

Research Article

Accurate Identification of Agricultural Inputs Based on Sensor Monitoring Platform and SSDA-HELM-SOFTMAX Model

Juan Zou,^{1,2} Hanjing Jiang,^{1,2} Qingxiu Wang,^{1,2} Ningxia Chen,^{1,2} Ting Wu ^{1,2} and Ling Yang ^{1,2,3}

¹Guangdong Provincial Food Safety Traceability and Control Engineering Technology Research Center, China

²School of Information Science and Technology, Zhongkai University of Agriculture and Engineering, Guangzhou, Guangdong 510225, China

³Agricultural and Rural Big Data Common Key Technology Innovation Team, Guangzhou, Guangdong 510225, China

Correspondence should be addressed to Ting Wu; 405684932@qq.com and Ling Yang; yl98613@163.com

Received 6 May 2021; Accepted 2 November 2021; Published 24 November 2021

Academic Editor: Yuan Li

Copyright © 2021 Juan Zou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The unreliability of traceability information on agricultural inputs has become one of the main factors hindering the development of traceability systems. At present, the major detection techniques of agricultural inputs were residue chemical detection at the postproduction stage. In this paper, a new detection method based on sensors and artificial intelligence algorithm was proposed in the detection of the commonly agricultural inputs in *Agastache rugosa* cultivation. An agricultural input monitoring platform including software system and hardware circuit was designed and built. A model called stacked sparse denoising autoencoder-hierarchical extreme learning machine-softmax (SSDA-HELM-SOFTMAX) was put forward to achieve accurate and real-time prediction of agricultural input varieties. The experiments showed that the combination of sensors and discriminant model could accurately classify different agricultural inputs. The accuracy of SSDA-HELM-SOFTMAX reached 97.08%, which was 4.08%, 1.78%, and 1.58% higher than a traditional BP neural network, DBN-SOFTMAX, and SAE-SOFTMAX models, respectively. Therefore, the method proposed in this paper was proved to be effective, accurate, and feasible and will provide a new online detection way of agricultural inputs.

1. Introduction

In recent years, agricultural product traceability systems have been gradually applied to the actual production process, but manually entered traceability information is difficult to gain the trust of consumers and regulators, and a lack of trust in traceability information has become one of the main factors hindering the uptake of traceability systems. Three main factors affect the quality and safety of agricultural products: air pollution, soil pollution, and agricultural input pollution [1]. Among them, agricultural inputs refer to products permitted for use in organic farming, including feedstuffs, fertilizers, and permitted plant protection products as well as cleaning agents and additives used in food production. To prevent air pollution, traceability systems can automatically collect and save environmental data. To

prevent soil pollution, traceability systems can record and save soil test reports. To prevent agricultural input pollution such as fertilizers and pesticides used in the production process, traceability systems are currently mainly used to record agricultural residue testing reports. However, the traditional chemical and biological detection methods are unable to cope with a large number of real-time online tests due to many problems such as sample preparation requirements, complicated operating processes, extended experiment durations, and sample destruction. In recent years, the rapid development of deep learning methods has directly promoted the in-depth application of artificial intelligence technology in the agricultural environment and other fields, especially for prediction and early warning based on the combination of real-time and prior information [2, 3]. Therefore, research on real-time online prediction of

agricultural inputs based on deep learning is highly significant, which can improve the accuracy of input prediction and ensure the timeliness and accuracy of the traceability information.

In recent years, some researchers have studied some techniques to predict agricultural inputs. For example, Chough et al., Kumaran and Tran-Minh, and others used electrochemical or biotechnology to quickly detect pesticides and achieved good results [4–6]. Galceran et al. used pyridine chloride hydrochloride as an electrolyte to identify several seasonal herbicides without UV absorption [7]. Andrade et al. established a liquid chromatography-electrospray tandem mass spectrometry method and used agents to neutralize the matrices and in turn to produce better recovery and faster detection in tomato samples [8]. Shamsipur et al. reported a method involving the DLLME method coupled with SPE for the identification of pesticides in fruit juice, water, milk, and honey samples [9]. Some researchers also used sensors to monitor agricultural inputs. For example, Datir and Wagh used a wireless sensor network to monitor and detect downy mildew in grapes, realizing a real-time system for detecting agricultural diseases based on weather data [10]. Zhu et al. used some nanozyme sensor arrays to detect pesticides [11]. Yi et al. used a photoluminescence sensor for ultrasensitive detection of pesticides [12]. However, these methods were residual detection after implementation, and input information needed to be recorded manually, which could not guarantee the real-time and accuracy of the traceability system.

In recent years, with the rapid development of technologies such as artificial intelligence and sensors, extreme learning machines (ELMs) have become an important part of machine learning and have excellent generalization performance, fast learning, and are less likely to become trapped in local optimums [13]. The method has been successfully applied in load forecasting [14] and fault diagnosis [15, 16]. However, the complex and changeable crop planting environment has produced many interferences which influence the physicochemical parameters of agricultural inputs and create nonlinear variation. Thus, there were two problems with using an ELM neural network to classify and predict agricultural inputs [17]. Firstly, the input weights and hidden layer deviation of the ELM neural network during the modeling process were generated randomly, so the classification performance was reduced. Secondly, the random initial parameters may also make the number of ELM hidden layer nodes more than the traditional parameter adjustment neural network, increasing test time. Therefore, the key factor to improve the performance of ELM neural networks is efficient pretraining parameters.

In order to solve the above problems, we have studied an algorithm of using deep learning to monitor agricultural inputs and achieved very good results [18]. Based on the previous research, this paper has made improvements in the algorithm model, experimental design, software architecture, hardware design, and preprocessing methods and achieved better prediction results than the previous research. This paper took eight kinds of agricultural inputs commonly used for *Agastache rugosa* as research objects (ammonium sulfate,

potassium fertilizer, phosphate fertilizer, Bordeaux mixture, chlortetracycline, imidacloprid, pendimethalin, and bromoxynil). A monitoring platform including software system and hardware circuit, which could realize sensor data collection, wireless transmission, and storage, was built. In algorithm research, greedy layer-wise training and fine-tuning of the stacked autoencoder were used to initialize the parameters, then removed the decoding part of the stacked sparse denoising autoencoder (SSDA) model, and connect with the hierarchical extreme learning machine (HELM) neural network. Finally, an agricultural input classification prediction model based on SSDA-HELM-SOFTMAX was established, which laid the foundation for accurate classification prediction of agricultural inputs.

2. SSDA-HELM-SOFTMAX Algorithm Description

2.1. Stacked Autoencoder. An autoencoder [19] is an unsupervised neural network model based on deep learning that can reconstruct the original input data into an approximate new data, expressing any data in low dimensions through the symmetric structure and weight coefficients of the network, and at its core is the ability to learn the deep representation of input data. The drawback is that the parameters of neurons in the network will continue to increase in the number of hidden layers, which affects the calculation speed of the network. One of its main applications is to get the weight parameters of the initial neural network by layer-wise pretraining and fine-tuning, with better results than the traditional symmetric random parameters.

Multiple autoencoders were stacked to form a stacked autoencoder [20, 21] whose main function is to extract deep characteristics and nonlinear dimension reduction. A stacked autoencoder combined with a supervised classifier can accomplish multicategory classification. Each autoencoder in the structure of the stacked autoencoder (Figure 1) performed encoding and decoding operations and feature extraction from the output of the previous autoencoder. The output of the autoencoder is a reconstruction or approximation of its input, but it cannot be used to directly classify the input information without a supervised classifier. The three autoencoders in Figure 1 can obtain three hidden layers through feature extraction, and a supervised classifier can be added to the output layer to realize classification prediction.

2.2. Sparse Autoencoder (SAE). There were three layers in the autoencoder, namely, the input layer, hidden layer, and output layer. The number of nodes on each layer was n , d , and n , respectively (Figure 2). Autoencoder attempts to approximate an identity function which causes the output data to approximate the input data, and the hidden layer activation value was $a_2 = (a_{21}, a_{22}, \dots, a_{2n})$, which were the features of the input vector.

The formula used in the encoding process of stacked autoencoder was as follows:

$$h = f_{\theta_1}(x) = \sigma(W_1x + b_1). \quad (1)$$

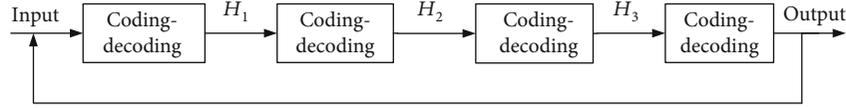


FIGURE 1: The structure of the stacked autoencoder.

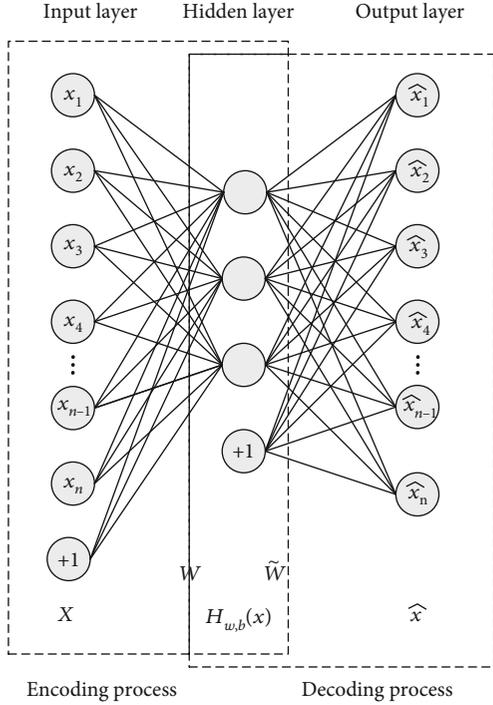


FIGURE 2: The structure of the AE.

The formula used in the decoding was as follows:

$$\hat{x} = f_{\theta_2}(h) = \sigma(W_2 h + b_2), \quad (2)$$

where W_1 and W_2 were the weight matrix of the input layer to the hidden layer and the hidden layer to the output layer, b_1 and b_2 were the unit bias coefficients of the hidden layer and the output layer, $\sigma(\bullet)$ was the logsig function, and θ was the network parameter matrix.

The goal of the autoencoder is to find the optimal parameter matrix and minimize the error between the input and the output. The reconstructed error loss function was expressed as follows:

$$J_E(W, b) = \text{loss} + R = \frac{1}{m} \sum_{i=1}^m \left(\frac{1}{2} \left\| x^{\wedge(i)} - h_{j_i, b}(x^{(i)}) \right\| \right) + \frac{\lambda}{2} \sum_{l=1}^{n_1} \sum_{i=1}^{s_1} \sum_{j=1}^{s_{l-1}} \left(W_{ji}^{(l)} \right)^2, \quad (3)$$

where loss was the loss function, R was the weight attenuation term which could effectively prevent overfitting, m was the number of samples. $x^{(i)}$ and $x^{\wedge(i)}$ were the input and output characteristics of the i_{th} sample, n_l was the num-

ber of network layers, s_l was the unit number of the j_{th} layer, and λ was the weight attenuation coefficient.

An SAE was used to add a sparsity restriction to the hidden layer of the autoencoder. The sparsity restriction was to control the number of network parameters by suppressing the activation of network neurons and to achieve more effective extraction of data features.

$a_j^2(x)$ was the activate degree of hidden layer neurons, and the mean degree of activation can be expressed as follows:

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^m \left(a_j^2(x^{(i)}) \right). \quad (4)$$

It was considered active when the neuron output was close to 1 and inactive when the output was close to 0. Therefore, by adding ρ , a sparsity parameter whose value approaches 0, and making $\hat{\rho} = \rho$, most neurons can be inhibited. To realize the sparsity limitation, the sparse penalty term was added to the cost function, and the total cost function was as follows:

$$J(W, b) = \frac{1}{m} \sum_{i=1}^m \left(\frac{1}{2} \left\| x^{\wedge(i)} - h_{j_i, b}(x^{(i)}) \right\| \right) + \frac{\lambda}{2} \sum_{l=1}^{n_1} \sum_{i=1}^{s_1} \sum_{j=1}^{s_{l-1}} \left(W_{ji}^{(l)} \right)^2 + \beta \text{KL}(\rho \parallel \hat{\rho}_j), \quad (5)$$

where β was the coefficient of sparse penalty and $\text{KL}(\rho \parallel \hat{\rho}_j)$ was the sparse penalty for the hidden layer neuron j .

$$\text{KL}(\rho \parallel \hat{\rho}_j) = \sum_{j=1}^{s_2} \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j}, \quad (6)$$

where s_2 was the number of neurons in the hidden layer. The sparse penalty got a unique minimum value when $\hat{\rho} = \rho$, minimizing the penalty term could make the mean activate degree of the hidden layer close to the sparse parameter.

2.3. ELM Modeling. If there were N different samples (x_i, t_i) , where $x_i = x_{i1}, x_{i2}, \dots, x_{in}^T \in R^n$, $t_i = t_{i1}, t_{i2}, \dots, t_{im}^T \in R^m$, a neural network with L hidden layer nodes could be expressed as follows:

$$\sum_{i=1}^L \beta_i g_i(X_j) = \sum_{i=1}^L \beta_i g(W_i \bullet X_j + b_i) = o_j (j = 1, \dots, N), \quad (7)$$

where $g(x)$ was the activation function, $W_i = [w_{i,1}, w_{i,2}, \dots, w_{i,n}]^T$ was the weight between the input node

and the i_{th} hidden node, β_i was the weight between the i_{th} hidden node and the output node, b_i was the bias of the i_{th} hidden layer node, and $W_i \cdot X_j$ was the inner product of W_i and X_j .

In order to ensure the most accurate of the output of the neural network model, it was necessary to

$$\sum_{j=1}^N \|o_j - t_j\| = 0. \quad (8)$$

Import Equation (8) to produce the following:

$$\sum_{i=1}^L \beta_i g(W_i \bullet X_j + b_i) = t_j (j = 1, \dots, N). \quad (9)$$

If H was the output of the hidden layer, β was the weight of H , and T was the desired output, then

$$H(W_1, \dots, W_L, b_1, \dots, b_L, X_1, \dots, X_L) = \begin{bmatrix} g(W_1 \bullet X_1 + b_1) & \dots & g(W_L \bullet X_1 + b_L) \\ \dots & \dots & \dots \\ g(W_1 \bullet X_N + b_1) & \dots & g(W_L \bullet X_N + b_L) \end{bmatrix}_{N \times L}, \quad (10)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \dots \\ \beta_N^T \end{bmatrix}_{L \times m}, \quad T = \begin{bmatrix} T_1^T \\ \dots \\ T_N^T \end{bmatrix}_{N \times m}.$$

From Equation (10), $H\beta = T$, $\widehat{W}_i, \widehat{b}_i, \widehat{\beta}_i$ would be calculated in order to be able to train the single-hidden layer neural network.

$$\|H(\bar{W}_1, \dots, \bar{W}_L, \bar{b}_1, \dots, \bar{b}_L)\| = \min_{W_i, b_i, \beta} \|H(\bar{W}_1, \dots, \bar{W}_L, \bar{b}_1, \dots, \bar{b}_L)\beta_i - T\|, \quad (11)$$

where $i = 1, \dots, L$.

The above formula was equivalent to minimizing the following loss function.

$$E = \sum_{j=1}^N \left(\sum_{i=1}^L \beta_i g(W_i \bullet X_j + b_i) - t_j \right)^2. \quad (12)$$

The gradient algorithm was adopted, and the parameters need to be adjusted during the iteration process, but in the ELM algorithm, once the input weight and hidden layer bias were determined randomly, the hidden layer output matrix H could be uniquely determined by the least square solution $\widehat{\beta}$.

$$\widehat{\beta} = H^\dagger T, \quad (13)$$

where H^\dagger was the generalized inverse matrix of H , and there were two conditions for $\widehat{\beta}$, that the norm was the smallest, and that the value of $\widehat{\beta}$ was unique.

The structure of the ELM algorithm is shown in Figure 3.

2.4. SSDA-HELM-SOFTMAX Modeling. The SSDA-HELM-SOFTMAX model took the SAE as the front-end pretrained for the initial weights and provided parameters for the multi-

layer ELM model to reach the optimal solution. Then, SOFTMAX was used for agricultural input identification in the output layer of this model. In the model training process, physical and chemical parameter values of agricultural inputs as collected by the sensor were sent to the SAE input layer as training samples. The SAE hidden layer would extract relevant features from these complex samples and adopt unsupervised learning methods for initial weight. Then, the decoding part of the SAE was removed, connected to the ELM and assigned the initial weight as the initial value of the EML, and used SOFTMAX for classification. The diagram of SSDA-HELM-SOFTMAX model constructed in this paper is shown in Figure 4.

The specific EML algorithm was as follows:

- (1) The number of hidden layers of the network was initialized to k , $X_1 = X$ and $X = [x^{(1)}, x^{(2)}, \dots, x^{(m)}]^T$, the number of hidden layers' nodes was \widehat{N} , and the decoding part of SSDA was deleted and connected to the HELM
- (2) From the first layer, the SSDA-HELM network was initialized using W_i and b_i of the SSDA network training as input weights
- (3) The input weight W_i and hidden layer bias b_i generated by pretraining were used to calculate the hidden layer output matrix: $A_i = H_{i-1} W_{li}$
- (4) According to ELM theory, the output weight matrix of neural network was calculated: $\widehat{\beta} = A^\dagger T$
- (5) The output calculation was as follows: $\widehat{H}_i = g(H_{i-1} \cdot \widehat{\beta})$, where \widehat{H}_i and H_{i-1} were the output and input

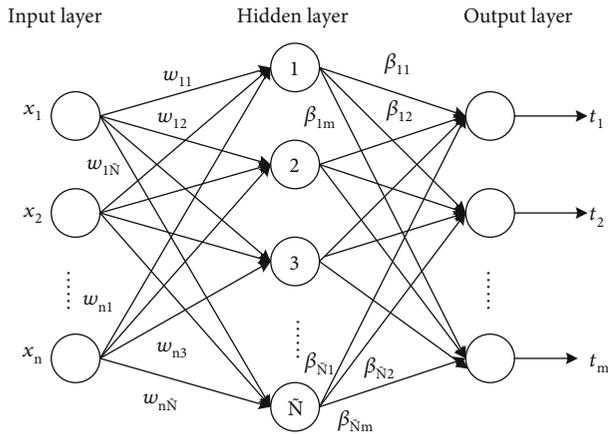


FIGURE 3: The structure of the ELM.

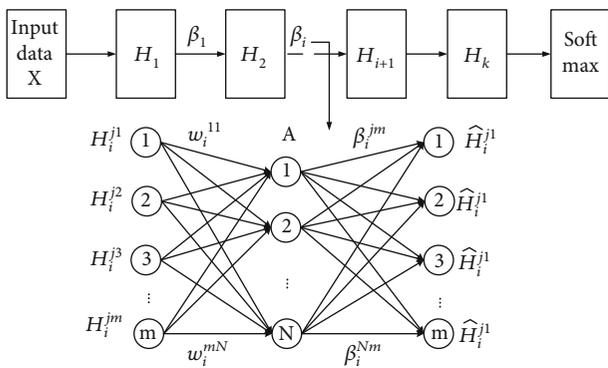


FIGURE 4: SSDA-HELM-SOFTMAX model.

of layer i , respectively, and $g(\cdot)$ was the activation function of the hidden layer

This was then repeated from step 2 for the next layer.

- (6) The extracted features were used as input values and sent to the SOFTMAX classifier for classification prediction

3. Monitoring Platform Design

3.1. Software Architecture. The software architecture of the agricultural input monitoring system is shown in Figure 5. It was mainly composed of four modules: data collection service, data mutation detection service, external display plan task, and server side. The functions of each part were as follows:

- (1) *Data Collection Service.* Through the TCP protocol and multithreaded programming, the acquisition program was designed. The sensors were queried every 15 seconds to obtain real-time data, such as pH value, EC value, temperature, and humidity, and stored in the database.
- (2) *Data Mutation Detection Service.* The sensor data stored in the database was detected in real time. When the sensor data changed more than 20% for

three consecutive times, it is considered that the data has a sudden change, and the prediction module was called to predict the type of agricultural inputs.

- (3) *External Display Plan Task Service.* According to the needs of the display page, the module was designed to calculate the relevant data. For example, the module calculated the original data at regular intervals to calculate the hourly average, daily average, and monthly average.
- (4) *Server-Side Architecture.* At the presentation layer, AJAX was used to request server-side data, the handlebar engine was used for page rendering, and graphs and tables were used for data visualization. In the business layer, logical business was designed according to application requirements. WCF framework and general processing program ashx were used to provide web interface services for the visualization layer. In the data layer, the ORM framework was designed and implemented through the bottom layer of the database, and the ADO.NET class library was used to complete the database access and provide data services for the business layer.

3.2. Hardware Design. The hardware structure is shown in Figure 6. The hardware was divided into four modules based on functions, such as sensor data acquisition, data conversion, data transmission, and power supply modules.

In the hardware system, the sensor module included pH sensors, electrical conductivity sensors, temperature sensors, and moisture sensors. During data collection, the RS485 serial communication module provided multisensor data communication services. The polling mode was used between different sensors to complete different data communication. Its working principle was to use the master 485 chip to convert the differential signal on the main bus into TTL level and then distributed it to the slave 485 chips of other branches by broadcasting, converted the differential signal from the slave chip, and sent it to each branch bus. During data transmission, the LoRa technology was applied with the wireless transmission chip SX1301, and the cyclic interleaving error correction coding algorithm was adopted to improve error correction ability. When sudden interference occurred, the maximum continuous 64 bit error could be corrected, which effectively improved the anti-interference performance of the sensors during transmission.

4. The Detection Process of the Agricultural Inputs

The prediction model of agricultural inputs includes real-time data collection, data feature analysis, data preprocessing, SSDA pretraining, SSDA-HELM feature extraction, and SOFTMAX classifier. The technical process is shown in Figure 7.

4.1. Data Collection. Ammonium sulfate (SinoChem, China), potash fertilizer (K_2O , SinoChem, China),

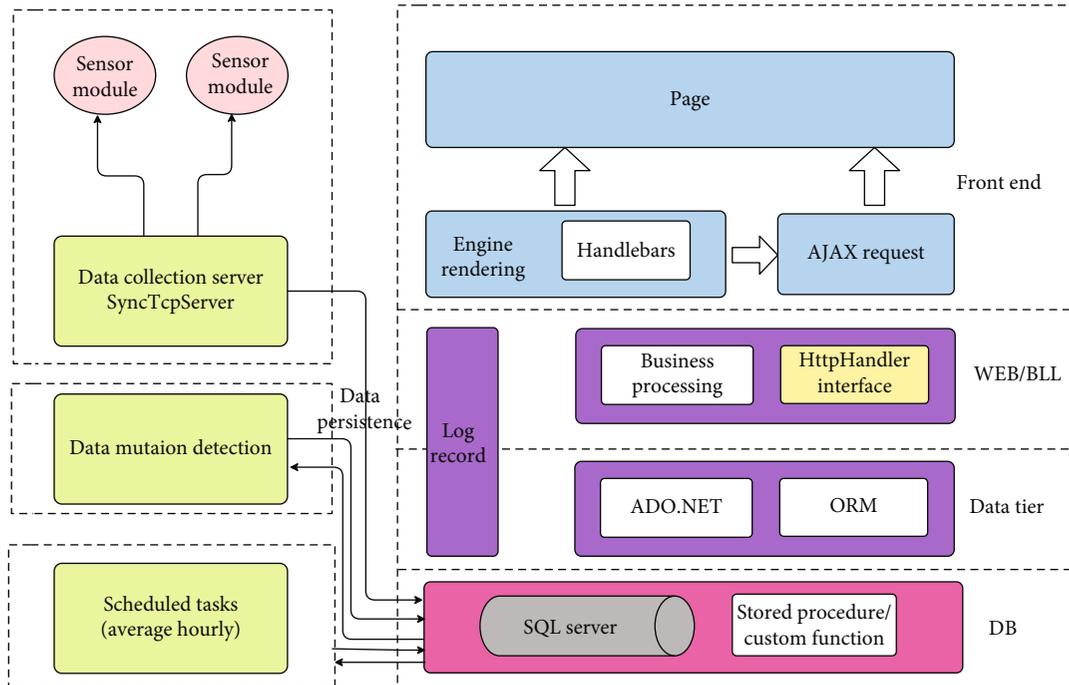


FIGURE 5: Schematic diagram of software architecture.

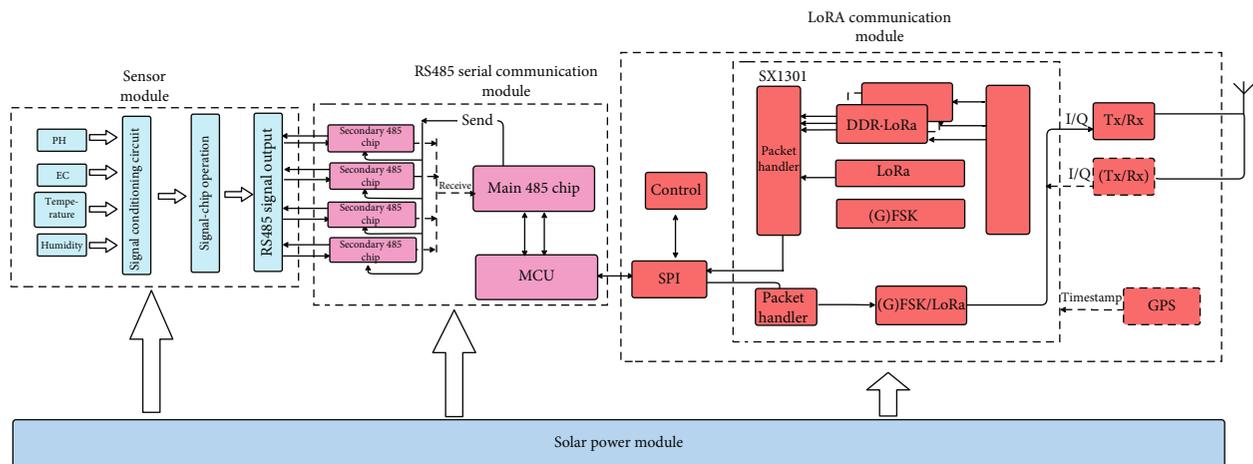


FIGURE 6: Hardware circuit architecture.

phosphate fertilizer (P2O5, SinoChem, China), Bordeaux mixture (TaoChun, China), metalaxyl (SinoChem, China), imidacloprid (Bayer, Germany), pendimethalin (SinoChem, China), and bromothalonil (SinoChem, China), which were commonly used in the cultivation of *Agastache rugosa*, were selected as experimental objects. Ammonium sulfate, potash fertilizer, and phosphate are common fertilizers. Bordeaux mixture and metalaxyl are pesticides commonly used to treat brown patch and fusarium wilt, respectively. The latter three are commonly used to kill aphids, weed, and sterilize, respectively. Aqueous solutions of these eight products were diluted according to the label directions for use. Twenty-four pots (20 cm × 20 cm × 25 cm of length, width, and height) with drainage holes at the bottom were filled with soil and were used to simulate the planting environment. Each agri-

cultural input used three pots for parallel experiments. In each pot, four sensors including electrical conductivity (EC), temperature, moisture, and pH were inserted into the soil to record the chemical data changes in real time. The experiment was carried out from October 2017 to March 2018. During the experiment, 200 ml of each agricultural input was put into the sprinkling can and sprayed into the soil. In order to expand the number of data, the same experiment was performed 50 times in each pot, so 150 experimental data were obtained for each agricultural input.

4.2. Data Analysis and Preprocessing. The sensor data was so messy that it was difficult to analyze, but with agricultural inputs in the same proportion, physical and chemical characteristics such as pH value and conductivity were relatively

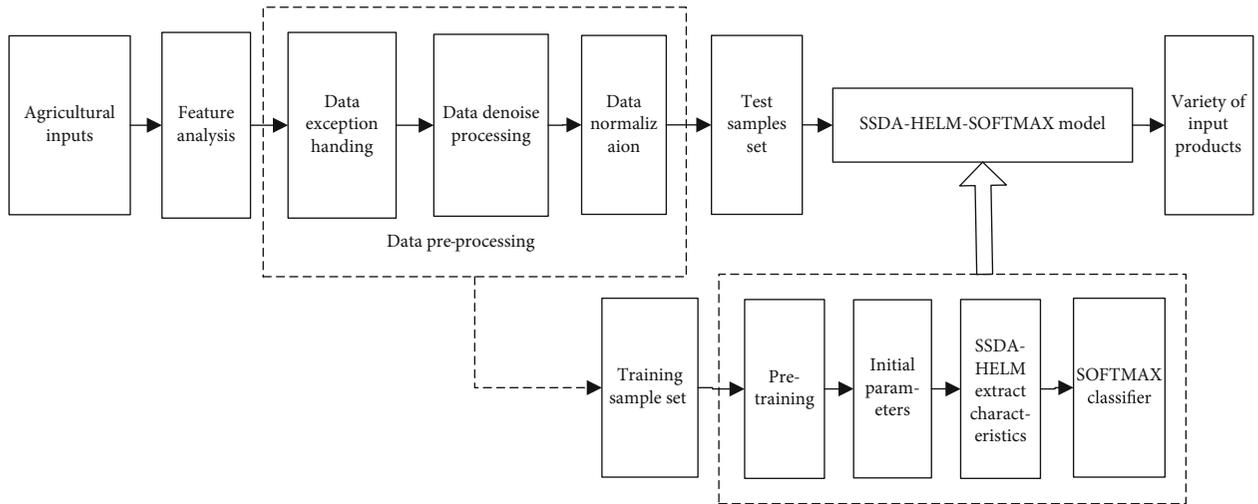


FIGURE 7: The flow diagram of modeling.

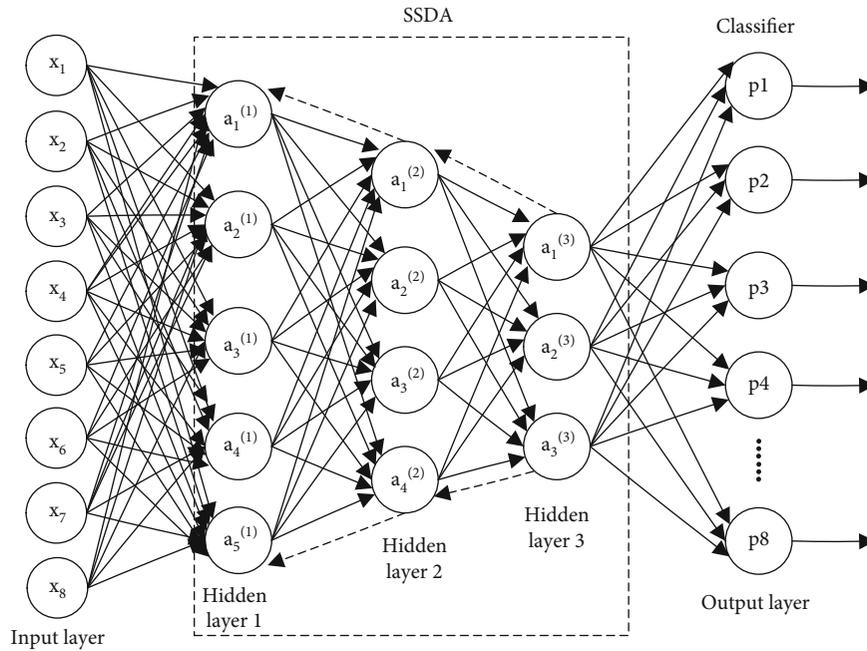


FIGURE 8: The diagram of SSDA pretraining.

fixed. They changed after the input and were affected by the soil chemical characteristics and sensor contact time. Therefore, we could analyze the sudden changes in sensor data to find the inherent law.

Due to insufficient contact between the sensor and the soil, unstable solar power supply, and other reasons, there were missing data and outliers in the sensor data. This paper used the mean method to deal with data anomalies. Every 15 seconds, the sensor data were polled, averaged to fill in missing data. When data outliers occurred, they were recorded if there were also abnormalities in other sensor data at the same time and otherwise were discarded.

In the process of data denoising, this paper uses the wavelet denoise method [22] based on thresholds [23] to remove the noise of key factors of the model input, providing a good data foundation for the construction of prediction models. Further, in the data normalization process, the z-score method was used to normalize the feature data X of the sample set, as shown in Equation (14).

$$y_i = \frac{x_i - \mu}{\sigma} \quad (1 \leq i \leq n), \quad (14)$$

where y_i was the eigenvalue of the i_{th} data after normalization, x_i was the eigenvalue of the i_{th} data, μ was the

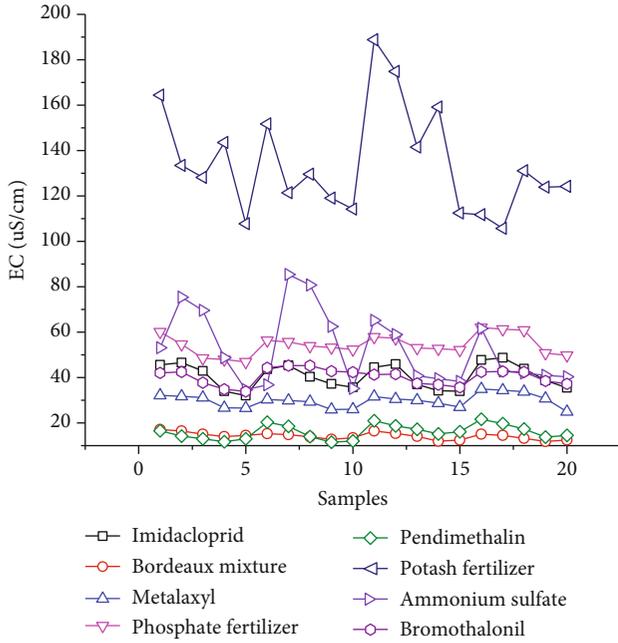


FIGURE 9: EC value after input.

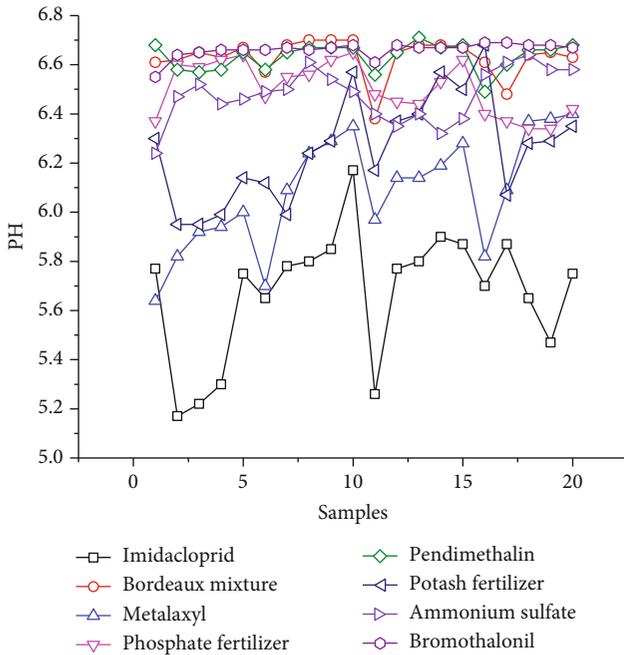


FIGURE 10: pH value after input.

TABLE 1: The value of RMSE in different model.

Layers	Nodes				
	50	100	200	300	400
2	0.0456	0.0413	0.0323	0.0183	0.0534
3	0.0437	0.0456	0.0319	0.0112	0.0478
4	0.0498	0.0406	0.0564	0.0140	0.0589
5	0.0473	0.0432	0.0589	0.0196	0.0673

average of all samples, and σ was the standard deviation of all samples.

4.3. *SSDA Pretraining and Modeling.* The diagram of SSDA pretraining is shown in Figure 8.

The network parameters were set as follows: the learning rate was 0.1, the maximum number of iterations for pre-training was 400 and for fine-tuning was 300, the sparse parameter was 0.5, the sparse penalty parameter was 3, and the activation function was the sigmoid function. During SSDA pretraining, we could extract features from complex input data and use layer-wise pretraining and fine-tuning based on unsupervised learning methods to obtain initial weights. After pretraining, we constructed the SSDA-HELM model by removing the decoding part of SSDA and connecting to the HELM. The SAE training weights were initialized to the SSDA-HELM model to extract the characteristic values of the agricultural inputs. After that, the extracted feature values were sent to the SOFTMAX classifier for classification to obtain the final agricultural input prediction model (Figure 4), which could predict the agricultural inputs based on test sample data.

5. Results and Analysis

5.1. *Data Analysis.* After the agricultural inputs were applied to the soil, the data collected by the sensors were impacted by the physical and chemical properties of the inputs. In this paper, eight agricultural inputs (ammonium sulfate, potash fertilizer, phosphate fertilizer, Bordeaux mixture, metalaxyl, imidacloprid, pendimethalin, and bromothalonil) were sprayed onto soil, and the data from 20 experiments were randomly selected for observation of the electric conductivity (EC) and pH data (Figures 9 and 10). EC refers to the ability to conduct electric current, which measures the concentration of soluble conductive ions. pH refers to the hydrogen ion concentration, which measures the proportion of the number of hydrogen ions in the solution. Since different agricultural inputs had different conductivity and hydrogen number, EC and pH could be used to distinguish different agricultural inputs. It could be observed that EC differences in response to pesticides were smaller, while the changes caused by fertilizers were comparatively large. Among them, potash fertilizer had the greatest impact on the EC value, which was above 100, while other agricultural inputs were below 80. When observing the pH value, it could be seen that the pH value of metalaxyl and imidacloprid was much lower than that of other agricultural inputs, which showed that different agricultural inputs have some differences in hydrogen ion concentration. Therefore, the differences could be used to distinguish them.

5.2. *Model and Analysis.* In the process of modeling, the EC, temperature, moisture, and pH differences before and after the agricultural inputs spraying into the soil were used as model inputs, the agricultural input categories were used as the model output, and the leave-one-out method [24] was used to cross-validate the model. Among the 1200 samples, 1199 samples were taken each time as the training set, and

TABLE 2: The characteristic data.

Inputs	Feature data					
Potash fertilizer	0.682953	0.227409	0.226057	0.494851	0.597923	0.229395
Potash fertilizer	0.569900	0.286457	0.286350	0.622601	0.205575	0.351254
Potash fertilizer	0.271896	0.729355	0.723736	0.642849	0.281236	0.668261
Ammonium sulfate	0.243788	0.251541	0.673414	0.262289	0.249249	0.572459
Ammonium sulfate	0.742556	0.345961	0.693896	0.737977	0.347710	0.726792
Ammonium sulfate	0.275226	0.686453	0.347621	0.292159	0.683852	0.550425
Imidacloprid	0.212000	0.715412	0.462585	0.447170	0.638341	0.233302
Imidacloprid	0.356072	0.629524	0.687172	0.573045	0.759579	0.637693
Imidacloprid	0.620000	0.641273	0.539924	0.328573	0.529835	0.222942
Bordeaux mixture	0.252055	0.221388	0.481921	0.593715	0.236094	0.627335
Bordeaux mixture	0.333259	0.288967	0.676369	0.200457	0.637080	0.762575
Bordeaux mixture	0.650558	0.727547	0.512039	0.281435	0.225168	0.544173
Metalaxyl	0.431794	0.667201	0.685735	0.691593	0.227002	0.714891
Metalaxyl	0.582159	0.472921	0.581209	0.579150	0.340513	0.547994
Metalaxyl	0.307182	0.772803	0.718831	0.712972	0.631892	0.542898
Phosphate fertilizer	0.562710	0.664056	0.561516	0.488892	0.688213	0.752463
Phosphate fertilizer	0.722078	0.472886	0.725181	0.612861	0.670366	0.594326
Phosphate fertilizer	0.569413	0.778950	0.562879	0.672525	0.328124	0.585843
Pendimethalin	0.770665	0.250254	0.490414	0.236053	0.377339	0.704024
Pendimethalin	0.573281	0.334498	0.651509	0.286713	0.593437	0.333318
Pendimethalin	0.571835	0.650748	0.527259	0.732943	0.371619	0.282657
Bromothalonil	0.592924	0.208807	0.689204	0.352409	0.762102	0.235710
Bromothalonil	0.202499	0.732488	0.570285	0.726488	0.580086	0.282564
Bromothalonil	0.277006	0.226396	0.729148	0.321272	0.566302	0.723354

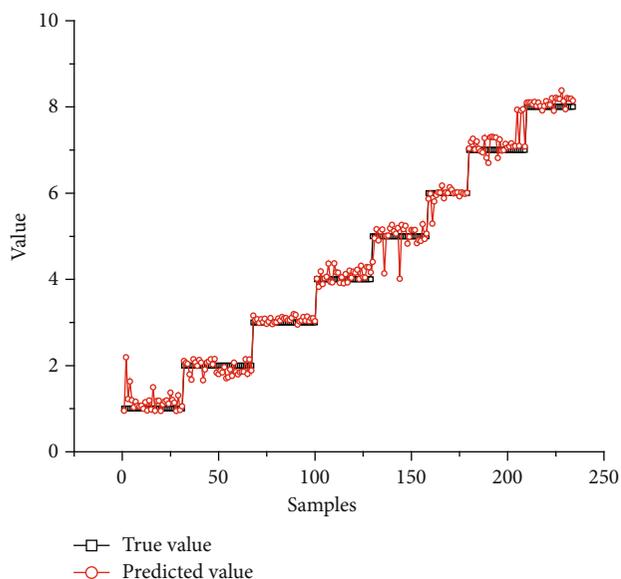


FIGURE 11: The predicted result of SSDA-HELM-SOFTMAX for test sets (ordinate 1: imidacloprid, 2: Bordeaux mixture, 3: metalaxyl, 4: phosphate fertilizer, 5: pendimethalin, 6: potash fertilizer, 7: ammonium sulfate, and 8: bromothalonil).

the remaining 1 sample was the test set. Each sample was tested individually, and the performance of the method was obtained by averaging the test results.

Because the number of nodes and layers of SSDA-HELM network directly affected the performance of the algorithm, pairwise combinations of 2, 3, 4, and 5 hidden layers and 50, 100, 200, 300, and 400 nodes were created and the root-mean-square error (RMSE) was selected to find the optimal parameters. The RMSE network parameters are shown in Table 1.

We could see that when the number of layers was 3 and the number of neurons in the first layer was 300, the performance of the model was the best. In the pretraining process, autoencoder was used for unsupervised training each layer of the SDA network. L-BFGS [25] was used for training, and the other parameters were the same as the settings in the previous study [18]. After the training was completed, the input network part of the trained weight parameters was used as the initial parameter of the SSDA-HELM network. According to the ELM network principle, the output network parameters were obtained using the least squares method and softmax was connected, and then, the supervised fine-tuning was executed. Because the ELM network used the least squares method to obtain the output network parameters, instead of the gradient descent algorithm, the problems of local convergence and poor generalization performance were avoided. Meanwhile, the problem of network

TABLE 3: The comparison of prediction accuracy.

Model	Input layer (neuron)	Hidden layer (neuron)	Output layer (neuron)	Accuracy
BP	8	300-100-50	8	93%
DBN-SOFTMAX	8	300-100-50	8	95.3%
SAE-SOFTMAX	8	300-100-50	8	95.5%
SSDA-HELM-SOFTMAX	8	300-100-50	8	97.08%

TABLE 4: The performance comparison between SSDA-HELM, BP, SAE, and DBN models.

Models	R^2_{cal}	RMSEC	R^2_{CV}	RMSECV
SSDA-HELM	0.99	0.02	0.99	0.12
DBN	0.99	0.03	0.97	0.15
SAE	0.99	0.09	0.96	0.35
BP	0.99	0.09	0.98	0.21

instability generated by the random initial value of the HELM network was solved by pretraining the network parameters of the autoencoder.

In the model training process, first SSDA was used for unsupervised training, then SSDA-HELM was used to extract data features, and then, the SOFTMAX classifier was connected and fine-tuned to improve model performance. The feature data is shown in Table 2, which refers to the nonlinear fitting and feature extraction of input data by neural network and is a higher-dimensional mapping of the input data. The prediction result is shown in Figure 11.

In order to evaluate the performance of the model, some other models such as BP, DBN [18], and SAE were also built to compare with SSDA-HELM-SOFTMAX. The BP model, SAE model, and DBN model were used to extract features, and a SOFTMAX classifier was added for prediction. Table 3 shows the prediction accuracy comparisons.

The comparison showed that SAE-SOFTMAX and DBN-SOFTMAX were more accurate than the BP, because the unsupervised layer-wise [26] was adopted and the extracted feature quality was better than by the back-propagation method. The difference between SAE and DBN was that the main feature direction though nonlinear transformation was found by SAE, while the high-level representation based on the probability distribution of samples was extracted by DBN. The results in Table 4 indicated that the high-level feature extraction based on the sample probability distribution by DBN was more in line with the characteristics of the input feature parameters. The SSDA-HELM model had the highest prediction accuracy because the SSDA model was used for pretraining; then, HELM was used to calculate the neural network output weights. Compared with other deep learning methods, SSDA could obtain the optimal parameters for initializing HELM, and HELM could be trained stably and quickly to get good classification results. Therefore, this method had the advantages of avoiding inappropriate initialization weights and avoiding local optimization and inappropriate learning rate, which was more stable and had stronger generalization ability than SAE and DBN.

The coefficient of determination (R^2) and root mean square error (RMSE) of BP, SAE, DBN, and SSDA-HELM are further compared in Table 4. It could be observed that the performance of the training set of the SAE model was the same as that of the BP model. However, the cross-validation showed that the R^2 of the SAE model was lower than that of the BP-NN model, while the RMSE was higher, indicating that SAE was not as stable as BP, although its prediction accuracy was superior.

After feature extraction, the R^2 of the SSDA-HELM model for the training set and cross-validation were the highest of all the models, both reaching 0.99. This indicated that the model was stable. Meanwhile, the RMSE for the training set and cross-validation of the SSDA-HELM model were 0.03 and 0.15, respectively, smaller than the other models. Unlike the DBN model, since the output matrix of HELM was generated by the least squares solution, once the input weights and hidden layer offsets were determined, the output matrix was uniquely determined. In this process, weight optimization was not a problem; the issues of local optimums, inappropriate learning rate, and overfitting were avoided. Therefore, the SSDA-HELM model was more stable than the DBN model. In terms of accuracy, the SSDA-HELM model was slightly lower than the DBN model [18], which was mainly due to the similarity of some inputs of *Agastache rugosa* (Figures 9 and 10) leading to more labeled categories, which decreases accuracy. In addition, the accuracy of the SSDA-HELM model was higher than that of the DBN model under the same experimental conditions.

6. Conclusions

The complex and changeable environment of *Agastache rugosa* cultivation means many factors influence the nonlinear physicochemical parameters of agricultural inputs, and traditional neural network classification used to predict agricultural inputs has the problems of local convergence, poor calculation efficiency, and poor generalization performance under the circumstances. To minimize these problems, this paper tested an input prediction model based on SSDA-HELM-SOFTMAX to predict inputs in real time. This model used the HELM to calculate the output network weights without feedback adjustment weights. It had excellent characteristics, such as fast learning speed, strong generalization ability, and resisted becoming trapped in locally optimal solutions. Meanwhile, the problem of network instability generated by the random initial value of the HELM network was solved by pretraining the network parameters of the autoencoder to initialize the parameters of SSDA-HELM model. Experiments showed that the accuracy of

the method proposed in this paper reached 97.08%, which was 4.08%, 1.78%, and 1.58% higher than BP neural network, DBN-SOFTMAX, and SAE-SOFTMAX neural networks, respectively. Therefore, the model proposed in this paper was effective and feasible, with good prediction accuracy and generalization performance, and can provide a theoretical basis and parameter support for real-time online prediction of agricultural inputs. However, quantitative detection was difficult for this paper, which required higher sensitivity of the sensors and expansion of experiments with different agricultural input concentrations. In addition, when the experiment exceeded the eight types of agricultural inputs in this paper, the model would not be applicable, which required further expanding the types of inputs and retraining the model. Nevertheless, this paper still gives a new idea and can provide a theoretical basis and method support for real-time online prediction of agricultural inputs.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

Ling Yang conceived and designed the experiments; Juan Zou performed the experiments; Hanjing Jiang and Qingxiu Wang analyzed the data; Ling Yang and Ting Wu wrote and finalized the manuscript.

Acknowledgments

This study was jointly supported by the Fund for Science and technology from Guangdong Province (2020A1515010834 and 2018A0303130034), Guangzhou Science and Technology Plan Project (202002030154 and 201903010063), Fund from Guangzhou Science and Technology Bureau (201704020030 and 201803020033), and Department of Education of Guangdong Province Bureau (2020ZDZX1060).

References

- [1] W. Haiyan and A. O. Stuanes, "Heavy metal pollution in air-water-soil-plant system of Zhuzhou City, Hunan Province, China," *Water, Air, and Soil Pollution*, vol. 147, no. 1/4, pp. 79–107, 2003.
- [2] T. Talaviya, D. Shah, N. Patel, H. Yagnik, and M. Shah, "Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides," *Artificial Intelligence in Agriculture*, vol. 4, pp. 58–73, 2020.
- [3] D. Shadrin, A. Menshchikov, A. Somov, G. Bornemann, J. Hauslage, and M. Fedorov, "Enabling precision agriculture through embedded sensing with artificial intelligence," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 7, pp. 4103–4113, 2020.
- [4] S. H. Chough, A. Mulchandani, P. Mulchandani, W. Chen, J. Wang, and K. R. Rogers, "Organophosphorus hydrolase-based amperometric sensor: modulation of sensitivity and substrate selectivity," *Electroanalysis*, vol. 14, no. 4, pp. 273–276, 2002.
- [5] S. Kumaran and C. Tran-Minh, "Insecticide determination with enzyme electrodes using different enzyme immobilization techniques," *Electroanalysis*, vol. 4, no. 10, pp. 949–954, 1992.
- [6] J. N. Banks, M. Q. Chaudhry, W. A. Matthews, M. Haverly, T. Watkins, and B. J. North Way, "Production and characterisation of polyclonal antibodies to the common moiety of some organophosphorus pesticides and development of a generic type ELISA," *Food and Agricultural Immunology*, vol. 10, no. 4, pp. 349–361, 1998.
- [7] M. T. Galceran, M. C. Carneiro, M. Diez, and L. Puignou, "Separation of quaternary ammonium herbicides by capillary electrophoresis with indirect UV detection," *Journal of Chromatography A*, vol. 782, no. 2, pp. 289–295, 1997.
- [8] G. C. R. M. Andrade, S. H. Monteiro, J. G. Francisco, L. A. Figueiredo, R. G. Botelho, and V. L. Tornisielo, "Liquid chromatography-electrospray ionization tandem mass spectrometry and dynamic multiple reaction monitoring method for determining multiple pesticide residues in tomato," *Food Chemistry*, vol. 175, pp. 57–65, 2015.
- [9] M. Shamsipur, N. Yazdanfar, and M. Ghambarian, "Combination of solid-phase extraction with dispersive liquid-liquid microextraction followed by GC-MS for determination of pesticide residues from water, milk, honey and fruit juice," *Food Chemistry*, vol. 204, pp. 289–297, 2016.
- [10] S. Datir and S. Wagh, "Monitoring and detection of agricultural disease using wireless sensor network," *International Journal of Computer Applications*, vol. 87, no. 4, pp. 1–5, 2014.
- [11] Z. Yunyao, W. Jiangjixing, H. Lijun et al., "Nanozyme sensor arrays based on heteroatom-doped graphene for detecting pesticides," *Analytical Chemistry*, vol. 92, no. 11, pp. 7444–7452, 2020.
- [12] Y. Yi, G. Zhu, C. Liu et al., "A label-free silicon quantum dots-based photoluminescence sensor for ultrasensitive detection of pesticides," *Analytical Chemistry*, vol. 85, no. 23, pp. 11464–11470, 2013.
- [13] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.
- [14] D. Lu, X. Wang, X. He, and SO Automation, "Hybrid population particle algorithm and multi-quantile robust extreme learning machine based short-term wind speed forecasting," *Power System Protection and Control*, vol. 47, 2019.
- [15] C. Chen, X. Jin, B. Jiang, and L. Li, "Optimizing extreme learning machine via generalized Hebbian learning and intrinsic plasticity learning," *Neural Processing Letters*, vol. 49, no. 3, pp. 1593–1609, 2019.
- [16] L. Zhong, Q. Zhou, K. Zhou, L. Chen, and S. Shen, "Fault diagnosis of transformer based on extreme learning machine optimized by genetic algorithm," *High Voltage Apparatus*, vol. 51, no. 8, pp. 49–53, 2015.
- [17] J. Tang, C. Deng, and G. B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Transactions on Neural Networks & Learning Systems*, vol. 27, no. 4, pp. 809–821, 2016.
- [18] L. Yang, V. Sarath Babu, J. Zou, X. C. Cai, T. Wu, and L. Lin, "The development of an intelligent monitoring system for

- agricultural inputs basing on DBN-SOFTMAX,” *Journal of Sensors*, vol. 2018, Article ID 6025381, 11 pages, 2018.
- [19] G. Hinton and R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [20] X. Yuan, J. Zhou, B. Huang, Y. Wang, C. Yang, and W. Gui, “Hierarchical quality-relevant feature representation for soft sensor modeling: a novel deep learning strategy,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 3721–3730, 2020.
- [21] X. Yuan, C. Ou, Y. Wang, C. Yang, and W. Gui, “A layer-wise data augmentation strategy for deep learning networks and its soft sensor application in an industrial hydrocracking process,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, 2021.
- [22] Y. Xu, Z. Meng, and M. Lu, “Fault diagnosis of rolling bearing based on dual-treecomplex wavelet packet transform,” *Transactions of the Chinese Society of Agricultural Engineering*, vol. 29, no. 10, pp. 49–56, 2013.
- [23] W. Hao, C. Ding, and N. Liang, “Comparison of wavelet denoise and FFT denoise,” *Electric Power Science and Engineering*, vol. 3, 2011.
- [24] X. Y. Liu, P. Li, and C. H. Gao, “Fast leave-one-out cross-validation algorithm for extreme learning machine,” *Journal of Shanghai Jiaotong University*, vol. 45, no. 8, pp. 1140–1145, 2011.
- [25] X.-y. Wang, W.-q. Wang, and H. Wang, “MRI brain image recognition method based on improved L-BFGS sparse denoising autoencoder,” *Journal of Graphics*, vol. 40, no. 2, p. 261, 2019.
- [26] B. Schölkopf, J. Platt, and T. Hofmann, “Greedy layer-wise training of deep networks,” *Advances in Neural Information Processing Systems*, vol. 19, pp. 153–160, 2007.