link\)](#)
- [18] J.W. Wang, R. Adriman, “Analysis of opportunistic spectrum access in cognitive radio networks using hidden Markov model with state prediction,” *EURASIP Journal on Wireless Communications and Networks*, vol.2015, no.10, pp.1-8, 2015. [Article \(CrossRef Link\)](#)
- [19] K.S. Narendra, K. Parthasarathy, “Identification and control of dynamical systems using neural networks,” *IEEE Transaction on Neural Networks*, vol.1, no.1, pp. 4–27, 1990. [Article \(CrossRef Link\)](#)
- [20] D. Svozil, V. Kvasnicka, J. Pospichal, “Introduction to multi-layer feed-forward neural networks,” *Chemometrics and Intelligent Laboratory Systems*, vol.39, no.1, pp.43-63, 1997. [Article \(CrossRef Link\)](#)
- [21] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, New Jersey, USA, pp.161–175, 1999. [Article \(CrossRef Link\)](#)
- [22] M. Wellens, J. Riihijärvi, P. Mähönen, “Empirical time and frequency domain models of spectrum use,” *Physical. Communication*, vol.2, no.1-2, pp.10-32, 2009. [Article \(CrossRef Link\)](#)
- [23] M. Wellens, P. Mähönen, “Lessons learned from an extensive spectrum occupancy measurement campaign and a stochastic duty cycle model,” *Mobile Networks and Applications*, vol.15, no.3, pp.461-474, 2010. [Article \(CrossRef Link\)](#)
- [24] M. Hermans, B. Schrauwen, “Training and analyzing deep recurrent neural networks,” in *Proc. of Advances in Neural Information Processing Systems 26*, Lake Tahoe, pp. 190-198, 5-10 December, 2013. [Article \(CrossRef Link\)](#)
- [25] V. Nair, G.E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” in *Proc. of International Conference on Machine Learning*, Haifa, Israel, 21-24 June 2010. [Article \(CrossRef Link\)](#)
- [26] S. B. Taieb, G. Bontempi, A. F. Atiya, “Sorjamaa, A. A review and comparison of strategies for multi-step ahead time series forecasting based on the {NN5} forecasting competition,” *Expert Syst. Appl.*, vol.39, no. 8, pp. 7067–7083, 2012. [Article \(CrossRef Link\)](#)
- [27] C. Hamzacebi, D. Akay, F. Kutay, “Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting,” *Expert Syst. Appl.*, vol.36, no.2, pp.3839–3844, 2009. [Article \(CrossRef Link\)](#)
- [28] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, A. Lendasse, “Methodology for long-term prediction of time series,” *Neurocomputing*, vol.70, no.16-18, pp.2861–2869, 2007. [Article \(CrossRef Link\)](#)



Zhi-ling Tang received the B.S. and M.S. degree in communications engineering from Guilin University of Electronic Technology, China, in 1997 and 2005, the Ph.D. degrees in information and communications engineering from Xidian University, China, in 2013. His main research interests include HF network, next generation of wireless sensor networks and cognitive radio.



Si-min Li received the BS degree in Wireless Communication Engineering from Nanjing University of Posts and Telecommunications, Jiangsu Province, China, in 1984, and the MS and PhD degrees in Electronics Engineering from University of Electronic Science and Technology of China, Chengdu, Sichuan Province, China, in 1989 and 2007, respectively. His current research interests lie in the design of electrically small antenna, the antenna arrays for HF communication systems and WSN.