Research Article
The Rayleigh Fading Channel Prediction via Deep Learning

Run-Fa Liao,[1] Hong Wen ⬡,[1] Jinsong Wu,[2] Huanhuan Song,[1] Fei Pan,[1] and Lian Dong[1]

[1]*The National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China*
[2]*Department of Electrical Engineering, Universidad de Chile, Santiago 833-0072, Chile*

Correspondence should be addressed to Hong Wen; wcdma_2000@hotmail.com

This paper presents a multi-time channel prediction system based on backpropagation (BP) neural network with multi-hidden layers, which can predict channel information effectively and benefit for massive MIMO performance, power control, and artificial noise physical layer security scheme design. Meanwhile, an early stopping strategy to avoid the overfitting of BP neural network is introduced. By comparing the predicted normalized mean square error (NMSE), the simulation results show that the performances of the proposed scheme are extremely improved. Moreover, a sparse channel sample construction method is proposed, which saves system resources effectively without weakening performances.

1. Introduction

The future wireless communications (5G) put forward the demands of high-speed transmission, quick access, high reliability, and strong security communications [1]. Hence, new technologies should be adopted to meet the high-speed and high-efficiency transmissions and access demands of 5G [2]. Massive MIMO, non-orthogonal multiple access (NOMA), and tight cooperation for wireless senor nodes are expected to become key technologies for the future 5G systems. A major limitation for massive MIMO, NOMA, and coordinated multipoint (CoMP) systems is the channel state information (CSI) knowledge at the transmitter, which can be obtained partly by the channel prediction techniques. Meanwhile, the physical layer security methods utilize channel reciprocity and diversity to accomplish the so-called "encryption" in the physical layer [3, 4]. Compared with the conventional cryptographic technologies, under the same security requirement, the key length in physical layer security is greatly reduced, and even not required, which is especially suitable for the quick access system. Unfortunately, physical layer security transmission is only dependent on physical CSI. For wireless fading channels, the change of channel information is not conducive to the implementation of the physical MIMO, cooperation, and the security. As shown in Figure 1,

the channel information is constantly changing due to the change of the location of the legal receiver, which makes the base station unable to perform robust precoding or beamforming etc. Therefore, the channel prediction is a key point for such problems.

There are considerable research results on the channel parameter prediction. The literatures [5–8] employ the optimal linear algorithm and autoregressive tracing algorithm to predict the flat fading channel, in which channel impulse response prediction is performed by linearly combining the current CSI with the past one. In [5], performance analysis is carried out for long range prediction (LRP) under the actual channel model and the stationary random phase model. Complex-valued neural networks are discussed by T. Ding and A. Hirose to predict time-varying channels and applying them on the hardware [9]. The error rates, compared to the traditional methods, have made improvements. The article [10] utilizes the echo state network to predict channel and proposes a fixed weight method, to reduce computing complexity. Literature [11] proposed a novel support vector machine method to predict a more sophisticated environment. The MUSIC algorithm for channel prediction is investigated in [12]. However, the above-mentioned algorithms either have the lack of high estimation error rate or suffer from high complexity. The major flaws of these methods are
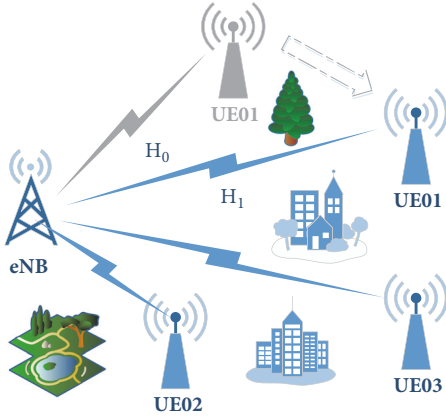
FIGURE 1: Predicting scene.

that all of them only predict the parameters for the next moment without providing the prediction of CSI after the multiple moments.

In this paper, the multi-time channel prediction system is proposed by taking advantage of the BP neural network with the single hidden layer. Hinton puts forward the concept of deep learning in the year of 2006, which is actually the multi-hidden-layer multi-sensor neural network, including BP neural network and convolutional network [13]. Deep learning is often used in computer vision, pattern recognition, and image classification [13–15]. In this paper, we employ deep learning for wireless channel prediction, while the early stopping strategy is adopted to avoid overfitting [16]. In addition, two-sample construction schemes, namely, the sparse sample construction scheme (SSCS) and normal samples construction scheme (NSCS), are proposed, which can reduce the computational cost and guarantee the prediction accuracy.

We adopt LTE standard frame structure, and the length of the frame is set to 10ms. Specifically, each frame is stratified into 10 subframes, and each subframe consists of two time slots. We assume that each time slot uses 1/3 overhead comb-type pilots. The single-time channel prediction system can save 1/10 pilot resources, and it can save 1/60 system resources. The multi-time algorithm we proposed can save 1/2 or 2/3 pilot resources, which can save 1/6 or 2/9 system resources. So the algorithm we proposed in this paper is very meaningful and useful.

This article is structured as follows. Section 2 introduced the Rayleigh fading channel model and BP neural network, which will be the basis of our novel mothed. The multi-time channel prediction system that can predict the channel information at multiple moments is presented in Section 3. Section 4 includes the simulation results and analyses. Conclusions are given in Section 5.

2. Preliminary

The symbols used in this article will be briefly described. Uppercase bold letters are used for the matrix and lowercase bold letters for vectors. The elements are represented by the

letters with subscripts and not bold. The nth vector and the nth samples are presented by the superscripts with round brackets.

2.1. Rayleigh Fading Channel Model. The propagation in any wireless channel is either a line-of-sight (LOS) propagation or a non-line-of-sight (NLOS) propagation. The probability density function (PDF) of a received signal in LOS environment obeys the Rician distribution, while the PDF of the received signal in the NLOS environment obeys the Rayleigh distribution. We can form a Rayleigh channel by scattering components without a direct path, which can be expressed as follows [17–19]:

$$h(t) = \sum_{n=1}^{N} a_n e^{j(2\pi f_n t + \varphi_n)} \tag{1}$$

where N is the number of multipaths and a_n is the amplitude of the nth path. f_n, φ_n represent the Doppler frequency shift and the phase of the nth path, respectively. The Doppler frequency shift is expressed as $f_n = (v/c) f_c \cos \theta_n$, where v is the moving speed of the user. c is the speed of light, f_c is the carrier frequency, and θ_n is the angle between the user's moving direction and the incident radio wave angle.

The sharp Rayleigh fading channels conforming to a given Doppler spectrum are generated by complex sine wave synthesis, just like the Jakes' channel model [20]. The final channel information of the Jakes model is complex-valued, which is given by the following:

$$h(t) = h_I(t) + jh_Q(t) \tag{2}$$

In this paper, the deep learning samples are sampled at different transmission time slots, and the associated complex-valued CSIs of h (t) can be divided into real and imaginary parts. Accordingly, we predict the real value and the imaginary value of channel state information separately. The related processing procedure is given by the following:

$$h(kT_s) = h_I(kT_s) + jh_Q(kT_s) \tag{3}$$

Then, we construct the deep learning samples by capturing channel information $h_I(kT_s)$, $h_Q(kT_s)$. Finally, BP neural network is adopted to predict the channel information at a later time based on learning from the channel information of the past time slots.

2.2. Back Propagation (BP) Neural Network. Hornik proved that the multi-layer feed forward network containing enough neurons in the hidden layer can approach a continuous function of arbitrary complexity and precision [21]. The deep learning technology has been extensively used in computer vision, pattern recognition, and image classification [13–15]. In this paper, the deep learning is exploited to match the fading channel changing trajectory and to achieve channel prediction. Backpropagation (BP) algorithm multi-layer feed forward neural network prediction model is used to predict the fading channel. Figure 2 illustrates a typical multi-input neural network, which includes an input layer, a hidden layer,
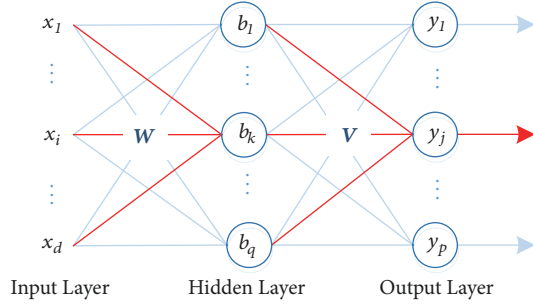
FIGURE 2: Single hidden layer BP neural network.

and an output layer. In the field of machine learning, the neural network as shown in Figure 2 is generally called two-layer neural network (the input layer does not count), or a single hidden layer neural network. We will adopt this statement in this paper.

Given the training sample set,

$$D = \left\{ \left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)} \right), \left(\mathbf{x}^{(2)}, \mathbf{y}^{(2)} \right), \ldots, \left(\mathbf{x}^{(m)}, \mathbf{y}^{(m)} \right) \right\},$$

$$\mathbf{x}^{(k)} \in \mathbb{R}^d, \quad \mathbf{y}^{(k)} \in \mathbb{R}^p \tag{4}$$

where $\mathbf{x}^{(k)}, \mathbf{y}^{(k)}$ denote the kth input sample and the kth output sample, respectively. d and p ar the dimension of the input sample and output sample, respectively. As discussed above, we use the lowercase bold letters with parentheses superscript, for example, $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$ represents the kth input-output sample.

Then we get the input sample matrix,

$$\mathbf{X} = \left[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)} \right], \quad \mathbf{X} \in \mathbb{R}^{d \times m}, \tag{5}$$

and the output sample matrix,

$$\mathbf{Y} = \left[\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(m)} \right], \quad \mathbf{Y} \in \mathbb{R}^{p \times m} \tag{6}$$

The input value of any node in neural network is the previous neuron multiplied by the weight plus the threshold and then activated by the activation function. Without loss of generality, taking the k-th hidden neuron as an example, $b_k = g_1(\sum_{i=1}^{d} x_i W_{ik} + \xi_k)$ [13]. This paper will use the vectorization description to the neural network transmission formula. The neural network forward propagation vectorization is expressed as follows:

$$\mathbf{Z}_1 = \mathbf{W}^T \mathbf{X} + \Xi$$

$$\mathbf{B} = g_1 \left(\mathbf{Z}_1 \right)$$

$$\mathbf{Z}_2 = \mathbf{V}^T \mathbf{B} + \Theta$$

$$\widehat{\mathbf{Y}} = g_2 \left(\mathbf{Z}_2 \right) \tag{7}$$

$$L \left(\widehat{\mathbf{y}}, \mathbf{y} \right) = \frac{1}{2} \sum_{j=1}^{p} \left(\widehat{y}_j - y_j \right)^2$$

$$J \left(\widehat{\mathbf{Y}}, \mathbf{Y} \right) = \frac{1}{m} \sum_{i=1}^{m} L \left(\widehat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)} \right)$$

where $\mathbf{W} \in \mathbb{R}^{d \times q}$ is the weight matrix connected the input layer and the hidden layer, and $\mathbf{V} \in \mathbb{R}^{q \times p}$ is the weight matrix connected the hidden layer and the output layer. $\Xi = \underbrace{[\xi, \xi, \cdots, \xi]}_{m}$ is the hidden layer threshold matrix which consists of the hidden layer threshold vector ξ and $\Theta = \underbrace{[\theta, \theta, \cdots, \theta]}_{m}$ is the output layer threshold matrix which consists of the output layer threshold vector θ. $\mathbf{Z}_1$ is the hidden layer input matrix and $\mathbf{B}$ is the hidden layer output matrix. $\mathbf{Z}_2$ is the input matrix for the output layer. $\widehat{\mathbf{Y}}$ is the output vector of the output layer which is also the final output of the neural network. g_1, g_2 are the hidden layer and output layer activation functions, respectively. Note that g_1 is always the sigmoid activation function and g_2 is the purelin activation function. The functions operating on vector or matrix mean act on each element separately (e.g., $f(\mathbf{x}) = (f(1), f(2))$, $\mathbf{x} = (1\ 2)$). Additionally, its dimension is the same as the original vector or matrix. $L(\widehat{\mathbf{y}}, \mathbf{y})$ is the loss function. We adopt the mean square error (MSE) of the output as the loss function. $J(\widehat{\mathbf{Y}}, \mathbf{Y})$ is the cost function, and it is equal to the average of the loss function with m samples. The neural network iteratively updates the network weight matrix and the threshold vector by minimizing the cost function $J(\widehat{\mathbf{Y}}, \mathbf{Y})$.

We get $\widehat{\mathbf{y}}$ by forward propagation and then update the weight matrix $\mathbf{W}, \mathbf{V}$ and threshold vector ξ, θ by backpropagation with the gradient descent method. For convenience, the partial derivative of the cost function $J(\widehat{\mathbf{Y}}, \mathbf{Y})$ to output $\widehat{\mathbf{Y}}$ is denoted by $d\widehat{\mathbf{Y}}$, which is $d\widehat{\mathbf{Y}} = \partial J(\widehat{\mathbf{Y}}, \mathbf{Y})/\partial \widehat{\mathbf{Y}}$. The vectorization representation of neural network backpropagation iteration formulas is given by the following:

$$d\mathbf{Z}_2 = \frac{\partial J \left(\widehat{\mathbf{Y}}, \mathbf{Y} \right)}{\partial \widehat{\mathbf{Y}}} * g_2{}' \left(\mathbf{Z}_2 \right)$$

$$d\mathbf{V} = \frac{1}{m} \mathbf{B} \left(d\mathbf{Z}_2 \right)^T \tag{8}$$

$$d\theta = \frac{1}{m} d\mathbf{Z}_2 \cdot \mathbf{e}_1$$

$$d\mathbf{Z}_1 = \mathbf{V} \cdot d\mathbf{Z}_2 * g_1{}' \left(\mathbf{Z}_1 \right)$$

$$d\mathbf{W} = \frac{1}{m} \mathbf{X} \left(d\mathbf{Z}_1 \right)^T \tag{9}$$

$$d\xi = \frac{1}{m} d\mathbf{Z}_1 \cdot \mathbf{e}_1$$

where the symbol $*$ denotes the elements in matrix (or vector) A and B multiplied correspondingly. $(\cdot)^T$ means the matrix transpose. $g'(\cdot)$ represents the function derivation, and vector $\mathbf{e}_1$ satisfies equation $\mathbf{e}_1 = \underbrace{(1, 1, \cdots, 1)}_{m}$. The parameter update rule is expressed as $\tau \longleftarrow \tau - \Delta\tau$. For example, the parameter $\mathbf{W}$ is adaptively updated in the form of $\mathbf{W} = \mathbf{W} - \eta d\mathbf{W}$, where η is the learning rate.

As we know, the regularization, dropout, and early stopping strategies are employed to prevent the overfitting of the neural network [21, 22]. This paper adopts the early stopping strategy to avoid the risk of overfitting; we divide the input

and output samples into training set, verification set, and test set. The training set is used to calculate the gradient and update the link weight and threshold. The verification set is used to estimate the error. If the training set error decreases with the validation set error increasing, training process will stop and the related weights and thresholds with the smallest validation set error will be returned.

In summary, combined with early stopping strategy, the gradient descent method is used to continuously update the neural network information, i.e., weight matrix $\mathbf{W}$, $\mathbf{V}$ and threshold vector $\boldsymbol{\xi}$, $\boldsymbol{\theta}$. Once the parameters update is completed, the neural network will be ready to predict the channel state information (CSI).

3. Rayleigh Fading Channel Prediction Method

The multi-time channel prediction system with single hidden layer effectively predict the channel information at multiple moments, and the proposed deep learning prediction system of the multi-layer neural network can cope with a more sophisticated channel information prediction.

3.1. Prediction Scheme

3.1.1. Single-Time Prediction Scheme. Literature [10] proposed the prediction scheme of channel prediction through the echo state network (ESN). In [10], the channel information of the first n moments is regarded as the input samples and the channel information at the $(n + 1)^{th}$ moment as the output samples. And the samples construction scheme of prediction system in this section will follow this program. Let the output layer's dimension be, which is mentioned above, equal to 1. That would be a single-time channel prediction system. Moreover, we add the estimation error at channel state information, which is

$$\widetilde{h}(t) = h(t) + n(t) \qquad (10)$$

where $n(t)$ is the Gaussian white noise, and its variance is σ_n^2. We construct the following neural network training sample [10]:

$$\mathbf{x}^{(k)} = \left[\widetilde{h}(kT_s), \widetilde{h}((k+1)T_s), \ldots, \widetilde{h}((k+d-1)T_s) \right]$$
$$y^{(k)} = \widetilde{h}((k+d)T_s) \qquad (11)$$

For the kth input and output sample set $(\mathbf{x}^{(k)}, y^{(k)})$, the input sample $\mathbf{x}^{(k)}$ represents the channel information samples at the kth time and the next $(d-1)$ times, and the output sample $y^{(k)}$ is the channel information samples at the $(k + n)^{th}$ time. We choose n_T training samples to train the neural network. And the training set matrix is given by

$$\mathbf{X}_{nT} = \left[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(n_T)} \right]$$
$$\mathbf{y}_{nT} = \left[y^{(1)}, y^{(2)}, \cdots, y^{(n_T)} \right] \qquad (12)$$

The test set matrix consisting of the next n_R samples will test the neural network, as shown below:

$$\mathbf{X}_{nR} = \left[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(n_R)} \right]$$
$$\mathbf{y}_{nR} = \left[\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \cdots, \mathbf{y}^{(n_R)} \right] \qquad (13)$$

The training set $(\mathbf{X}_{nT}, \mathbf{y}_{nT})$ is used to train the neural network, while the test set $(\mathbf{X}_{nR}, \mathbf{y}_{nR})$ will be utilized to measure the performance of neural network.

3.1.2. Multi-Time Prediction System. The existing researches of channel prediction [5–12] are either not accurate enough or too complicated to perform in resource-constrained sensor networks. The echo state network (ESN) channel prediction [10] greatly improves the system predictive performance and reduces the system complexity properly. But it only can predict channel information at the next moment. In this section, a multi-time prediction system is exploited to predict channel information at multiple time slots. It achieves a stronger engineering performance. Furthermore, we propose two-sample construction methods, i.e., *sparse sample construction scheme* (SSCS) and *normal samples construction scheme* (NSCS). In the following, we will discuss these two schemes in detail.

(a) Normal Samples Construction Scheme (NSCS). The normal samples construction scheme is a continuous sample construction method. It just adds more outputs on Y. *Zhao's* scheme. The kth input and output samples are expressed as follows:

$$\mathbf{x}^{(k)} = \left[\widetilde{h}(kT_s), \widetilde{h}((k+1)T_s), \ldots, \widetilde{h}((k+d-1)T_s) \right]$$
$$\mathbf{y}^{(k)} = \left[\widetilde{h}((k+d)T_s), \widetilde{h}((k+d+1)T_s), \ldots, \qquad (14) \right.$$
$$\left. \widetilde{h}((k+d+q-1)T_s) \right]$$

The NSCS takes full use of the channel information, but it increases the amount of computation. For example, when $d = 10, q = 10$, we can construct 4990 training samples from 5000 channel information sample values. There is only one channel data difference between two adjacent training samples.

(b) Sparse Samples Construction Scheme (SSCS). In order to reduce the computational complexity, we propose a sparse sample construction scheme. That is, there is no duplicate channel state information for any two input sample information, as follows:

$$\mathbf{x}^{(k)} = \left[\widetilde{h}(k \cdot dT_s), \widetilde{h}((k \cdot d + 1)T_s), \ldots, \right.$$
$$\left. \widetilde{h}((k \cdot d + d - 1)T_s) \right]$$
$$\mathbf{y}^{(k)} = \left[\widetilde{h}(((k+1) \cdot d)T_s), \widetilde{h}(((k+1) \cdot d + 1)T_s), \ldots, \qquad (15) \right.$$
$$\left. \widetilde{h}(((k+1) \cdot d + q - 1)T_s) \right]$$

The SSCS only needs 500 samples to traverse 5000 channel information sample values, if $d = 10$, $q = 10$.

1. The weight matrices $\mathbf{W}$, $\mathbf{v}$ are initialized randomly from 0 to 1. The threshold vectors $\boldsymbol{\xi}$, θ initialized to 0. Set the training goal $\varepsilon_{\text{goal}}$ and learning rate η a reasonable value, respectively;
2. Input the channel information training set $(\mathbf{X}_{nT}, \mathbf{y}_{nT})$.
3. while $J(\hat{\mathbf{y}}_{nT}, \mathbf{y}_{nT}) > \varepsilon_{\text{goal}}$ do:
4. Calculate $\mathbf{Z}_1$, $\mathbf{B}$, $\mathbf{z}_2$, $\hat{\mathbf{y}}_{nT}$, and the $L(\hat{\mathbf{y}}_{nT}, \mathbf{y}_{nT})$, $J(\hat{\mathbf{y}}_{nT}, \mathbf{y}_{nT})$ of the loss function and cost function according to equation (7);
5. According to equation (8), the gradient of the output layer weight matrix $d\mathbf{v}$ and the gradient of the threshold $d\theta$ are calculated respectively;
6. The weight matrix of the hidden layer $d\mathbf{W}$ and the gradient of the threshold vector $d\boldsymbol{\xi}$ are calculated are calculated according to (9);
7. Update the weight matrix of the hidden matrix and the output layer $\mathbf{W}$, $\mathbf{V}$ and the threshold vectors $\boldsymbol{\xi}$, θ;
8. End while
9. Input the channel information test set $(\mathbf{X}_{nR}, \mathbf{y}_{nR})$ and calculate the NMSE according to (16).

ALGORITHM 1

1. The weight matrices $\mathbf{W}$, $\mathbf{V}$ are initialized randomly from 0 to 1. The threshold vectors $\boldsymbol{\xi}$, $\boldsymbol{\theta}$ are initialized to 0. Set the training goal ε_{goal} and the learning rate η to a reasonable value, respectively. The intermediate variable ω is initialized to 1;
2. Input the channel information training set $(\mathbf{X}_{nT}, \mathbf{Y}_{nT})$ and verification set $(\mathbf{X}_{nV}, \mathbf{Y}_{nV})$ to train the neural network.
3. For $J(\hat{\mathbf{Y}}_{nT}, \mathbf{Y}_{nT}) > \varepsilon_{\text{goal}}$:
4. Calculate the hidden layer $\mathbf{Z}_1$, $\mathbf{B}$, output layer data $\mathbf{Z}_2$, $\hat{\mathbf{Y}}$ and cost function $J(\hat{\mathbf{Y}}, \mathbf{Y})$ of training set and verification set according to equation (7), respectively;
5. If $J(\hat{\mathbf{Y}}_{nV}, \mathbf{Y}_{nV}) > \omega$;
6. Quit
7. Else do:
8. $\omega = J(\hat{\mathbf{Y}}_{nV}, \mathbf{Y}_{nV})$;
9. According to (8), calculate the gradient of the output layer weight matrix $d\mathbf{V}$ and threshold vector $d\boldsymbol{\theta}$, respectively;
10. According to (9), calculate the gradient of hidden layer weight matrix $d\mathbf{W}$ and threshold vector $d\boldsymbol{\xi}$, respectively;
11. Update the weight matrix of hidden layer and output layer $\mathbf{W}$, $\mathbf{V}$, and the threshold vector $\boldsymbol{\xi}$, $\boldsymbol{\theta}$;
12. End for
13. Input the channel information test set $(\mathbf{X}_{nR}, \mathbf{Y}_{nR})$, and calculate the NMSE according to (16)

ALGORITHM 2

The simulation and analysis of the two schemes will be carried out, respectively. The prediction performance metric is expressed by the normalized mean squared error (NMSE). The NMSE at the qth time slot is given by the following:

$$NMSE_q = \frac{\sum_{k \in n_R} \left| \hat{y}_q^{(k)} - y_q^{(k)} \right|^2}{\sum_{k \in n_R} \left| y^{(k)} \right|^2} \qquad (16)$$

In addition, we adopt the early stopping strategy for the multi-time channel prediction system to avoid overfitting. More specifically, the early stopping divides the sample set into training set $(\mathbf{X}_{nT}, \mathbf{y}_{nT})$, validation set $(\mathbf{X}_{nV}, \mathbf{y}_{nV})$, and test set $(\mathbf{X}_{nR}, \mathbf{y}_{nR})$. The training set $(\mathbf{X}_{nT}, \mathbf{y}_{nT})$ is used to calculate the gradient and update the link weight matrixes $\mathbf{W}$, $\mathbf{V}$ and threshold $\boldsymbol{\xi}$, $\boldsymbol{\theta}$. The verification set is only used to estimate the cost $J(\hat{\mathbf{Y}}_{nV}, \mathbf{Y}_{nV})$. If the training set error decreases and the validation set error increases, the training will stop and the connection weights $\mathbf{W}_{epoch}$, $\mathbf{V}_{epoch}$ and thresholds $\boldsymbol{\xi}_{epoch}$, $\boldsymbol{\theta}_{epoch}$ will return. The specific algorithm is shown in Algorithm 1 and Algorithm 2.
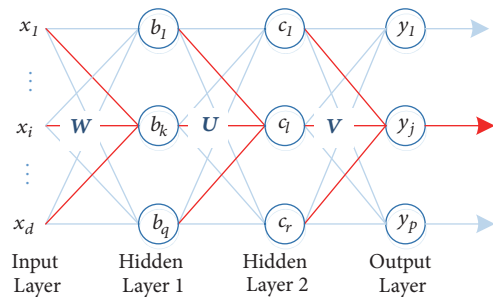

FIGURE 3: Three-layer multi-time prediction system.

3.1.3. Multi-Input and Multi-Output Multi-Layer Neural Network Channel Prediction System. Deep neural network can effectively predict the channel information when channel environment is complicated. More importantly, the deep neural network has better performance than other processing means. Meanwhile, deep neural networks achieve the same performance with fewer neurons. As shown in Figure 3, there
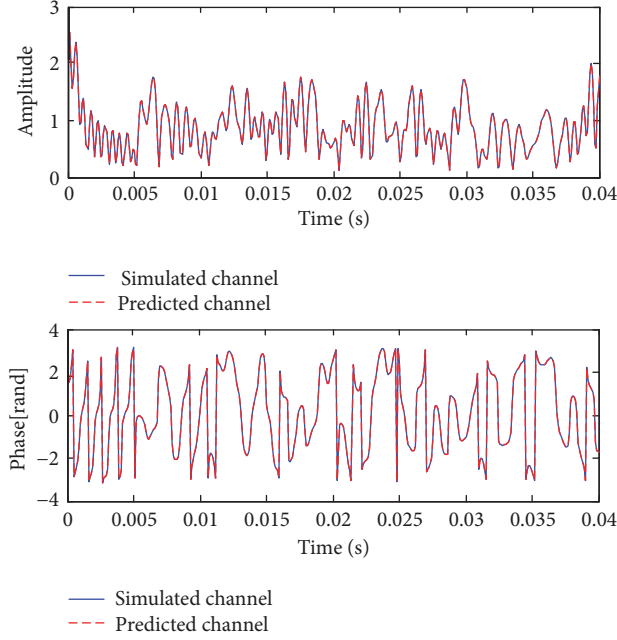
FIGURE 4: Simulated channel and the predicted channel amplitude, phase.

is a three-layer neural network structure with double hidden layer, which is used to predict a more complicated channel.

The parameter update of the three-layer neural network is almost the same as that of the two-layer neural network. Both adopt the backpropagation algorithm, except for the former needing to update three weight matrixes and three threshold vectors in one epoch.

3.2. Complexity Analysis. In this section, we compare the computational complexity of existing algorithms with deep learning methods. The existing methods, such as AR methods [6], DWT-AR-LR methods [9], the ESN method [10], and SVM prediction methods [10], will be mentioned below. The computing complexity of AR method is $O(N_{AR})$, where N_{AR} denotes the order of AR. The complexity of ESN prediction method is $O(\max(M, N_{nz}, N))$, where M is the number of variables, N_{nz} is the number of nonzero elements of middle layer weight matrix, and N is the number of variables in the middle layer. DWT-AR-LR method's complexity is $O(\max(N_{DWT}, N_{AR}, N_{LR}))$ where N_{DWT}, N_{AR}, and N_{LR} represent the number of samples and the order of AR and LR, respectively.

Note that the propagation weight matrix $\mathbf{W}$, $\mathbf{V}$ and threshold vector $\boldsymbol{\xi}$, $\boldsymbol{\theta}$ of the neural network prediction algorithm proposed in this paper are calculated offline. In addition, its overhead is very small. The computational complexity of the mathematical operations of $\mathbf{W}^T\mathbf{x}^{(k)}$, $\mathbf{V}^T\mathbf{b}$ and $g_2(\mathbf{V}^T\mathbf{b} + \boldsymbol{\theta})$ in the neural network are $O(d \times q)$, $O(q \times p)$ and $O(p)$, respectively. Accordingly, the computational complexity of the neural network channel prediction system is $O(\max(d \times q, q \times p))$, where d, q, p are the number of neurons in input layer, hidden layer, and output layer, respectively. In this paper, the number of neurons is very small

(e.g., d=10, q=10, p=10), especially in the multi-layer neural network. And there comes the low complexity.

4. Simulations

4.1. Single-Time Channel Prediction. Firstly, we use the Jakes model to simulate three channel predicted systems [20]. The channel power is fixed to $E_0^2 = 1$. In simulation, we set 34 ($N = 34$) scattering components, 500 ($Ns = 500$) channel information samples, and the sampling interval to be $Ts = 1 \times 10^{-4}s$. The maximum Doppler frequency shift is $f_d = 926Hz$. Phase φ observes uniform distribution, i.e., $\varphi \sim U(-\pi, \pi)$. We obtain 400 neural network samples through 500 channel information samples. Set the training samples $n_T = 200$, test samples $n_R = 200$, learning rate $\eta = 0.001$, and the target error $\varepsilon_{goal} = 1 \times 10^{-4}$. Figure 4 depicts the amplitude and phase of the simulated and predicted channels under the Jakes model. We can see that the channel predicted by the BP neural network is almost identical to the simulated channel of Jakes model.

NMSE is the performance measure; Figure 5 is the comparison of different prediction methods. The x-axis is the signal-to-noise ratios (SNR) of the channel information $h(t)$ and noise $n(t)$, while the y-axis is the NMSE. The red line with triangle is the performance of single-time predict system employing two-layer BP neural network. Rich neuron information gives the BPNN long-term channel memory capability. Thus, it can effectively perform channel prediction. As shown in Figure 5, the NMSE of BP neural network prediction algorithm gradually decreases with the increase of SNR and eventually reaches zero. With a low computational complexity, the accuracy of BPNN method is better than other methods (i.e., SVM, ESN, and DWT-AR-LR).
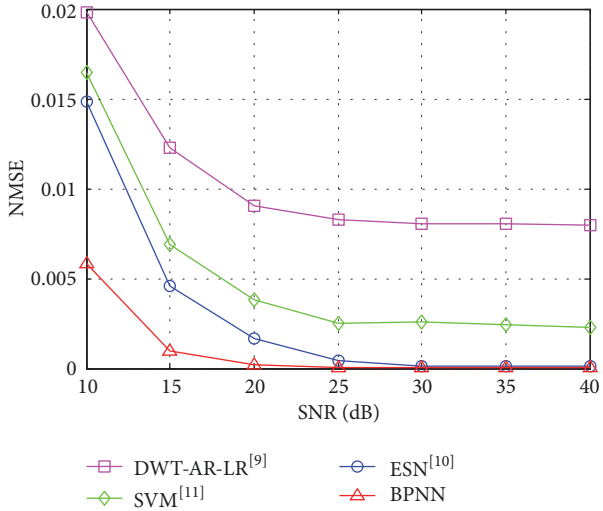
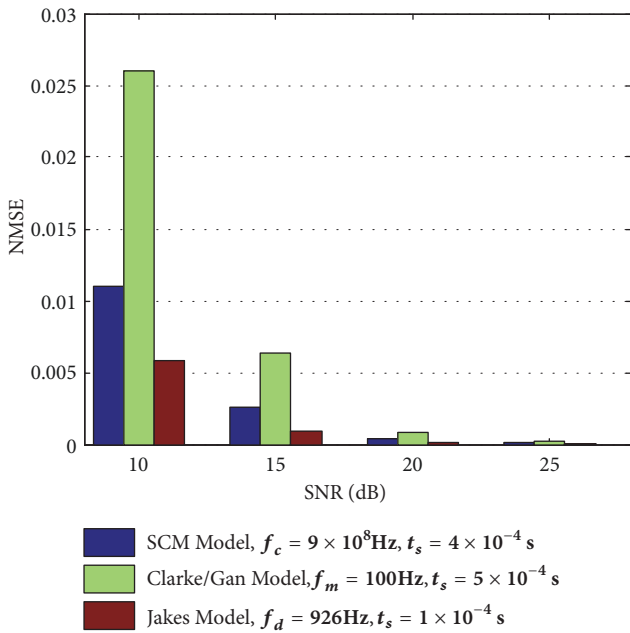FIGURE 5: The prediction accuracy, BPNN versus existing methods.

FIGURE 6: The prediction accuracy under different channel models.

In order to verify the robustness of the algorithm, we present various simulations under different fading Rayleigh channels, for example, the fast fading channel Clarke/Gan's model [23] and the well-known 3GPP Spatial Channel Model (SCM) [24] for MIMO systems.

Figure 6 demonstrates the predicted normalized mean square error (NMSE) under different Rayleigh fading channels. As we know, the autocorrelation of CSI satisfies the zero-order Bessel function over time. Thus, a bad time domain correlation leads to a more difficult channel prediction. For the poor time domain correlation, the CSI of Clarke/Gan's model is sampled in the frequency domain and transformed to the time domain by IFFT, leading to a poor channel prediction performance. Owing to the strong time domain

correlation, Jakes model has the best prediction performance under different values of SNR.

Undeniably, the BP neural network also faces the problem, as other algorithms, which is that the poorer time domain correlation of CSI, the more difficulty to predict future channel. In short, the BPNN algorithm performs better than the other two algorithms, and the prediction performance of the Jakes channel model is the best.

4.2. Multi-Time Channel Prediction System. Similarly, the CSI is generated by the Jakes model for multi-time channel prediction system. The major difference is that we use 5000 channel information samples, i.e., $Ns = 5000$. Other parameters are the same as the single-time channel prediction system.

We research two-sample construction methods (NSCS and SSCS) of multi-time channel prediction system. Under the two strategies, we select 4000 samples and 400 samples, respectively, and 75% of which are training samples, 15% of which are verification samples, and the remaining samples are the test samples. The dimension of input layer sample is = 10. The number of hidden layer neurons is $q = 10$. Output layer neurons are $p = 10$ and = 20, respectively.

4.2.1. Comparison of Two-Sample Construction Schemes with 10-Input and 10-Output (d=10, p=10). For multi-time channel prediction system, as the prediction time increases, the corresponding error increases exponentially. We compare the prediction of two sampling methods, i.e., NSCS and SSCS.

Figure 7 shows that prediction accuracy generally improves as the number of epochs increases. Owing to the early stopping strategy, it is shown that the normal sample construction scheme stops after 224 epochs and the sparse sample prediction scheme stops after 42 epochs. The SSCS scheme has fewer iterations than the NSCS scheme, which can increase the speed of operation and save system resources.

Figure 8 is the NMSE performance of two-sample construction schemes. We can see that the performance of NSCS is better than that of SSCS at the cost of computation complexity. On the other hand, the performance difference between NSCS and SSCS is less than 10^{-4} which can be ignored. Moreover, in order to achieve the same target of NMSE, the latter has less epochs than the former, which is more practical. Notice that an epoch SSCS takes less time than NSCS. To summarize, the NSCS and SSCS we proposed both meet the requirement in [25] of a low estimated error. The SSCS effectively reduces the resource consumption without degrading system performance.

4.2.2. Comparison of Two-Sample Construction Schemes with 10-Input and 20-Output (d = 10, p = 20). Figure 9 shows that the normal samples construction scheme stops after 86 epochs. The sparse sample prediction scheme stops after 61 epochs. Figure 10 is the NMSE of two-sample construction schemes. The multi-time prediction, just like the 10-input and 10-output prediction system, error increases exponentially
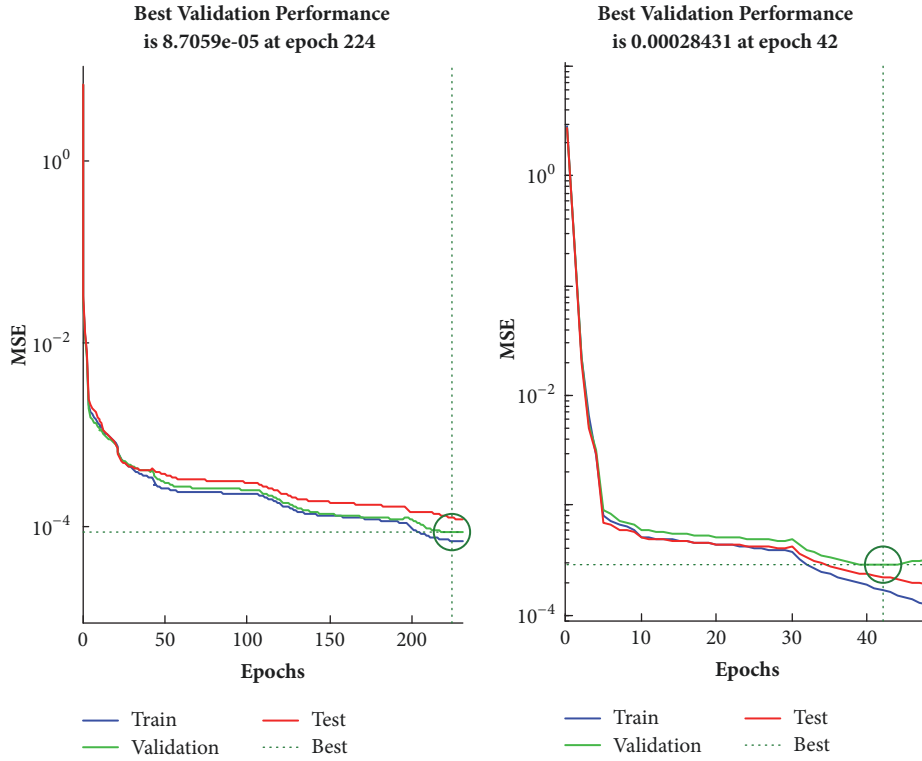
FIGURE 7: The epochs of NSCS and SSCS (d=10, p=10).
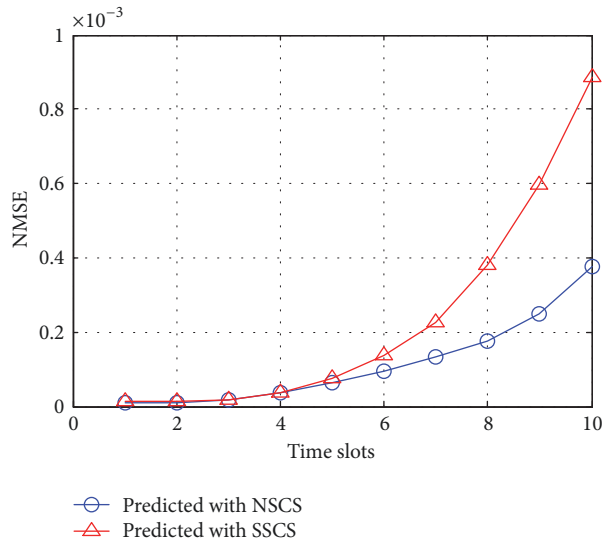
FIGURE 8: The NMSE of NSCS and SSCS (d=10, p=10).

with the different timeslot. The performance of SSCS is slightly worse than that of NSCS.

Nevertheless, the calculation and time costs of SSCS are much smaller than NSCS's.

4.2.3. The Performance of SSCS with Different Power of Noise. Figure 11 demonstrates the predicted NMSE at different SNR in the multi-time prediction system. With the weakening

of the time domain correlations, the prediction errors of different time slot increase exponentially.

4.3. Multi-Hidden Multi-Moment Prediction System. Figure 12 compares the prediction performance of three-layer neural networks and two-layer neural networks. It reveals that the three-layer neural network outperforms the two-layer neural
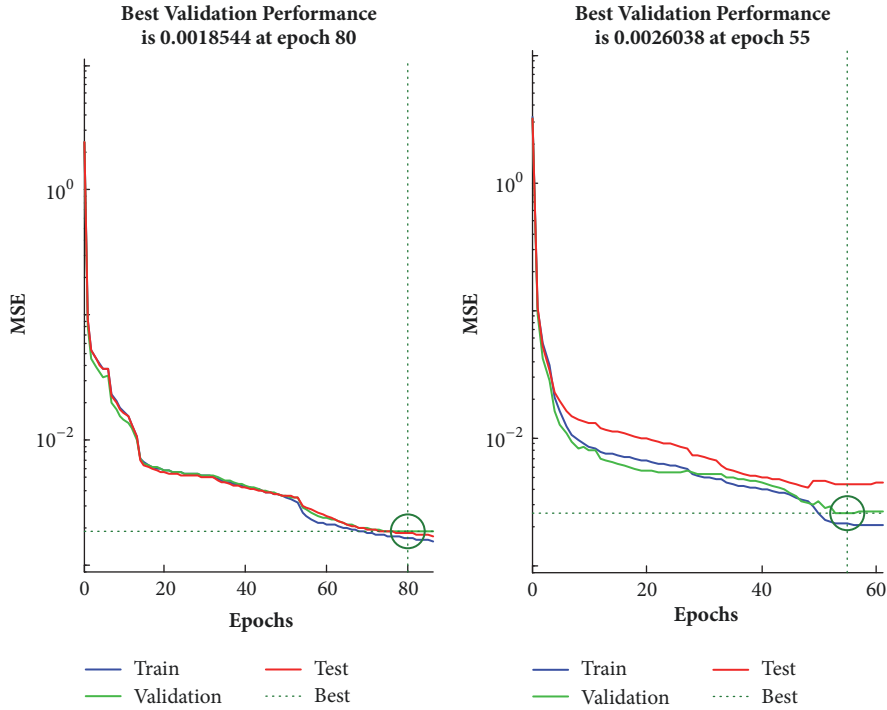
FIGURE 9: The epochs of NSCS and SSCS (d=10, p=20).

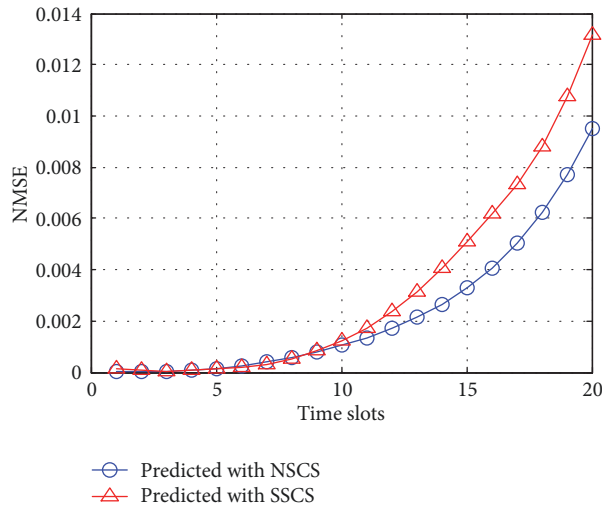FIGURE 10: The NMSE of NSCS and SSCS (d=10, p=20).

network. However, its effectiveness is not obvious since the channel information is not very complicated.

5. Conclusions

The BP neural network with multi-hidden layer is introduced into the channel prediction application. A novel multiple moment CSI prediction scheme is proposed for improving the performance of the massive MIMO, NOMA, CoMP, and physical layer security schemes. The proposed prediction scheme can perform effectively with a short pilot overhead, which is suitable for resource-constrained communication scenes. Meanwhile, we proposed two significant sample construction methods, which extremely improves the prediction performance and reduces the computing complexity. Wide experiences verified the effectiveness of our proposed scheme

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.
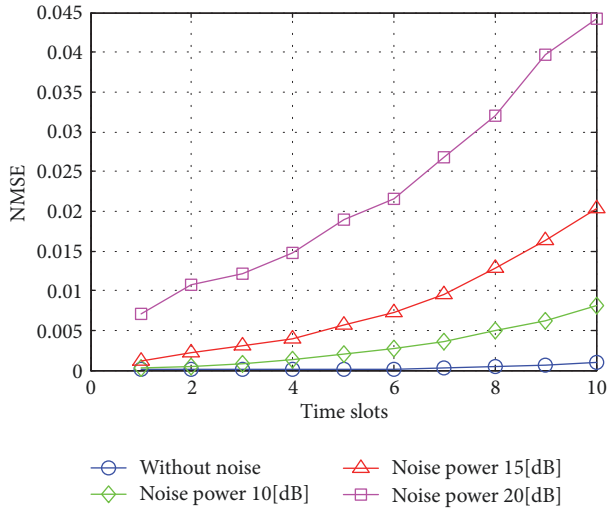
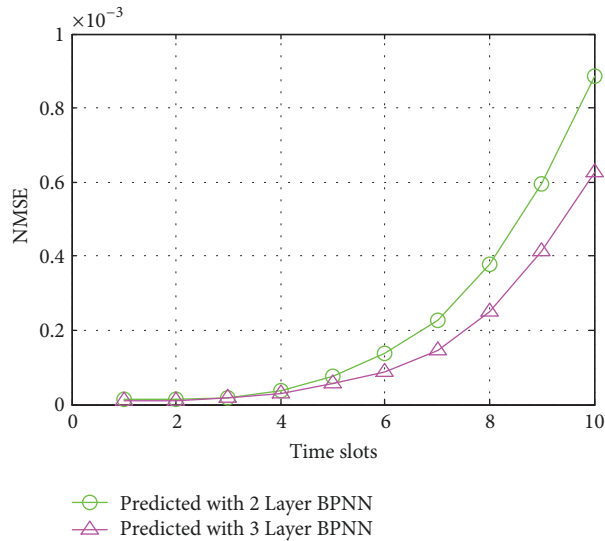Figure 11: The multi-time prediction with noise.

Figure 12: Two-layer neural network and three-layer neural network performance comparison.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

References

[1] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.

[2] W. Cheng, X. Zhang, and H. Zhang, "Statistical-QoS Driven Energy-Efficiency Optimization over Green 5G Mobile Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3092–3107, 2016.

[3] L. Hu, H. Wen, B. Wu, J. Tang, and F. Pan, "Adaptive Secure Transmission for Physical Layer Security in Cooperative Wireless Networks," *IEEE Communications Letters*, vol. 21, no. 3, pp. 524–527, 2017.

[4] L. Hu, H. Wen, B. Wu et al., "Cooperative Jamming for Physical Layer Security Enhancement in Internet of Things," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 219–228, 2018.

[5] A. Duel-Hallen, H. Hallen, and T.-S. Yang, "Long range prediction and reduced feedback for mobile radio adaptive OFDM systems," *IEEE Transactions on Wireless Communications*, vol. 5, no. 10, pp. 2723–2732, 2006.

[6] P. Sharma and K. Chandra, "Prediction of state transitions in Rayleigh fading channels," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 416–425, 2007.

[7] A. Arredondo, K. R. Dandekar, and G. Xu, "Vector channel modeling and prediction for the improvement of downlink received power," *IEEE Transactions on Communications*, vol. 50, no. 7, pp. 1121–1129, 2002.

[8] M. Sternad and D. Aronsson, "Channel estimation and prediction for adaptive OFDM downlinks," in *Proceedings of the Proc. IEEE Veh. Technol*, vol. 2, pp. 1283–1287, 2003.

[9] T. Ding and A. Hirose, "Fading channel prediction based on combination of complex-valued neural networks and chirp Z-transform," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 9, pp. 1686–1695, 2014.

[10] Y. Zhao, H. Gao, NC. Beaulieu, Z. Chen, and H. Ji, "Echo State Network for Fast Channel Prediction in Rice of Fading Scenarios," *IEEE Communications Letters*, vol. 21, no. 3, pp. 672–675, 2017.

[11] X. Zhao, C. Hou, and Q. Wang, "A new SVM-based modeling method of cabin path loss prediction," *International Journal of Antennas and Propagation*, vol. 2013, Article ID 279070, 7 pages, 2013.

[12] K.-T. Kim, D.-K. Seo, and H.-T. Kim, "Efficient radar target recognition using the MUSIC algorithm and invariant features," *IEEE Transactions on Antennas and Propagation*, vol. 50, no. 3, pp. 325–337, 2002.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[14] G. Arulampalam and A. Bouzerdoum, "A generalized feed-forward neural network architecture for classification and regression," *Neural Networks*, vol. 16, no. 5-6, pp. 561–568, 2003.

[15] G. A. Carpenter, *Neural Network Models for Pattern Recognition and Associative Memory*, vol. 2, Elsevier Science Ltd., 1989.

[16] L. Prechelt, "Automatic early stopping using cross validation: Quantifying the criteria," *Neural Networks*, vol. 11, no. 4, pp. 761–767, 1998.

[17] N. C. Beaulieu and Y. Chen, "Maximum likelihood estimation of local average SNR in ricean fading channels," *IEEE Communications Letters*, vol. 9, no. 3, pp. 219–221, 2005.

[18] Y. Chen and N. C. Beaulieu, "Estimation of ricean K parameter and local average SNR from noisy correlated channel samples," *IEEE Transactions on Wireless Communications*, vol. 6, no. 2, pp. 640–648, 2007.

[19] G. L. Stuber, *Principles Mobile Communication*, Kluwer, Boston, Massachusetts, Mass, USA, 2nd edition, 2001.

[20] W. C. Jakes, *Microwave Mobile Communications*, Wiley, New York, NY, USA, 1974.

[21] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[22] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Computation*, vol. 7, no. 2, pp. 219–269, 1995.

[23] R. H. Clarke, "A statistical theory of mobile-radio reception," *Bell Labs Technical Journal*, vol. 47, no. 6, pp. 957–1000, 1968.

[24] 3GPP, Spatial channel model for MIMO simulations 25.996 V6.1.0, 2003, http://www.3gpp.org/.

[25] J. Tang, H. Wen, L. Hu et al., "Associating MIMO beamforming with security codes to achieve unconditional communication security," *IET Communications*, vol. 10, no. 12, pp. 1522–1531, 2016.