remote sensing

MDPI

Article

Multi-Parameter Inversion of AIEM by Using Bi-Directional Deep Neural Network

Yu Wang [1], Zi He [1,*], Ying Yang [1], Dazhi Ding [1], Fan Ding [2] and Xun-Wang Dang [3]

[1] Department of Communication Engineering, Nanjing University of Science and Technology, Nanjing 210094, China; wangyu409@njust.edu.cn (Y.W.); yangying@njust.edu.cn (Y.Y.); dzding@njust.edu.cn (D.D.)
[2] China Ship Development and Design Centre, Wuhan 430064, China; 10725@hbuas.edu.cn
[3] Science and Technology on Electromagnetic Scattering Laboratory, Beijing 100089, China; dangxunwang_207@casic.com.cn
* Correspondence: zihe@njust.edu.cn

Abstract: A novel multi-parameter inversion method is proposed for the Advanced Integral Equation Model (AIEM) by using bi-directional deep neural network. There is a very complex nonlinear relationship between the surface parameters (dielectric constant and roughness) and radar backscattering coefficient. The traditional inverse neural network, which is constructed by using the backscattering coefficients as the input and the surface parameters as the output, leads to bad convergence and wrong results. This is because many sets of surface parameters can get the same backscattering coefficient. Therefore, the proposed bi-directional deep neural network starts with building an AIEM-based forward deep neural network (AIEM-FDNN), whose inputs are the surface parameters and outputs are the backscattering coefficients. In this way, the weights and biases of the forward deep neural network can be optimized and predicted, which can be used for the backward deep neural network (AIEM-BDNN). Then, the multi-parameters are updated by minimizing the loss between the output backscattering coefficients with the measured ones. By inserting a sigmoid function between the input and the first hidden layer, the input multi-parameters can be efficiently approximated and continuously updated. As a result, both the forward and backward deep neural networks can be built with these weights and biases. By sharing the weights and biases of the forward network, the training of the inverse network is avoided. The bi-directional deep neural network can not only predict the backscattering coefficient but can also inverse the surface parameters. Numerical results are given to demonstrate that the RMSE of the backscattering coefficients calculated by the proposed bi-directional neural network can be reduced to 0.1%. The accuracy of the inversion parameters, including the real and imaginary parts of the dielectric constant, the root mean square height and the correlation length, can be improved to 97.56%, 91.14%, 99.04% and 98.45%, respectively. At the same time, the bi-directional neural network also has good accuracy for the inversion of the POLARSCAT measured data.

Keywords: bi-directional neural network; AIEM; surface parameters; backscattering coefficients

check for updates

Citation: Wang, Y.; He, Z.; Yang, Y.; Ding, D.; Ding, F.; Dang, X.-W. Multi-Parameter Inversion of AIEM by Using Bi-Directional Deep Neural Network. *Remote Sens.* **2022**, *14*, 3302. https://doi.org/10.3390/rs14143302

Academic Editor: Yue Wu

Received: 31 May 2022
Accepted: 5 July 2022
Published: 8 July 2022

1. Introduction

The inversion of the surface parameters is the key problem in remote sensing science research [1–4]. Surface parameters can effectively reflect environmental conditions and understand the dynamic information for Earth monitoring. Therefore, there is a great significance in obtaining the surface parameters. Surface parameters inversion is to solve or calculate the target parameters that describe the actual situation of landforms according to the observation information and the forward physical model. How to combine the numerical and experimental results has always been a hot research topic. It has an important guiding significance for overland disturbances and environmental monitors. The inversion of the actual surface information is usually based on a random rough surface scattering model. Over the past few decades, many researchers focused on surface scattering characteristics by using experimental and theoretical methods. The Kirchhoff Approximation (KA) was mostly applied to large-scale rough surfaces [5,6]. On the other hand, for small-scale rough surfaces, the Small Perturbation Model (SPM) was developed [7,8]. Subsequently, the Small Slope Approximation (SSA), which was proposed by Voronovich, combines the perturbation theory with the tangent approximation [9,10]. It should be noted that the KA is only suitable for a large curvature, while the SPM is only suitable for small roughness. Therefore, the Integral Equation Model (IEM) [11] was proposed by Fung to bridge the KA and SPM. The dependence of the surface height on the phase of the Green's function was ignored for the traditional IEM, which led to a big error. Then, a series of modified schemes were proposed to increase the accuracy, such as the Advanced Integral Equation Model (AIEM) and its derivatives [12,13]. Therefore, the AIEM can be used as an efficient tool to model the landform for its robustness and scalability.

The research methods in this area are generally divided into the empirical formula method, intelligent optimization algorithm and neural network method. In the past decades, the semi-empirical models were used as one of the most popular methods to predict the parameters [14–16]. This method is to summarize the laws of a large number of measured data and express them with simple functions. Inspired by evolutionary phenomena in nature, many intelligent optimization algorithms have emerged, such as the GA (Genetic Algorithm) and PSO (Particle Swarm Optimization). Such methods have been widely used for hydrogeological parameters and rough surface parameters inversion [17,18]. The core idea of the intelligent optimization algorithm is to use the algorithm to traverse the model space constructed by all the parameters to obtain the optimal solution of the objective function. However, it is difficult to obtain the global optimal solution using these methods, and only a small number of parameters can be inverted. At present, neural networks are being widely used in engineering fields such as machinery, materials and architecture, and their applications can be traced back to the late 1980s. Neural networks can perform complex data processing and are usually used to complete classification tasks and function approximation tasks. Therefore, a neural network is a promising tool for solving the inverse problems arising from its generalization ability. In [3], a back propagation Neural Network (BP) based on IEM was developed to inverse the surface parameters. In [19,20], neural networks with different structures were used for the prediction of metasurface geometric parameters or color parameters. Meanwhile, a Convolutional Neural Network (CNN) has been used in SAR target recognition and terrain classification [21–23]. In [24], a CNN and Generative Adversarial Network (GAN) were combined to extract simulation parameters from SAR images.

In this paper, a novel bi-directional DNN (deep neural network) is proposed to predict the multi-parameters of the AIEM. The proposed bi-directional DNN consists of two DNNs. Both DNNs share the same network structure and the same set of network weights. The bi-directional DNN can successfully complete the two tasks of predicting backscattering coefficients and inverting surface parameters. At first, a forward DNN needs to be established. This forward DNN takes the surface parameters as the input and the backscattering coefficients as the output. After training, this network can fit the AIEM model well. Then, a backward DNN is constructed by reusing the network structure of the forward DNN and the weights after training. Before backward network training, the input surface parameters need to be initialized as constants. Finally, the initialized surface parameters can be updated by calculating the loss of the output backscattering coefficients and the actual backscattering coefficients. The traditional inverse neural network, which is constructed by using the backscattering coefficients as the input and the surface parameters as the output, leads to a bad convergence and wrong result. However, the proposed bi-directional deep neural network is proposed to overcome these problems. Compared with the BP neural network, the proposed bi-directional network has a higher inversion accuracy. To verify the inversion accuracy of the bi-directional network, POLARSCAT [25–27] measured data on bare soil surfaces under three different roughness and humidity conditions was used.

The numerical results showed that the bi-directional network has high accuracy for the prediction of backscattering coefficients and the inversion of surface parameters.

2. Materials and Methods

2.1. Experimental Data

In this study, the training data of the bi-directional network was obtained based on the mapping relationship between the surface parameters and radar observations. In fact, training data satisfying such conditions cannot be obtained from the point datasets measured in the field. The AIEM model can simulate the backscattering characteristics under various surface conditions. Given the range of variations in the surface permittivity, the root mean square height (RMS) height and correlation length of interest bi-directional neural network training data can be generated by the AIEM model [12,28].

The general formula of the AIEM model is shown in Figure 1.

$$\sigma_{qp}(s) \;=\; \sigma_{qp}^{k} + \sigma_{qp}^{kc} + \sigma_{qp}^{c} \tag{1}$$

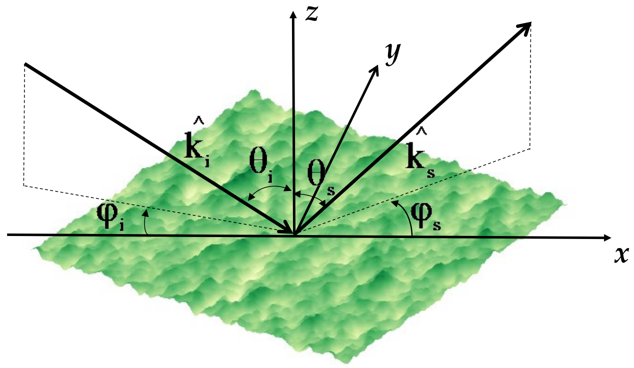


Figure 1. Schematic diagram of scattering from rough surfaces.

It can be seen that the scattering coefficient is composed of Kirchhoff terms σ_{qp}^{k}, cross terms σ_{qp}^{kc} and compensation terms σ_{qp}^{c}. The explicit form of AIEM can be given as

$$\sigma_{qp}(s) = \begin{array}{l} \frac{k^2}{2} e^{-\sigma^2(k_{sz}^2+k_z^2)} \\ \cdot \sum\limits_{n=1}^{\infty} \frac{\sigma^{2n}}{n!} \left| I_{qp}^{n} \right|_2 S^{(n)}(k_{sx}-k_x, k_{sy}-k_y) \end{array} \tag{2}$$

where k is the incident wave number, σ^2 represents the variance of the surface height and $S^{(n)}(k_{sx}-k_x, k_{sy}-k_y)$ denotes the surface roughness spectrum of the surface in terms of the nth power of the surface correlation function by two-dimensional Fourier transform.

As shown in Figure 1, the incident and scattered wave vectors can be defined as

$$k_x \;=\; k\sin\theta_i\cos\varphi_i \;\; ; k_y \;=\; k\sin\theta_i\sin\varphi_i ; k_z \;=\; -k\cos\theta_i \tag{3}$$

$$k_{sx} \;=\; k\sin\theta_s\cos\varphi_s \;\; ; k_{sy} \;=\; k\sin\theta_s\sin\varphi_s ; k_{sz} \;=\; k\cos\theta_s \tag{4}$$

where θ_i and φ_i are the incident angle, and θ_s and φ_s are the scattering angle. The backscattering direction is at $\theta_i = \theta_s$, $\varphi_s = \varphi_i + 180^\circ$.

POLARSCAT is a polarizing scatterometer that operates on different bare surfaces, each with wet and dry conditions. The polarimetric measurements are conducted at the L-, C- and X-band frequencies at incident angles ranging from 10° to 70°. In this paper, the experimental data in the L- (the center frequency is 1.5 GHz) and X-bands (the center frequency is 4.75 GHz) are selected. As shown in Table 1, three soils of different roughness were measured in dry and wet conditions. Where σ is the RMS height, l is the correlation length and $k = 2\pi/\lambda$, ($\lambda = c/f, c = 3 \times 10^8$). The RMS height ranged from 0.40 cm to

1.12 cm, and the correlation length ranged from 8.4 cm to 9.9 cm. In [25–27], for the three surfaces (S1–S3), the measured autocorrelation function was found to be closer in shape to an exponential function.

Table 1. POLARSCAT measured parameters.

Surface Number	Freq. (GHz)	$k\sigma$	kl	$\sigma\ (cm)$	$l\ (cm)$	ε_r	ε_r''
S1-dry	1.5 GHz	0.13	2.6	0.40	8.4	7.99	2.02
	4.75 GHz	0.4	8.4			8.77	1.04
S1-wet	1.5 GHz	0.13	2.6			15.57	3.71
	4.75 GHz	0.4	8.4			15.42	2.15
S2-dry	1.5 GHz	0.1	3.1	0.32	9.9	5.85	1.46
	4.75 GHz	0.32	9.8			6.66	0.68
S2-wet	1.5 GHz	0.1	3.1			14.43	3.47
	4.75 GHz	0.32	9.8			14.47	1.99
S3-dry	1.5 GHz	0.35	2.6	1.12	8.4	7.70	1.95
	4.75 GHz	1.11	8.4			8.50	1.00
S3-wet	1.5 GHz	0.35	2.6			15.34	3.66
	4.75 GHz	1.11	8.4			15.23	2.12

2.2. Method

In this section, it will be introduced separately from the overall framework of the bi-directional network, the structure of the forward network and the structure of the reverse network. At the same time, the workflow of the bi-directional network will be introduced in detail.

2.2.1. Framework of the Bi-Directional Deep Neural Network

There are usually two smart methods for solving inverse problems, namely the optimization algorithm and neural network inverse modeling method. The core idea of the optimization algorithm is to traverse the model space constructed by all parameters to obtain the optimal solution of the objective function. However, this kind of method needs to manually set the range of each parameter, and it is easy to fall into the local optimal solution when dealing with complex problems. In [29], a genetic algorithm was used to invert the surface parameters. It is often necessary to perform multiple searches to select the optimal solution, and the accuracy is not high. Another method is to use the backscattering coefficients as the input and the surface parameters as the output and use the neural network to directly construct the inverse model. However, since there is no exact analytical formula from the backscattering coefficients to the surface parameters, at the same time, the non-uniqueness of the dataset itself will make the overall training of the dataset difficult for the inverse model, thus affecting the inversion accuracy.

In this paper, a novel DNN-based surface parameters inversion method is proposed. As shown in Figure 2, this framework consists of two DNNs, namely an AIEM-Based Forward Deep Neural Network (AIEM-FDNN) and AIEM-Based Backward Deep Neural Network (AIEM-BDNN). The same network structure and weights are shared by them. The AIEM-FDNN takes the surface parameters as the input and the backscattering coefficients as the output. After training, it can be used to quickly calculate the backscattering coefficients outside the dataset. The AIEM-BDNN can be formed by reusing the network structure and well-trained weights of AIEM-FDNN. The input nodes need to be set as the variables. First, the input surface parameters are randomly initialized as constants. Then, the loss between the output backscattering coefficients and the actual backscattering coefficients will be calculated by the AIEM-BDNN. Finally, based on the back propagation of the error, the initialized surface parameters are continuously updated by the optimizer until the error converges into a sufficiently small value. Meanwhile, the updated surface parameters are the inversion results of the AIEM-BDNN based on this set of backscattering coefficients.

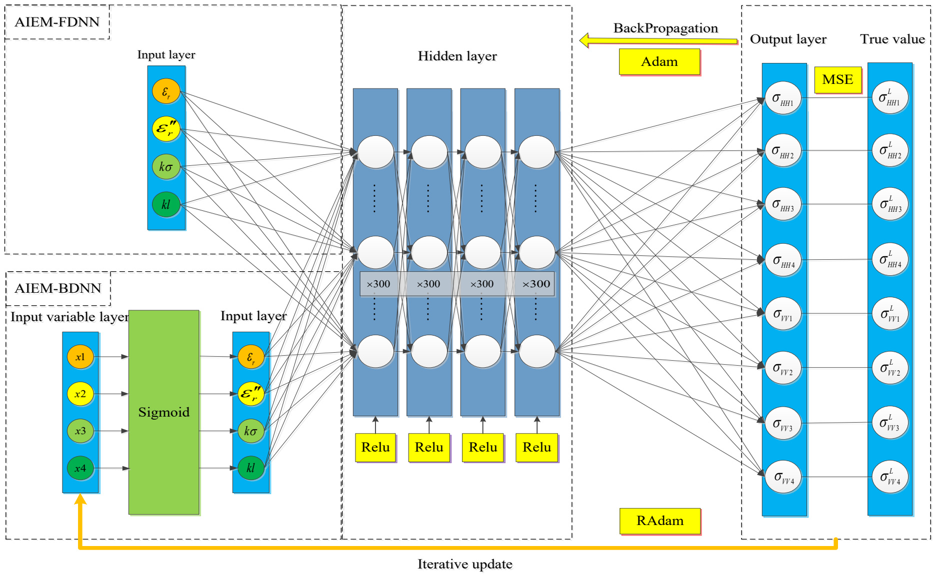


Figure 2. The framework for the proposed AIEM-based bi-directional deep neural network.

The flowchart of the overall working process of the bi-directional deep neural network is provided in Figure 3. The workflow of the AIEM-FDNN and AIEM-BDNN will be presented in detail in the following two parts.

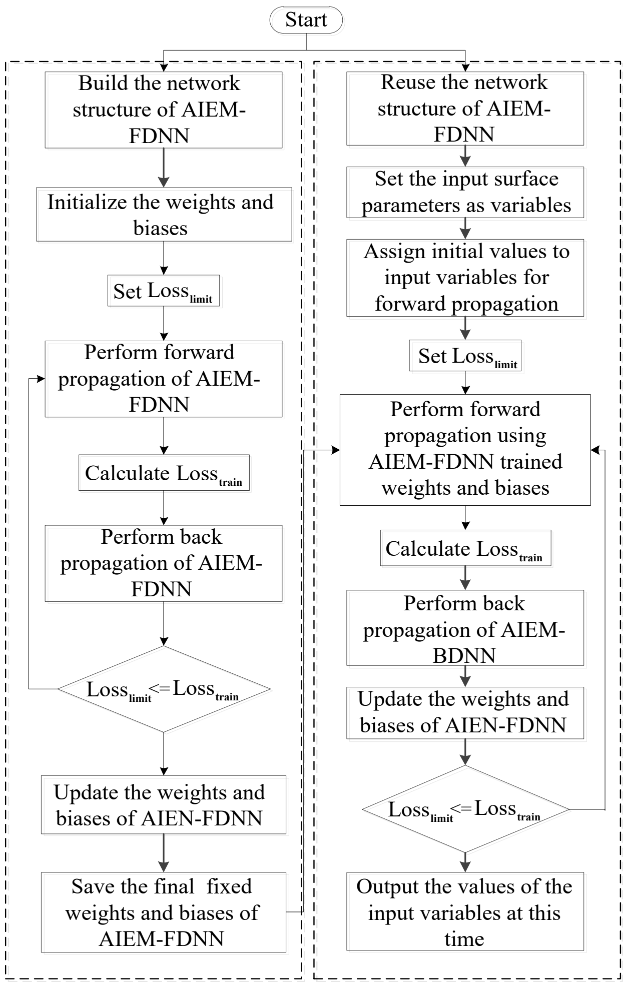


Figure 3. Flowchart of the working process of the bi-directional DNN.

2.2.2. AIEM-Based Forward Deep Neural Network

As shown in Figure 1, the AIEM-FDNN is a fully connected network that contains an input layer, multiple hidden layers and an output layer. Its input is the surface parameters, including the real part ε_r and imaginary part ε_r'' of the dielectric constant, the root mean square height $k\sigma$ and the correlation length kl, and the output is the backscattering coefficients σ_{HH}, σ_{VV}.

The AIEM-FDNN is designed to calculate the backscattering coefficients. The trained AIEM-FDNN has similar computational accuracy to the AIEM model, and it is less complex to calculate. Since the AIEM-BDNN used for surface parameters inversion uses the network structure of AIEM-FDNN and the weights after training, the accuracy of the AIEM-FDNN directly affects the performance of the entire bi-directional DNN. The training process of the AIEM-FDNN consists of two stages: forward propagation and back propagation. The forward propagation is to calculate the loss of the output backscattering coefficients and the actual backscattering coefficients according to the current network weights. The back propagation is to update the weights using gradient descent techniques based on the current loss.

The forward propagation calculation process of AIEM-FDNN can be given as

$$\mathbf{Z}_{\mathbf{AF}}^{0} = [\varepsilon_r, \varepsilon_r'', k\sigma, kl]_A \tag{5}$$

$$\mathbf{Z}_{\mathbf{AF}}^{\mathbf{i}} = g_i\left(\mathbf{W}_{\mathbf{AF}}^{\mathbf{i}} \cdot \mathbf{Z}_{\mathbf{AF}}^{\mathbf{i\text{-}1}} + \mathbf{b}_{\mathbf{AF}}^{\mathbf{i}}\right)(i = 1, \cdots, N) \tag{6}$$

$$\mathbf{Z}_{\mathbf{AF}}^{N} = g_N\left(\mathbf{W}_{\mathbf{AF}}^{N} \cdot \mathbf{Z}_{\mathbf{AF}}^{N\text{-}1} + \mathbf{b}_{\mathbf{AF}}^{N}\right) \tag{7}$$

$$[\boldsymbol{\sigma}_{\mathbf{HH}}, \boldsymbol{\sigma}_{\mathbf{VV}}] = \mathbf{Z}_{\mathbf{AF}}^{\mathbf{N}} \tag{8}$$

where $\mathbf{Z}_{\mathbf{AF}}^{0}$ and $\mathbf{Z}_{\mathbf{AF}}^{\mathbf{N}}$ represent the input surface parameters and the output backscattering coefficients for HH and VV polarizations for different incident angles, respectively. $\mathbf{Z}_{\mathbf{AF}}^{\mathbf{i}}(i = 1, \cdots, N)$ represents the calculation result of the ith layer after the activation function. $\mathbf{W}_{\mathbf{AF}}^{\mathbf{i}}$ represents the weights matrix from the *(i-1)*th layer to the ith layer. $\mathbf{b}_{\mathbf{AF}}^{\mathbf{i}}$ represents the biases of the ith layer, and $g_i(\cdot)$ represents the nonlinear activation function of the ith layer. As shown in Figure 3, the calculated loss between the output backscattering coefficients and the actual backscattering coefficients will be calculated. The loss function of AIEM-FDNN is defined as the mean square error, which can be expressed as

$$Loss_{AF} = \frac{1}{n}\sum_{j=1}^{n}\left[\left(\sigma_{HH,j}^{L} - \sigma_{HH,j}\right)^2 + \left(\sigma_{VV,j}^{L} - \sigma_{VV,j}\right)^2\right] \tag{9}$$

where $\sigma_{HH,j}^{L}, \sigma_{VV,j}^{L}$ represents the actual backscattering coefficients of the HH and VV polarization for the jth incident angle. The back propagation of AIEM-FDNN is based on the chain derivation rule. $\frac{\partial Loss_{AF}}{\partial \mathbf{W}_{\mathbf{AF}}^{\mathbf{i}}}$ and $\frac{\partial Loss_{AF}}{\partial \mathbf{b}_{\mathbf{AF}}^{\mathbf{i}}}$ are calculated to update $\mathbf{W}_{\mathbf{AF}}^{\mathbf{i}}$ and $\mathbf{b}_{\mathbf{AF}}^{\mathbf{i}}$ until $Loss_{AF}$ converges to a minimum. The calculation process can be given as

$$\mathbf{E}_{\mathbf{AF}}^{\mathbf{N}} = -\left(\mathbf{y}_{\mathbf{label}} - g_N\left(\begin{array}{c}\mathbf{W}_{\mathbf{AF}}^{\mathbf{N}} \cdot \mathbf{Z}_{\mathbf{AF}}^{\mathbf{N\text{-}1}} \\ +\mathbf{b}_{\mathbf{AF}}^{\mathbf{N}}\end{array}\right)\right) \circ g'_N\left(\begin{array}{c}\mathbf{W}_{\mathbf{AF}}^{\mathbf{N}} \cdot \mathbf{Z}_{\mathbf{AF}}^{\mathbf{N\text{-}1}} \\ +\mathbf{b}_{\mathbf{AF}}^{\mathbf{N}}\end{array}\right) \tag{10}$$

$$\mathbf{E}_{\mathbf{AF}}^{\mathbf{i}} = \left(\left(\mathbf{W}_{\mathbf{AF}}^{\mathbf{i+1}}\right)^T \cdot \mathbf{E}_{\mathbf{AF}}^{\mathbf{i+1}}\right) \circ g'_i\left(\begin{array}{c}\mathbf{W}_{\mathbf{AF}}^{\mathbf{i}} \cdot \mathbf{Z}_{\mathbf{AF}}^{\mathbf{i\text{-}1}} \\ +\mathbf{b}_{\mathbf{AF}}^{\mathbf{i}}\end{array}\right) \tag{11}$$

$$\frac{\partial Loss_{AF}}{\partial \mathbf{W}_{\mathbf{AF}}^{\mathbf{i}}} = \mathbf{E}_{\mathbf{AF}}^{\mathbf{i}} \cdot \left(\mathbf{Z}_{\mathbf{AF}}^{\mathbf{i}-1}\right)^T \tag{12}$$

$$\frac{\partial Loss_{AF}}{\partial \mathbf{b}_{\mathbf{AF}}^{\mathbf{i}}} = \mathbf{E}_{\mathbf{AF}}^{\mathbf{i}} \tag{13}$$

where $\mathbf{E_{AF}^{N}}$ represents the error vector in the output layer of AIEM-FDNN, $g'_i(\cdot)$ is the derivative of the activation function. $\mathbf{E_{AF}^{i}}$ is the error vector in the ith layer, and $\circ$ is the Hadamard product. Finally, the formulas for updating the weights and biases can be given as

$$\mathbf{W_{AF}^{i}} = \mathbf{W_{AF}^{i}} - \eta_{AF}\frac{\partial Loss_{AF}}{\partial \mathbf{W_{AF}^{i}}} \quad (14)$$

$$\mathbf{b_{AF}^{i}} = \mathbf{b_{AF}^{i}} - \eta_{AF}\frac{\partial Loss_{AF}}{\partial \mathbf{b_{AF}^{i}}} \quad (15)$$

where η_{AF} represents the learning rate of the AIEM-FDNN.

2.2.3. AIEM-Based Backward Deep Neural Network

The AIEM-BDNN is constructed by directly reusing the network structure of the AIEM-FDNN and loading the training weights and biases to invert the surface parameters. Simply put, it is only necessary to set the input node of the trained AIEM-FDNN as variables. The training process of the AIEM-BDNN also includes forward propagation and back propagation, but it is different from the training object of AIEM-FDNN. The training objects of AIEM-FDNN are the weights and biases of the network, while the training objects of the AIEM-BDNN are the input surface parameters of the network. The AIEM-BDNN is trained by giving a set of backscattering coefficients to be inverted. By initializing the input surface parameters as constants, the forward propagation of the AIEM-BDNN is performed to calculate the backscattering coefficients. Back propagation is performed according to the loss between the output backscattering coefficients and the true backscattering coefficients. Finally, the initialized surface parameters are continuously updated until the loss converges to a small enough value. The last surface parameters updated are the inversion values.

The forward propagation calculation process of the AIEM-FDNN can be given as

$$\mathbf{Z_{AB}^{0}} = [\varepsilon_r, \varepsilon_r'', k\sigma, kl]_B \quad (16)$$

$$\mathbf{Z_{AB}^{i}} = g_i\left(\mathbf{W_{AF}^{i}} \cdot \mathbf{Z_{AB}^{i\text{-}1}} + \mathbf{b_{AF}^{i}}\right)(i = 1, \cdots, N) \quad (17)$$

$$\mathbf{Z_{AB}^{\mathit{N}}} = g_N\left(\mathbf{W_{AF}^{\mathit{N}}} \cdot \mathbf{Z_{AB}^{\mathit{N}-1}} + \mathbf{b_{AF}^{\mathit{N}}}\right) \quad (18)$$

$$[\boldsymbol{\sigma}_{\mathbf{HH}}, \boldsymbol{\sigma}_{\mathbf{VV}}] = \mathbf{Z_{AB}^{N}} \quad (19)$$

where $[\varepsilon_r, \varepsilon_r'', k\sigma, kl]_B$ are randomly initialized surface parameters. $\mathbf{W_{AF}^{i}}$ represents the weights matrix from the *(i-1)*th layer to the ith layer of AIEM-FDNN. $\mathbf{b_{AF}^{i}}$ represents the biases of the ith layer of the AIEM-FDNN. Since the AIEM-FDNN has been trained, $\mathbf{W_{AF}^{i}}$ and $\mathbf{b_{AF}^{i}}$ have been fixed. They will not be updated in both the forward and backward propagation of the AIEM-BDNN. $g_i(\cdot)$ represents the nonlinear activation function of the ith layer of the AIEM-FDNN. $\mathbf{Z_{AB}^{i}}(i = 1, \cdots, N)$ represents the calculation results of the ith layer of the AIEM-BDNN after the activation function. The loss function of the AIEM-BDNN is also defined as the mean squared error, which can be expressed as

$$Loss_{AB} = \frac{1}{n}\sum_{j=1}^{n}\left[\left(\sigma_{HH,j}^{L} - \sigma_{HH,j}\right)^2 + \left(\sigma_{VV,j}^{L} - \sigma_{VV,j}\right)^2\right] \quad (20)$$

The back propagation of the AIEM-BDNN is also based on the chain derivation rule. $\frac{\partial Loss_{AB}}{\partial \mathbf{Z_{AB}^{0}}}$ is calculated to update $\mathbf{Z_{AB}^{0}}$ until $Loss_{AB}$ converges to a minimum. The calculation process can be given as

$$\mathbf{E_{AB}^{N}} = -\left(\mathbf{y_{label}} - g_N\begin{pmatrix} \mathbf{W_{AF}^{N}} \cdot \mathbf{Z_{AB}^{N-1}} \\ +\mathbf{b_{AF}^{N}} \end{pmatrix}\right) \circ g'_N\begin{pmatrix} \mathbf{W_{AF}^{N}} \cdot \mathbf{Z_{AB}^{N-1}} \\ +\mathbf{b_{AF}^{N}} \end{pmatrix} \quad (21)$$

$$\mathbf{E}^{\mathbf{i}}_{\mathbf{AB}} = \left(\left(\mathbf{W}^{\mathbf{i+1}}_{\mathbf{AF}} \right)^T \cdot \mathbf{E}^{\mathbf{i+1}}_{\mathbf{AB}} \right) \circ g'_i \left(\begin{array}{c} \mathbf{W}^{\mathbf{i}}_{\mathbf{AF}} \cdot \mathbf{Z}^{\mathbf{i}-1}_{\mathbf{AB}} \\ +\mathbf{b}^{\mathbf{i}}_{\mathbf{AF}} \end{array} \right) \tag{22}$$

$$\frac{\partial Loss_{AB}}{\partial \mathbf{Z}^0_{\mathbf{AB}}} = \left(\mathbf{W}^1_{\mathbf{AF}} \right)^T \cdot \mathbf{E}^1_{\mathbf{AB}} \tag{23}$$

where $\mathbf{E}^{\mathbf{N}}_{\mathbf{AB}}$ represents the error vector in the output layer of the AIEM-BDNN, and $g'_i(\cdot)$ is the derivative of activation function of AIEM-FDNN. $\mathbf{E}^{\mathbf{i}}_{\mathbf{AB}}$ is the error vector in the ith layer, and $\circ$ is the Hadamard product. Finally, the formulas for updating $\mathbf{Z}^0_{\mathbf{AB}}$ can be given as

$$\mathbf{Z}^0_{\mathbf{AB}} = \mathbf{Z}^0_{\mathbf{AB}} - \eta_{AB} \frac{\partial Loss_{AB}}{\partial \mathbf{Z}^0_{\mathbf{AB}}} \tag{24}$$

in which η_{AB} represents the learning rate of the AIEM-BDNN.

From the formula derivation of AIEM-FDNN and AIEM-BDNN, it can be seen that the training purpose of the AIEM-FDNN is to update the weights and biases of the network. Instead, AIEM-BDNN uses the weights and biases that AIEM-FDNN has already trained and fixed. Therefore, its training purpose is only to update the input parameters. It can be seen that the AIEM-FDNN and AIEM-BDNN are closely related. The quality of the AIEM-FDNN training will directly affect the inversion accuracy of the AIEM-BDNN. Therefore, using the bi-directional network to invert the surface parameters, we first need to ensure that the accuracy of the backscattering coefficients calculated by the AIEM-FDNN is high enough. The pseudocode of the bi-directional deep neural network was added as Appendix A to the article.

3. Results

3.1. Performance of the AIEM-Based Forward Deep Neural Network

The selection of the datasets is crucial for the training of neural networks. Since the AIEM model can simulate the backscattering characteristics under various surface parameters, the training set required for the AIEM-FDNN can be generated as long as the variation range of the surface parameters is given. As shown in Table 2, the range of each surface parameter for generating the dataset is given. The range of the radar incident angle is set from 20° to 50°. Four surface parameters, namely the real and imaginary parts of the dielectric constant, the normalized root mean square height and the normalized correlation length, are used as the input of the AIEM-FDNN, while the backscattering coefficients for HH and VV polarization are the output. The sampling interval of the real part and imaginary part of the dielectric constant is 1.2 and 1, respectively. The sampling interval of the normalized root mean square height is 0.1. The normalized relative length is 0.7. A number of (21,009) sets of surface parameter combinations were generated by a cyclic combination within the range of surface parameters, and the corresponding backscattering coefficients were calculated by using the AIEM model. Many (3000) groups were selected as the validation set, and 1300 groups were selected as the test set.

Table 2. Surface parameters and radar parameters.

Parameter	Value
Real part of the dielectric constant(ε_r)	2–26
Imaginary part of the dielectric constant (ε''_r)	0.1–10.1
Normalized root mean square height height ($k\sigma$)	0.1–1
Normalized relative lenght (kl)	1–10.8
Range of incident angle (θ_i)	20°–50°
Polarization mode	HH, VV
$k\sigma/kl$	0.01–0.5
$\varepsilon''_r/\varepsilon_r$	0–0.5
Surface roughness spectrum (S)	Exponential

Next, the AIEM-FDNN is built for forward prediction. After continuous testing and adjustment of the hyperparameters, the hyperparameter settings shown in Table 3 are finally determined. There are four hidden layers added in the AIEM-FDNN, and each layer has 300 neurons. The activation function of each hidden layer adopts the ReLU function. Then, using the mean squared error (MSE) as the loss function, the error between the output value and the true value for each epoch is calculated. At the same time, the popular optimizer Adam is used to realize the back propagation. Finally, the continuous updating of the weights and the biases can be realized. A decaying learning rate is used, so that the training loss can converge more smoothly. Setting the batch size to 20, the network converges when the epoch is equal to 1300.

Table 3. Training the hyperparameters of the AIEM-FDNN.

Parameter	**Value**
Weight initialization method	Uniform distribution initialization
Activation function	ReLU
Loss function	MSE
Optimizer	Adam
Learning rate	0.001
Learning decay rate	0.9
Hidden layers	4
Hidden neurons	300
Epoch	1300
Batch size	20

The test set was used to test the ability of the AIEM-FDNN to predict backscattering coefficients. As shown in Table 4, the RMSE between the output backscattering coefficients for HH and VV polarizations for different incident angles and the actual backscattering coefficients for HH and VV polarizations for different incident angles can be reduced to be less than 0.1%. It can be seen that the training of the AIEM-FDNN is successful, and the accuracy is high. The trained AIEM-FDNN has almost the same computational accuracy as the AIEM model. The 21,009 sets of data generated by the AIEM model need 75.6 s, with 7.34 s for the proposed AIEN-FDNN. Therefore, the AIEM-FDNN has a faster computation speed when faced with a large amount of data generation tasks.

Table 4. The RMSE between the output backscattering coefficients and the actual backscattering coefficients for the proposed AIEM-FDNN with $\varphi = 0° - 180°$.

Polarization	**Incident Angle (θ)**	**RMSE**
VV	20°	0.1055%
VV	30°	0.0585%
VV	40°	0.0557%
VV	50°	0.0708%
HH	20°	0.0905%
HH	30°	0.0589%
HH	40°	0.0661%
HH	50°	0.0655%

At the same time, the degree of agreement between the backscattering coefficients calculated by AIEM-FDNN and the measured data has a great influence on the accuracy of the bi-directional network inversion of actual surface parameters. POLARSCAT measured data are used to test the AIEM-FDNN. The comparison of backscattering coefficients of the AIEM (AIEM-VV and AIEM-HH), POLARSCAT measured data (data_VV and data_HH) and AIEM-FDNN (AIEM-FDNN_VV and AIEM-FDNN_HH) for exponential correlated surface are shown in Figure 4. It can be seen that the three have good consistency. This lays a good foundation for the AIEM-BDNN to invert POLARSCAT measured parameters.

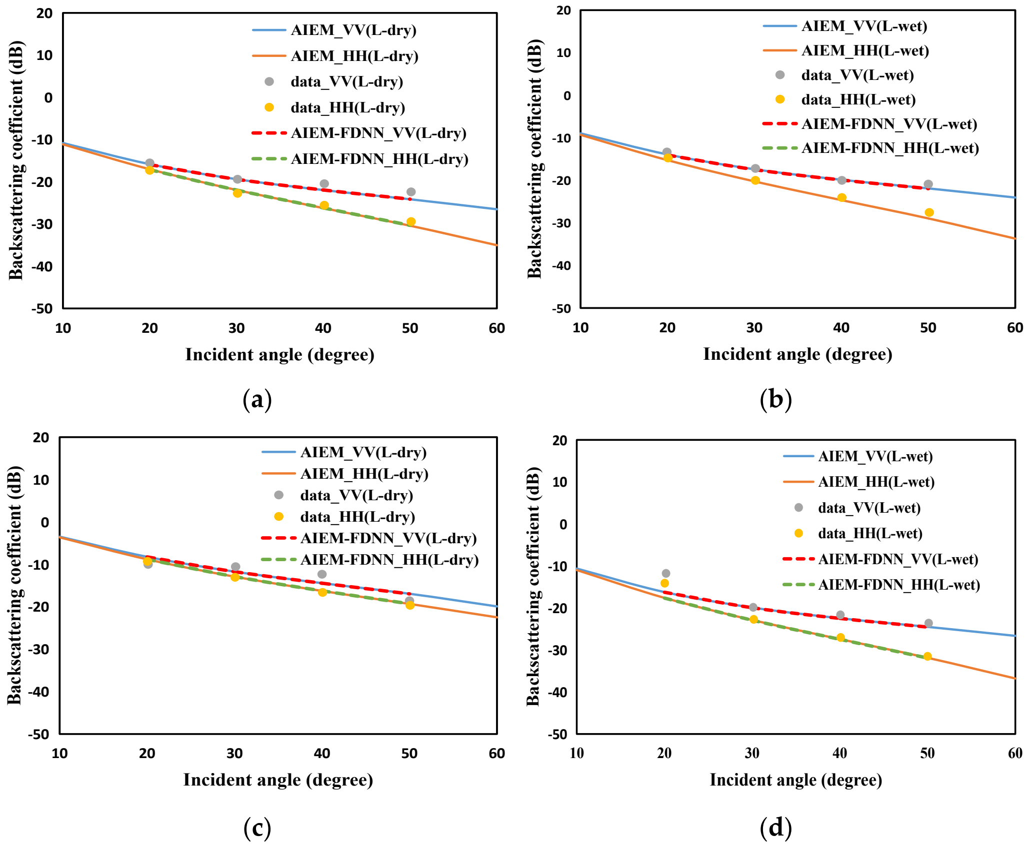


Figure 4. Comparison of backscattering coefficients of the AIEM, POLARSCAT measured data and AIEM−FDNN for exponential correlated surface with (**a**) $\varepsilon_r = 7.99$, $\varepsilon_r'' = 2.02$, $k\sigma = 0.13$ and $kl = 2.6$ at 1.5 GHz; (**b**) $\varepsilon_r = 15.57$, $\varepsilon_r'' = 3.71$, $k\sigma = 0.13$ and $kl = 2.6$ at 1.5 GHz; (**c**) $\varepsilon_r = 7.7$, $\varepsilon_r'' = 1.95$, $k\sigma = 0.35$, $kl = 2.6$ at 1.5 GHz and (**d**) $\varepsilon_r = 14.43$, $\varepsilon_r'' = 3..47$, $k\sigma = 0.1$ and $kl = 3.1$ at 1.5 GHz.

3.2. Performance of the AIEM-Based Backward Deep Neural Network

AIEM-BDNN is designed to complete the surface parameters inversion task. It can be established by reusing the network structure of the AIEM-FDNN and the well-trained weights and biases. It is worth noting that the weights and biases of the AIEM-FDNN have been fixed and will not change after being reused by the AIEM-BDNN. Simply put, only the input surface parameters of the AIEM-BDNN are updated during training. The hyperparameters used by the AIEM-FDNN are not suitable for the AIEM-BDNN. After continuous tuning, the RAdam optimizer was chosen instead of the Adam optimizer. Xavier Initialization is chosen as the initialization method of the input surface parameters.

Two outstanding problems were found in the experiments, one of which is that the surface parameters are not updated in the desired direction. As a result, although the training loss can converge normally, the surface parameters obtained by the final inversion often deviate from the conventional parameter space. The update of the surface parameters is not automatically limited to the respective data ranges shown in Table 2, and even negative values may appear. The reason for this is that the AIEM-BDNN can accept arbitrary update parameters due to the training mechanism of the DNN, and even the wrong parameter combination can calculate the same result as the real value. In order to limit the update range of the input surface parameters, before AIEM-FDNN training, the input surface parameters are normalized by the method of Min–Max_scale, and the parameters can be limited to 0–1. Next, a sigmoid layer is inserted between the input layer and the first hidden layer of AIEM-BDNN. As a commonly used nonlinear function, the sigmoid function can limit any input value between 0 and 1. In this way, you do not need the need to care whether the updated surface parameters are out of a reasonable range, because no matter how unreasonable the value of the updated surface parameter is, the sigmoid function will adjust it to the normal range. It should be noted here that, although the update object of the network is still the input surface parameters, the real input parameters of the AIEM-BDNN have become the values adjusted by the sigmoid function. At the same time, the value adjusted by the sigmoid function will also be used as the surface parameters inversed by AIEM-BDNN.

Another problem in the experiment is that there is a "premature" phenomenon when the input surface parameters are updating. This phenomenon is reflected in the fact that the training error cannot converge in the early stage of training. The reason is that, in the early stage of network training, the gradient decreases sharply, resulting in the slow update of neurons and ineffective learning. To alleviate such problem, the RAdam optimizer is used instead of the Adam optimizer, and the Xavier initialization method is used. The RAdam optimizer introduces a warm-up mechanism based on the commonly used Adam optimizer. Simply put, it is to use a small learning rate in the early stage of network training, so that the early training can be carried out smoothly and avoid excessive variance. The Xavier Initialization method will control the variance of the initial value within an appropriate range, usually making the variance of the initial value 1. It is also possible to choose to use the solution in [30,31]. By scanning all the variable hyperparameters in the AIEM-BDNN and recording the loss value, the one with the smallest loss is selected as the optimal inversion result. After continuous testing and adjustment of the hyperparameters, the hyperparameter settings shown in Table 5 are finally determined.

Table 5. Training hyper-parameters of AIEM-BDNN.

Parameter	Value
Input value initialization method	Xavier Initialization
Activation function	ReLU
Loss function	MSE
Optimizer	RAdam
Learning rate	0.001
Learning decay rate	0.9
Hidden layers	4
Hidden neurons	300
Epoch	10,000

Many (1300) sets of test sets are used to examine the inversion accuracy of the AIEM-BDNN. As shown in Figure 5, the comparison of the true surface parameters and the AIEM-BDNN predicted surface parameters is given. The numerical results show that the predicted surface parameters and the true surface parameters are concentrated near the contour, which shows that the accuracy of the predicted parameters is high. The correlation coefficient between the two is calculated, respectively, 97.56% (ε_r), 91.14% (ε_r''), 99.04% ($k\sigma$) and 98.45% (kl), as shown in Table 6.

Table 6. Inversion accuracy of the bi-directional neural networks.

Parameter	RMSE	Similarity(1-RMSE)
ε_r	0.0244	97.56%
ε_r''	0.0886	91.14%
$k\sigma$	0.0096	99.04%
kl	0.0155	98.45%

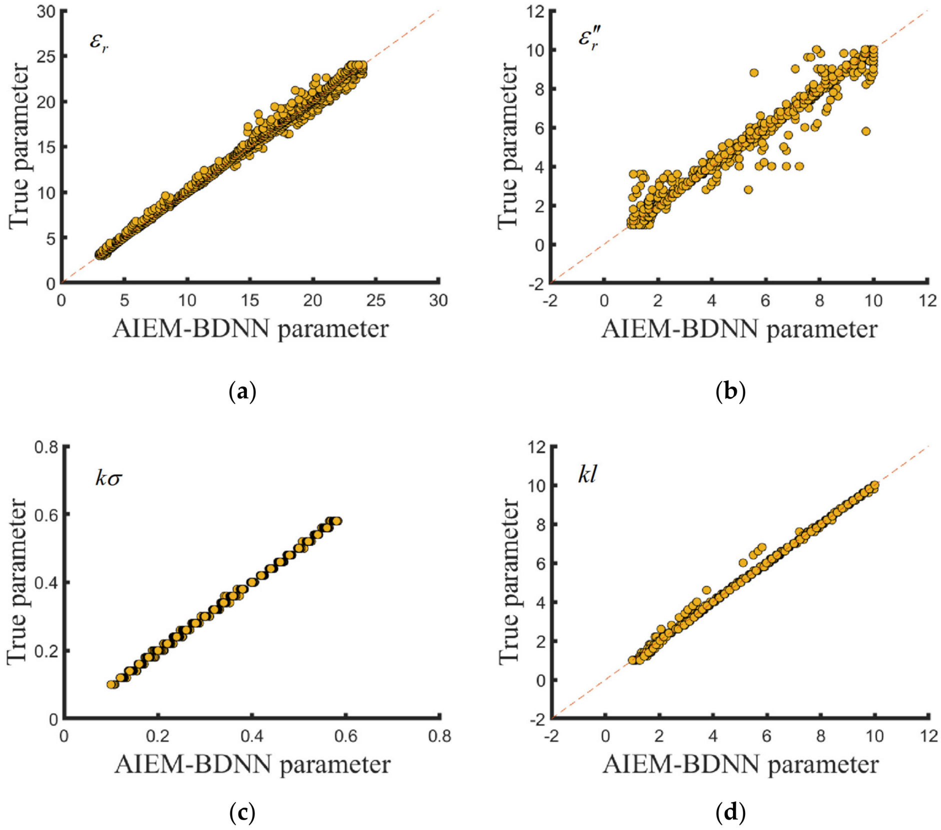


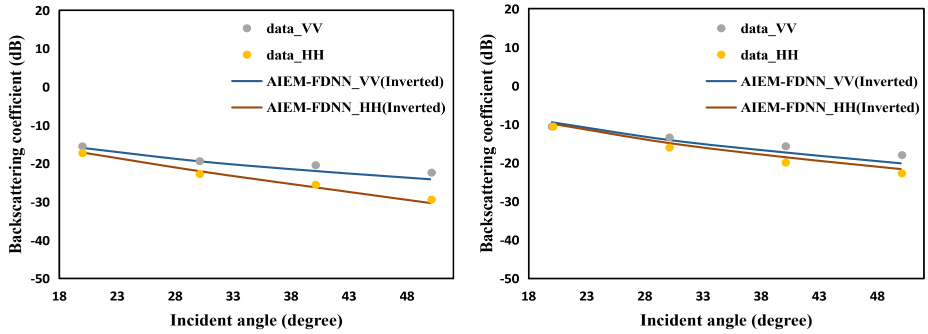
Figure 5. Comparison of true parameters and the AIEM−BDNN predicted parameters for: (**a**) the real part of the dielectric constant, (**b**) the imaginary part of the dielectric constant, (**c**) the normalized root mean square height and (**d**) the normalized correlation length.

As shown in Table 7, twelve sets of inversion results between POLARSCAT measured data and inverted by the AIEM-BDNN are compared. Three exponential distribution surfaces of POLARSCAT measured data are selected. As we can see, the comparison of the inversion results with the measured surface parameters can achieve satisfactory accuracy.

Table 7. Comparison of the surface parameters between POLARSCAT measured data and inverted by the AIEM-BDNN.

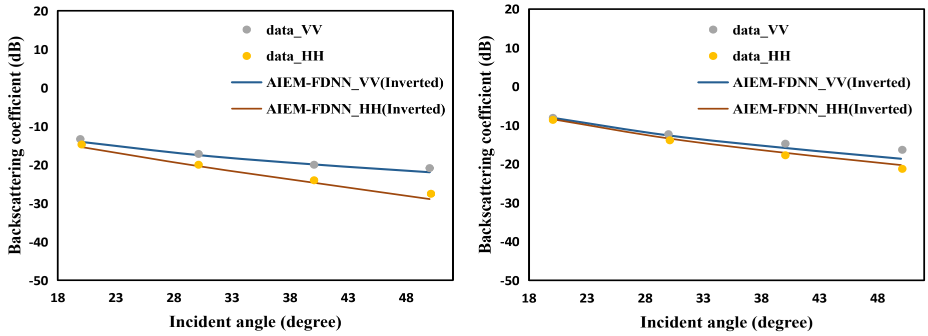
Surface Number	**POLARSCAT (Measured)**				**AIEM-BDNN (Inverted)**			
	ε_r	ε_r''	$k\sigma$	kl	ε_r	ε_r''	$k\sigma$	kl
S1-dry	7.99	2.02	0.13	2.6	9.07	1.23	0.13	2.81
	8.77	1.04	0.40	8.4	9.33	1.19	0.40	8.49
S1-wet	15.57	3.71	0.13	2.6	15.19	4.09	0.13	2.79
	15.42	2.15	0.40	8.4	16.00	0.36	0.40	8.44
S2-dry	5.85	1.46	0.10	3.1	3.02	2.96	0.16	1.29
	6.66	0.68	0.32	9.8	3.23	0.95	0.36	1.00
S2-wet	14.43	3.47	0.10	3.1	10.58	5.43	0.10	3.09
	14.47	1.99	0.32	9.8	14.91	1.63	0.32	9.88
S3-dry	7.7	1.95	0.35	2.6	7.41	2.53	0.31	1.89
	8.5	1.00	1.11	8.4	9.34	0.42	0.99	6.66
S3-wet	15.34	3.66	0.35	2.6	20.79	4.49	0.32	1.04
	15.23	2.12	1.11	8.4	15.00	4.58	0.99	8.86
		ε_r		ε_r''		$k\sigma$		kl
RMSE		2.36		1.21		0.055		2.69
nRMSE		0.1328		0.2386		0.0617		0.3029

As shown in Figures 6–11 the inverted surface parameters are brought into the AIEM-FDNN. The obtained backscattering coefficients are compared with the measured values. It can be seen that the two have a good consistency.



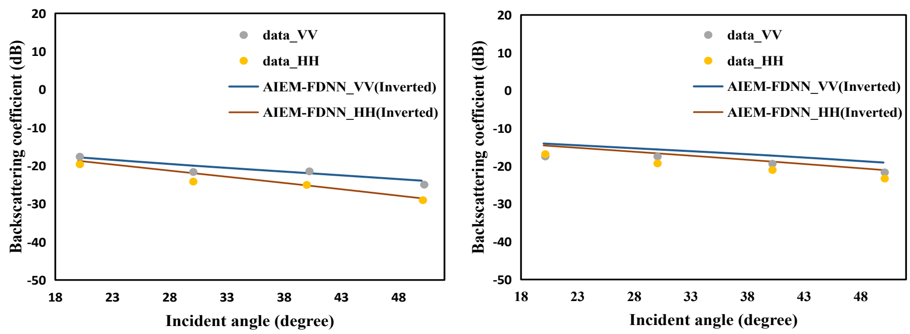
(**a**) (**b**)

Figure 6. Comparison of the backscattering coefficients of the AIEM, POLARSCAT measured data and AIEM−FDNN for exponential correlated surfaces with (**a**) measured: ε_r = 7.99, ε_r'' = 2.02, $k\sigma$ = 0.13 and kl = 2.6 at 1.5 GHz; inverted: ε_r = 9.07, ε_r'' = 1.23, $k\sigma$ = 0.13 and kl = 2.81; (**b**) measured: ε_r = 8.77, ε_r'' = 1.04, $k\sigma$ = 0.4 and kl = 8.4 at 4.75 GHz; inverted: ε_r = 9.33, ε_r'' = 1.19, $k\sigma$ = 0.40 and kl = 8.49.



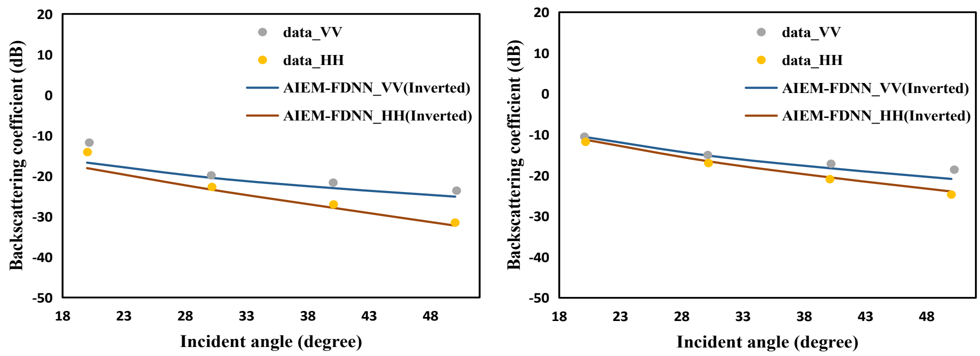
(**a**) (**b**)

Figure 7. Comparison of the backscattering coefficients of the AIEM, POLARSCAT measured data and AIEM−FDNN for the exponential correlated surface with (**a**) measured: ε_r = 15.57, ε_r'' = 3.71, $k\sigma$ = 0.13, and kl = 2.6 at 1.5 GHz; inverted:ε_r = 15.19, ε_r'' = 4.09, $k\sigma$ = 0.13 and kl = 2.79; (**b**) measured: ε_r = 15.42, ε_r'' = 2.15, $k\sigma$ = 0.40 and kl = 8.4 at 4.75 GHz; inverted: ε_r = 16.00, ε_r'' = 0.36, $k\sigma$ = 0.40 and kl = 8.44.



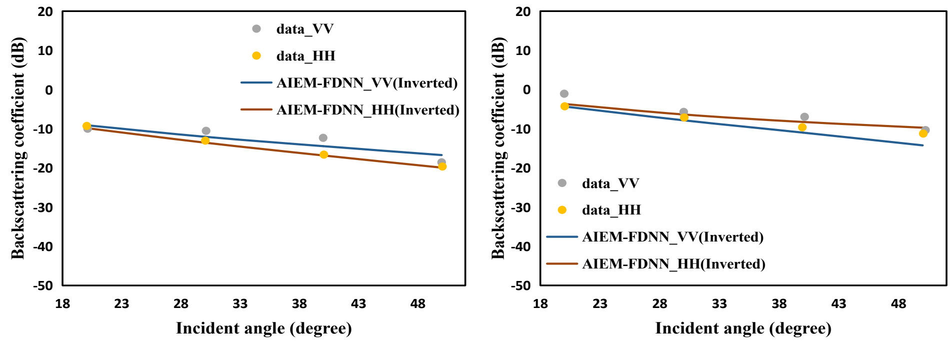
(**a**) (**b**)

Figure 8. Comparison of backscattering coefficients of the AIEM, POLARSCAT measured data and AIEM−FDNN for the exponential correlated surface with (**a**) measured: ε_r = 5.85, ε_r'' = 1.46, $k\sigma$ = 0.10 and kl = 3.1 at 1.5 GHz; inverted: ε_r = 3.02, ε_r'' = 2.96, $k\sigma$ = 0.16 and kl = 1.29; (**b**) measured: ε_r = 6.66, ε_r'' = 0.68, $k\sigma$ = 0.32 and kl = 9.8 at 4.75 GHz; inverted: ε_r = 3.23, ε_r'' = 0.95, $k\sigma$ = 0.36 and kl = 1.00.



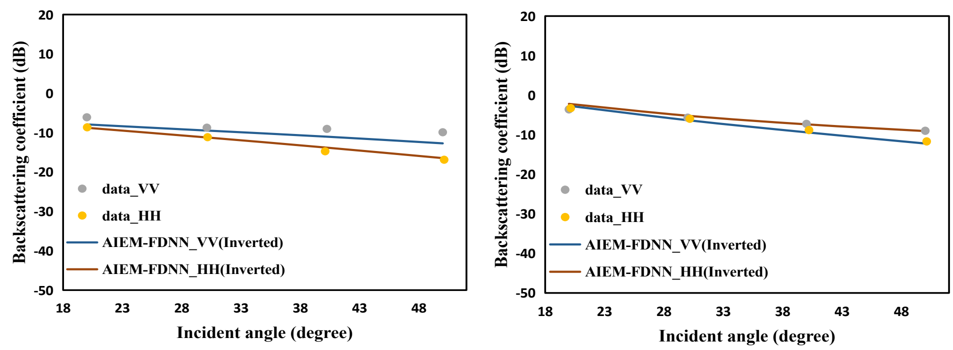
(**a**) (**b**)

Figure 9. Comparison of the backscattering coefficients of the AIEM, POLARSCAT measured data and AIEM−FDNN for the exponential correlated surface with (**a**) measured: ε_r = 14.43, ε_r'' = 3.47, $k\sigma$ = 0.10 and kl = 3.1 at 1.5 GHz; inverted: ε_r = 10.58, ε_r'' = 5.43, $k\sigma$ = 0.10 and kl = 3.09; (**b**) measured: ε_r = 14.47, ε_r'' = 1.99, $k\sigma$ = 0.32 and kl = 9.8 at 4.75 GHz; inverted: ε_r = 14.91, ε_r'' = 1.63, $k\sigma$ = 0.32 and kl = 9.88.



(**a**) (**b**)

Figure 10. Comparison of the backscattering coefficients of the AIEM, POLARSCAT measured data and AIEM−FDNN for the exponential correlated surface with (**a**) measured: ε_r = 7.77, ε_r'' = 1.95, $k\sigma$ = 0.35 and kl = 2.6 at 1.5 GHz; inverted: ε_r = 7.41, ε_r'' = 2.53, $k\sigma$ = 0.31 and kl = 1.89; (**b**) measured: ε_r = 8.5, ε_r'' = 1.00, $k\sigma$ = 1.11 and kl = 8.4 at 4.75 GHz; inverted: ε_r = 9.34 ε_r'' = 0.42, $k\sigma$ = 0.99 and kl = 6.66.



(**a**) (**b**)

Figure 11. Comparison of the backscattering coefficients of the AIEM, POLARSCAT measured data and AIEM−FDNN for the exponential correlated surface with (**a**) measured: ε_r = 15.34, ε_r'' = 3.66, $k\sigma$ = 0.35 and kl = 2.6 at 1.5 GHz; inverted: ε_r = 20.79, ε_r'' = 4.49, $k\sigma$ = 0.32 and kl = 1.04; (**b**) measured: ε_r = 15.23, ε_r'' = 2.12, $k\sigma$ = 1.11 and kl = 8.4 at 4.75 GHz; inverted: ε_r = 15.00, ε_r'' = 4.58, $k\sigma$ = 0.99 and kl = 8.86.

4. Discussion

In this paper, the bi-directional network performs well in the task of surface parameter inversion. The bi-directional network has a high inversion accuracy for the AIEM model dataset. Similarly, for the inversion of the POLARSCAT measured data by the bi-directional network, the inversion value has a good correlation with the real value.

The bi-directional network is proposed to solve the problem of non-uniqueness, which leads to the poor effect of direct training of the inverse network. The nonunique data in the dataset itself will cause the training error of the directly constructed inverse network (with the backscattering coefficients as the input and the surface parameters as the output) to be unable to decrease and converge well. To solve this problem, bi-directional networks are proposed. The forward network AIEM-FDNN (with the surface parameters as the input and the backscattering coefficients as the output) is first trained, and the inverse network is constructed by reusing the weights trained by the AIEM-FDNN. In this way, the problem of directly constructing the inverse network can be avoided, and the bi-directional network achieves better inversion accuracy.

A BP (back propagation) neural network with backscattering coefficients as the input and surface parameters as the output is directly constructed. The 21,009 datasets generated by the AIEM model are used for training, and the training loss curve is shown in Figure 12a. Note that the training stops when the validation loss does not drop for 40 consecutive epochs. It can be seen that the training and validation losses for the BP neural network are 1.6257 and 1.5519, respectively, and the loss value barely dropped. This shows that the directly built inverse network performs poorly for the task of inverting surface parameters from input backscattering coefficients. The biggest reason that the inverse network cannot be trained well is the most common non-uniqueness problem in the inverse task of the neural network. Since the combination of different surface parameters can obtain the same or similar backscattering coefficients, this leads to a one-to-many situation during inverse network training. Once there are too many nonunique data in the dataset, the training loss of the network cannot be reduced well. On the contrary, the training of the forward network with the surface parameters as the input and the backscattering coefficient as the output does not have the influence of nonunique data on it. Therefore, it is hoped to start from the forward network and design a new method of surface parameter inversion. A bi-directional network was designed to overcome the above problems.

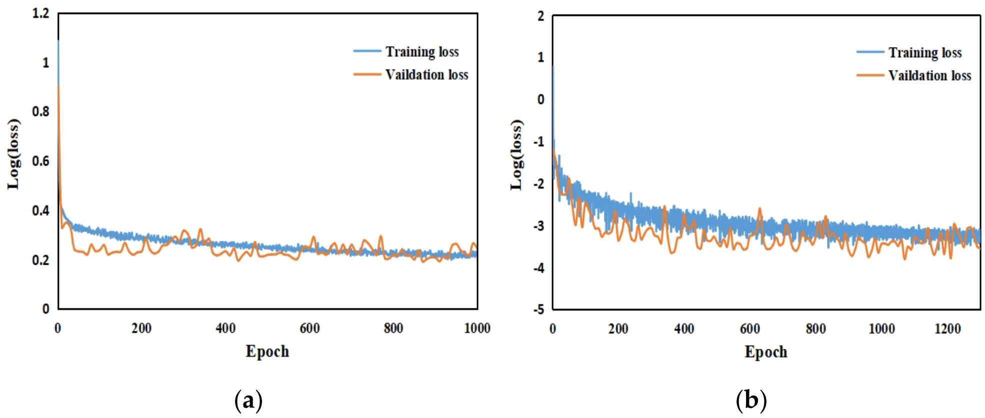


Figure 12. (**a**) Learning curve of the BP neural network. (**b**) Learning curve of the AIEM−FDNN.

As shown in Figure 12b, the loss curve of training and validation converges to a small value and keeps fluctuating after the AIEM-FDNN trained for 1300 epochs. Finally, the training loss value and validation loss value of the network are 6.45×10^{-4} and 1.17×10^{-4}, respectively. This loss value of the proposed bi-directional DNN is smaller than the traditional inverse network by several magnitudes. The weights trained by the AIEM-FDNN can be directly reused by the AIEM-BDNN, which can show a better loss convergence. As shown in Table 8, the bi-directional network achieves a better inversion accuracy.

Table 8. Inversion accuracy of BP neural networks and Bi-directional DNN.

	Bi-Directional DNN		BP	
Parameter	RMSE	Similarity (1-RMSE)	RMSE	Similarity (1-RMSE)
ε_r	0.0244	97.56%	0.0528	94.72%
ε_r''	0.0886	91.14%	0.4948	50.52%
$k\sigma$	0.0096	99.04%	0.0457	95.43%
kl	0.0155	98.45%	0.0374	96.26%

5. Conclusions

In this paper, a novel bi-directional neural network was proposed to invert the surface parameters. The establishment of the bi-directional network is divided into two steps. The AIEM-FDNN established first takes the surface parameters as the input and the backscattering coefficients as the output. The trained AIEM-FDNN can predict the backscattering coefficients outside the training dataset, and the predictions are also very accurate for the measured data. The AIEM-BDNN is built by reusing weights and biases trained by the AIEM-FDNN. At the same time, it is necessary to give the input surface parameter initialization constants, and a sigmoid layer between the input layer and the first hidden layer is inserted. After the error between the output backscattering coefficients and the true backscattering coefficients is continuously reduced, the input surface parameters can be continuously updated. The numerical results show that the bi-directional network not only has a good inversion effect for the data in the dataset but also has a high inversion accuracy for the measured data outside the dataset.

The bi-directional network is divided into a forward network (AIEM-FDNN) and an inverse network (AIEM-BDNN). The AIEM-BDNN is constructed by reusing the weights and biases of the AIEM-FDNN and does not require secondary training. Therefore, the training accuracy of the AIEM-FDNN will directly determine the inversion accuracy of the AIEM-BDNN. If the training effect of the forward network on some datasets is not good, then the bi-directional network will not be able to achieve a good inversion result.

One limitation we had to deal with in this paper is that the datasets used were only for backscattering coefficients under HH and VV polarizations. As a future work direction, we plan to incorporate the backscattering coefficients under HV and VH polarizations. The more abundant features of the four polarizations were used to further improve the accuracy of the surface parameters inversion. In addition, we considered adding part of the measured data to the dataset generated by the AIEM for training. We hope to reduce some of the differences between the simulated and measured data.

Author Contributions: Conceptualization, Y.W. and Z.H.; methodology, Y.W. and Z.H.; software, Y.W.; validation, Y.W., Z.H. and Y.Y.; formal analysis, Y.W., Z.H., Y.Y., D.D., F.D. and X.-W.D.; investigation, Y.W.; resources, Y.W. and Z.H.; data curation, Y.W.; writing—original draft preparation, Y.W.; writing—review and editing, Y.W., Z.H., Y.Y., D.D., F.D. and X.-W.D.; visualization, Y.W.; supervision, Z.H.; project administration, Z.H. and funding acquisition, Z.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by Natural Science 62071231, 61890541 and 61931021; Jiangsu Province Natural Science Foundation under Grant BK20211571 and the Fundamental Research Funds for the Central Universities of No. 30921011207, Laboratory of Pinghu (Beijing Institute of infinite electric Measurement), Science and Technology on Electromagnetic Scattering Laboratory.

Data Availability Statement: No applicable.

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

Appendix A

Algorithm A1: Bi-directional Deep Neural Network

Input: The input surface parameters $\mathbf{Z}^0_{\mathbf{AF}}$ and $\mathbf{Z}^0_{AB}$, the true backscattering coefficients $[\sigma^L_{HH}, \sigma^L_{VV}]$, the maximum epoch I, the weights matrix $\mathbf{W}^{\mathbf{i}}_{\mathbf{AF}}$ and $\mathbf{W}^{\mathbf{i}}_{AB}$, the bias vector $\mathbf{b}^{\mathbf{i}}_{\mathbf{AF}}$ and $\mathbf{b}^{\mathbf{i}}_{AB}$, the nonlinear activation function $g_i(\cdot)$, the loss function MSE, the learning rate η_{AF}, η_{AB}

initialize $\mathbf{W}^{\mathbf{i}}_{\mathbf{AF}}$ and $\mathbf{b}^{\mathbf{i}}_{\mathbf{AF}}$
for $j = 1; j \leq I$ **do**
  $\mathbf{Z}^{\mathbf{i}}_{\mathbf{AF}} = g_i\left(\mathbf{W}^{\mathbf{i}}_{\mathbf{AF}} \cdot \mathbf{Z}^{\mathbf{i\text{-}1}}_{\mathbf{AF}} + \mathbf{b}^{\mathbf{i}}_{\mathbf{AF}}\right)(i = 1, \cdots, N)$
  $\mathbf{Z}^N_{\mathbf{AF}} = g_N\left(\mathbf{W}^N_{\mathbf{AF}} \cdot \mathbf{Z}^{N-1}_{\mathbf{AF}} + \mathbf{b}^N_{\mathbf{AF}}\right)$
  $Loss_{AF} = MSE(\mathbf{Z}^N_{\mathbf{AF}}, [\sigma^L_{HH}, \sigma^L_{VV}])$
  $\mathbf{W}^{\mathbf{i}}_{\mathbf{AF}} = \mathbf{W}^{\mathbf{i}}_{\mathbf{AF}} - \eta_{AF}\frac{\partial Loss_{AF}}{\partial \mathbf{W}^{\mathbf{i}}_{\mathbf{AF}}}, \mathbf{b}^{\mathbf{i}}_{\mathbf{AF}} = \mathbf{b}^{\mathbf{i}}_{\mathbf{AF}} - \eta_{AF}\frac{\partial Loss_{AF}}{\partial \mathbf{b}^{\mathbf{i}}_{\mathbf{AF}}}$
  if $Loss_{AF}$ convergence **then**
    break loop
  end if
  $j = j + 1$
end for
return $\mathbf{W}^{\mathbf{i}}_{\mathbf{AF}}$ and $\mathbf{b}^{\mathbf{i}}_{\mathbf{AF}}$
Initialize $\mathbf{Z}^0_{\mathbf{A}B}$
for $k = 1; k \leq I$ **do**
  $\mathbf{Z}^{\mathbf{i}}_{\mathbf{AB}} = g_i\left(\mathbf{W}^{\mathbf{i}}_{\mathbf{AF}} \cdot \mathbf{Z}^{\mathbf{i}-1}_{\mathbf{AB}} + \mathbf{b}^{\mathbf{i}}_{\mathbf{AF}}\right)(i = 1, \cdots, N)$
  $16: \mathbf{Z}^N_{\mathbf{AB}} = g_N\left(\mathbf{W}^N_{\mathbf{AF}} \cdot \mathbf{Z}^{N\text{-}1}_{\mathbf{AB}} + \mathbf{b}^N_{\mathbf{AF}}\right)$
  $Loss_{AB} = MSE(\mathbf{Z}^N_{\mathbf{AB}}, [\sigma^L_{HH}, \sigma^L_{VV}])$
  $\mathbf{Z}^0_{\mathbf{AB}} = \mathbf{Z}^0_{\mathbf{AB}} - \eta_{AB}\frac{\partial Loss_{AB}}{\partial \mathbf{Z}^0_{\mathbf{AB}}}$
  if $Loss_{AB}$ convergence **then**
    break loop
  end if
  $k = k + 1$
end for
return $\mathbf{Z}^0_{\mathbf{A}B}$

Output: Inversion results $\mathbf{Z}^0_{\mathbf{A}B}$

References

1. Mohammad, H.M.; Amir, A.; Hamid, S.S. Substitution of satellite-based land surface temperature defective data using GSP method. *Adv. Space Res.* **2021**, *67*, 3106–3124.
2. Kim, Y.; Jackson, T.; Bindlish, R.; Lee, H.; Hong, S. Monitoring soybean growth using L-, C- and X-band scatterometer data. *Int. J. Remote Sens.* **2013**, *34*, 4069–4082. [CrossRef]
3. Yang, H.; Guo, H.D.; Wang, C.L.; Li, X.W.; Yue, H.Y. Polarimetric SAR surface parameters inversion based on network. *J. Remote Sens.* **2002**, *6*, 451–455.
4. Shen, X.; Mao, K.; Qin, Q.; Hong, Y.; Zhang, G. Bare surface soil moisture estimation using double-angle and dual-polarization L-band radar data. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 3931–3942. [CrossRef]
5. Chiang, C.Y.; Chen, K.S.; Yang, Y.; Wang, S.Y. Computation of backscattered fields in polarimetric SAR imaging simulation of complex targets. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 2004113. [CrossRef]
6. Sancer, M. Modified Beckmann-Kirchhoff scattering model for rough surface with large incident and scattering angles. *Opt. Eng.* **2007**, *46*, 078002.
7. Thorsos, E.I. The validity of the perturbation approximation for rough surface scattering using a Gaussian roughness spectrum. *Acoust. Soc. Am.* **1989**, *86*, 261–277. [CrossRef]
8. Soto-Crespo, J.M.; VesPerinas, M.N.; Friberg, A.T. Scattering from slightly rough random surfaces: A detailed study on the validity of the small perturbation method. *J. Opt. Soc. Am. A* **1990**, *7*, 1185–12017. [CrossRef]
9. Gilbert, M.S.; Johnson, M.S. A study of the higher-order small-slope approximation for scattering from a Gaussian rough surface. *Waves Random Media* **2003**, *13*, 137–149. [CrossRef]
10. Berginc, G.; Bourrely, C. The small-slope approximation method applied to a three-dimensional slab with rough boundaries. *Prog. Electromagn. Res.* **2007**, *73*, 131–211. [CrossRef]
11. Xu, F.; Jin, Y.Q. Imaging simulation of po-larimetric SAR for a comprehensive terrain scene using the mapping and projection algorithm. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 3219–3234. [CrossRef]

12. Zeng, J.Y.; Chen, K.S.; Bi, H.Y.; Zhao, T.J.; Yang, X.F. A comprehensive analysis of rough soil surface scattering and emission predicted by AIEM with comparison to numerical simulations and experimental measurements. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 1696–1708. [CrossRef]
13. Chen, K.S.; Wu, T.D.; Tsang, L.; Li, Q.; Shi, J.; Fung, A.K. Emission of rough surfaces calculated by the integral equation method with comparison to three-dimensional moment method simulations. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 90–101. [CrossRef]
14. Ulaby, F.T.; Sarabandi, K.; Mcdonald, K.Y.L.E.; Whitt, M.; Dobson, M.C. Michigan microwave canopy scattering model. *Int. J. Remote Sens.* **1990**, *11*, 1223–1253. [CrossRef]
15. Dubois, P.; Van Zyl, J.; Engman, T. Measuring soil moisture with imaging radars. *IEEE Trans. Geosci. Remote Sens.* **1995**, *33*, 915–926. [CrossRef]
16. Oh, Y. Quantitative retrieval of soil moisture content and surface roughness from multipolarized radar observations of bere soil surfaces. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 596–601. [CrossRef]
17. Zhao, S.T. Inverse calculation of hydrogeological parameters in Henan based on improved genetic algorithm. *Ground Water* **2019**, *41*, 77–79.
18. Wang, L.X.; Wang, A.Q.; Huan, Z.X. Parameter inversion of rough surface optimization based on multiple algorithms for SVM. *Chin. J. Comput. Phys.* **2019**, *36*, 577–585.
19. Peurifoy, J.; Shen, Y.; Jing, L.; Yang, Y.; Cano-Renteria, F.; DeLacy, B.G.; Joannopoulos, J.D.; Tegmark, M.; Soljačić, M. Nanophotonic particle simulation and inverse design using artificial neural networks. *Sci. Adv.* **2018**, *4*, eaar4206. [CrossRef]
20. Li, G.; Li, X.Z.; Liu, D.J.; Wang, L.H.; Yu, Z.F. A bidirectional deep neural network for accurate silicon color design. *Adv. Mater.* **2019**, *31*, 1905467.
21. Xu, F.; Wang, H.P.; Jin, Y.Q. Deep learning as applied in SAR target recognition and terrain classification. *J. Radars* **2017**, *6*, 136–148.
22. Sharifzadeh, F.; Akbarizadeh, G.; Kavian, Y.S. Ship classification in SAR images using a new hybrid CNN-MLP classifier. *J. Indian Soc. Remote Sens.* **2019**, *47*, 551–562. [CrossRef]
23. Ding, J.; Chen, B.; Liu, H.W.; Huang, M.Y. Convolutional Neural Network with Data Augmentation for SAR Target Recognition. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 364–368. [CrossRef]
24. Niu, S.R.; Qiu, X.L.; Lei, B.; Ding, C.B.; Fu, K. Parameter extraction based on deep neural network for SAR target simulation. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4901–4914. [CrossRef]
25. Oh, Y.; Sarabandi, K.; Ulaby, F.T. An empirical model and inversion technique for radar scattering from bare soil surfaces. *IEEE Trans. Geosci. Remote Sens.* **1992**, *30*, 370–381. [CrossRef]
26. Yang, Y.; Chen, K.S.; Shang, G.F. Surface parameters retrieval from fully bistatic radar scattering data. *Remote Sens.* **2019**, *11*, 596. [CrossRef]
27. Chen, K.S. *Radar Scattering and Imaging of Rough Surfaces*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2021; pp. 160–163.
28. Yang, Y.; Chen, K.S.; Tsang, L.; Yu, L. Depolarized backscattering of rough surface by AIEM model. *IEEE J. Sel. Top. Appl. Earth Sci. Remote Sens.* **2017**, *10*, 4740–4752. [CrossRef]
29. Zhang, Y.Y.; Wu, Z.S.; Zhang, Y.S. The effective permittivity and roughness parameters inversion by the land backscattering measured data. *Chin. J. Radio Sci.* **2016**, *31*, 79–84.
30. So, S.; Badloe, T.; Noh, J.; Bravo-Abad, J.; Rho, J. Deep learning enable inverse design in nanophotonics. *Nanophotonics* **2020**, *9*, 1041–1057. [CrossRef]
31. Li, J.; Li, X.Z.; Wu, Q.X.; Wang, L.H.; Li, G. Neural network enabled metasurface design for phase manipulation. *Opt. Express* **2021**, *29*, 2521–2528.