Research Article

The Development of an Intelligent Monitoring System for Agricultural Inputs Basing on DBN-SOFTMAX

Ling Yang,¹ V. Sarath Babu,² Juan Zou,¹ Xu Can Cai,¹ Ting Wu ¹ and Li Lin ²

¹School of Information Science and Technology, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China

²Guangdong Provincial Water Environment and Aquatic Products Security Engineering Technology Research Center, Guangzhou Key Laboratory of Aquatic Animal Diseases and Waterfowl Breeding, Guangdong Provincial Key Laboratory of Waterfowl Healthy Breeding, College of Animal Sciences and Technology, Zhongkai University of Agriculture and Engineering, Guangzhou, Guangdong 510225, China

Correspondence should be addressed to Ting Wu; 405684932@qq.com and Li Lin; linli@zhku.edu.cn

Received 7 May 2018; Revised 13 August 2018; Accepted 30 August 2018; Published 28 October 2018

Academic Editor: Marco Grossi

Copyright © 2018 Ling Yang 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.

To solve the problem of unreliability of traceability information in the traceability system, we developed an intelligent monitoring system to realize the real-time online acquisition of physicochemical parameters of the agricultural inputs and to predict the varieties of input products accurately. Firstly, self-developed monitoring equipment was used to realize real-time acquisition, format conversion and pretreatment of the physicochemical parameters of inputs, and real-time communication with the cloud platform server. In this process, LoRa technology was adopted to solve the wireless communication problems between long-distance, low-power, and multinode environments. Secondly, a deep belief network (DBN) model was used to learn unsupervised physicochemical parameters of input products and extract the input features. Finally, these input features were utilized on the softmax classifier to establish the classification model, which could accurately predict the varieties of agricultural inputs. The results showed that when six kinds of pesticides, chemical fertilizers, and other agricultural inputs were predicted through the system, the prediction accuracy could reach 98.5%. Therefore, the system can be used to monitor the varieties of agrarian inputs effectively and use in real-time to ensure the authenticity and accuracy of the traceability information.

1. Introduction

The traceability system of agricultural products is a powerful tool for solving the food safety issues [1]. The information on the farm inputs, such as pesticides and fertilizers used for cultivation, is one of the most concerned problems of the consumers. Currently, some companies have established their own traceability system of agricultural products [2]. However, it is reasonable that the consumers do not trust the appreciable information recorded by the producers themselves because of the lack of supervision. Therefore, the establishment of a traceability system, which can record the information timely, accurately, and ultimately is an urgent need.

There have been many reports about the rapid techniques for detection of agricultural inputs. To name a few, Deng et al. established a liquid chromatography-tandem mass

spectrometry (LC-MS) method for the simultaneous determination of benzoylurea pesticide residues in vegetables [3]. Zheng et al. found LC-MS method for the detection of pesticide residues in milk [4]. Selisker et al. used a competitive enzyme-linked immunosorbent assay (ELISA) to detect paraquat [5]. Alcocer et al. have developed a polyclonal antibody for detection of organophosphorus pesticides [6]. Kumaran and Tran-Minh used cholinesterase electrode to detect pesticides [7]. Chough et al. used a carbon electrode to identify the organophosphorus insecticide [8]. Seemingly, many of the rapid detection techniques for agricultural inputs have been established. However, most of the current monitoring of agricultural input information is still a kind of residue detection of the postproduction stage, and it is still difficult to monitor the data via real-time online. Furthermore, the current established traceability system records the

FIGURE 1: The overall system architecture.

traceability information, which is mainly entered manually; therefore, the information is not timely and accurate.

It is desirable to seek an alternative method to overcome these drawbacks. In this report, based on sensors and DBN-SOFTMAX algorithm, we developed an intelligent monitoring system for the agricultural inputs. Different from chemical-based agrarian inputs detection methods described above. This paper proposed using the sensors arranged in the soil to realize the monitoring and prediction of farming inputs. In general, sensors were employed in agriculture to achieve environmental monitoring such as moisture and temperature [9, 10] or to attain precise agricultural control [11]. In this paper, the sensors placed in the soil were used to collect the physicochemical characteristics of the inputs such as pH and EC value, and then the artificial intelligence algorithm was used to analyze the above sensor data and finally realized the intelligent monitoring and prediction of agricultural inputs.

2. Monitoring System Design

2.1. Working Principles and Overall Architecture. The overall structure of the intelligence-monitoring platform for agricultural inputs is shown in Figure 1.

The monitoring equipment collects data every 15 seconds to obtain the physicochemical parameters of agricultural inputs, such as pH value, electronic conductivity (EC), and temperature, in real time. After data preprocessing, analog-to-digital conversion, and RS485 [12] format conversion, LoRa (long range) module transmits the data to the LoRa gateway [13] and converts them into the RJ45 format. Subsequently, the data will be received and stored in the cloud

server, and data cleaning and reduction process are performed to obtain useful data for further modeling and classification. During the modeling process, the input data is continuously increased to the training samples, and the model is updated once a week to obtain more accurate prediction results.

2.2. Hardware Design. The monitoring system mainly consists of sensor module, low-power digital processor, multichannel AD/DA conversion module, RS485 serial communication module, LoRa wireless communication module, and solar power module. The sensor module includes a pH sensor, an EC sensor, and a temperature sensor. The RS485 serial port communication module provides multisensor data fusion service. It uses polling mode to collect different sensor data of the same monitoring point through the RS485 interface to complete multisensor data fusion. LoRa wireless communication module developed by LoRa spread spectrum chip SX1278; its transmission distance and penetration ability are more than one time higher than those of traditional FSK [14]. In LoRa wireless communication, the capability of error correction is stronger since the algorithm of cyclic interleaving error correction coding is expected to be adopted. The maximum continuous error correction is 64 bits, which can reduce the retransmission of a large number of erroneous data packets, to improve the anti-interference performance and transmission distance. The hardware structure of the monitoring system is shown in Figure 2.

2.3. Software Design

2.3.1. The LoRa Node Software Design. There are three kinds of nodes in LoRa, namely, sensor, routing, and aggregation

FIGURE 2: The hardware structure of monitor terminal.

nodes. The routing node is responsible for forwarding data. The aggregation node does not collect data, but as a control center, it sends synchronization information to the monitoring network and the received data to the local monitoring and remote monitoring centers. The corresponding node software is designed to perform the functions of each node. In this paper, the sensor node was used as an example to introduce the software design method. The C language was used to develop software, and the flow chart of the program is shown in Figure 3:

The entire programming process uses the modular design, mainly including equipment initialization, data acquisition and processing, serial communication, and wireless communication. PC monitoring controls acquisition cycle and acquisition command and controlling center software. If the node software receives the acquisition command sent by the PC monitoring center program, it immediately responds and transmits the collected data to the corresponding sensor according to different Modbus protocol commands.

2.3.2. The Monitoring Center Software Design. The software workflow diagram is shown in Figure 4. The monitoring center software uses C# to develop and communicate with the LoRa gateway module through TCP/IP network programming and obtains monitoring data transmitted by the LoRa gateway to the Internet network. The monitoring software mainly includes parameters setting, real-time monitoring, data processing, and other functions. Parameter setting function is to set the acquisition cycle, acquisition Modbus command, and other parameter settings. The real-time monitoring function is to collect sensor data in real time. A data processing function included calling DBN-SOFTMAX prediction model code, analyzing the received data, matching the established model, and predicting the varieties of inputs.

3. DBN-SOFTMAX Algorithm and Modeling

3.1. The DBN-Based Feature Extraction Method. Restricted Boltzmann Machine (RBM) [15], which was part of DBN

FIGURE 3: The node program flow.

[16, 17], could extract features that are more abstract and significantly improve the ability of neural network generalization [18]. Each RBM was a two-layer model that contained only one hidden layer, and each RBM training output was used as the input for the next RBM.

3.1.1. Restricted Boltzmann Machine (RBM). If $\theta = \{W_{ij}, a_i, b_j\}$, where W_{ij} represented the connection weight between the visible unit i and the hidden element j . m, n was the

FIGURE 4: The software workflow.

number of hidden cells and visible cells, respectively. Both the visible and hidden units were binary variables. That is, $\forall i, j, v_i \in \{0, 1\}, h_j \in \{0, 1\}$, a_i was the offset of the visible element i , b_j was the offset of the visible element j , T was the number of samples, h was a hidden layer unit, and v was a visible layer unit.

RBM was an undirected graph model [19, 20] which was used to solve the value of the parameter θ , to fit the given training data, and the extracted feature (Figure 5).

RBM task was used to fit the input training data, figured out the optimal parameter θ , and completed the feature extraction. The parameter θ could be learned in the training set to maximize the logarithmic likelihood function. The formula was as follows:

$$\theta^* = \arg \max \mathcal{L}(\theta) = \arg \max \sum_{t=1}^T \log P(V^{(t)} | \theta), \quad (1)$$

where

$$\begin{aligned} \mathcal{L}(\theta) &= \sum_{t=1}^T \log P(V^{(t)} | \theta) = \sum_{t=1}^T \log \sum_h P(V^{(t)}, h | \theta) \\ &= \sum_{t=1}^T \left(\log \sum_h \exp[-E(V^{(t)}, h | \theta)] \right. \\ &\quad \left. - \log \sum_v \sum_h \exp[-E(v, h | \theta)] \right). \end{aligned} \quad (2)$$

The key to solving the optimal parameter θ^* was to obtain the partial derivative of $\log P(V^{(t)} | \theta)$ for W_{ij} , a_i , b_j , and

FIGURE 5: The RBM model.

other parameters. Assuming that θ' was a parameter value of θ , the logarithmic likelihood function concerning θ' was

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta'} &= \sum_{t=1}^T \frac{\partial}{\partial \theta'} \left(\log \sum_h \exp[-E(V^{(t)}, h | \theta)] \right. \\ &\quad \left. - \log \sum_v \sum_h \exp[-E(v, h | \theta)] \right) \\ &= \sum_{t=1}^T \left(\sum_h \frac{\exp[-E(V^{(t)}, h | \theta)]}{\sum_h \exp[-E(V^{(t)}, h | \theta)]} \times \frac{\partial(-E(V^{(t)}, h | \theta))}{\partial \theta} \right. \\ &\quad \left. - \sum_v \sum_h \frac{\exp[-E(v, h | \theta)]}{\sum_v \sum_h \exp[-E(v, h | \theta)]} \times \frac{\partial(-E(v, h | \theta))}{\partial \theta} \right) \\ &= \sum_{t=1}^T \left(\left\langle \frac{\partial(-E(V^{(t)}, h | \theta))}{\partial \theta} \right\rangle_{P(h|V^{(t)}, \theta)} \right. \\ &\quad \left. - \left\langle \frac{\partial(-E(v, h | \theta))}{\partial \theta} \right\rangle_{P(v, h | \theta)} \right). \end{aligned} \quad (3)$$

Since the number of samples T was known, the partial derivative of the logarithmic likelihood function for the connection weight W_{ij} , the offset a_i of the visible layer element, and the offset b_j of the hidden layer unit could be expressed by $P(h | V^{(t)}, \theta)$ and $P(v, h | \theta)$. $P(h | V^{(t)}, \theta)$ was a hidden probability distribution of training sample $V^{(t)}$; $P(v, h | \theta)$ was a joint probability function for a given state (v, h) ; the function was

$$P(v, h | \theta) = \frac{e^{-E(v, h | \theta)}}{Z(\theta)}, \quad (4)$$

where $E(v, h | \theta)$ was the energy function of RBM and $Z(\theta)$ was the normalization factor.

$$\begin{aligned} E(v, h | \theta) &= -\sum_{i=1}^n a_i v_i - \sum_{j=1}^m b_j h_j - \sum_{i=1}^n \sum_{j=1}^m v_i W_{ij} h_j, \\ Z(\theta) &= \sum_{v, h} e^{-E(v, h | \theta)}. \end{aligned} \quad (5)$$

3.1.2. CD Algorithm. It has been shown that the normalization factor $Z(\theta)$ was difficult to be solved [21]. Therefore, the joint probability function $P(v, h | \theta)$ was also difficult to calculate. To solve this problem, the fast learning algorithm based on contrast divergence was used to training data, and the steps were as follows:

- (1) $P(v, h | \theta)$ was given by the formula $P(v, h | \theta) = e^{-E(v, h | \theta)} / Z(\theta)$, $P(v | \theta) = 1 / Z(\theta) \sum_h e^{-E(v, h | \theta)}$
- (2) The RBM network structure was connected between layers, no connection within the layer and the structure of symmetry, i.e., when the state of the visible cell was fixed, an activating probability of the j th hidden element was

$$P(h_j = 1 | v, \theta) = \sigma \left(b_j + \sum_i v_i W_{ij} \right). \quad (6)$$

When the state of the hidden cell was fixed, activating probability of the i th hidden element was

$$P(v_i = 1 | h, \theta) = \sigma \left(a_i + \sum_j h_j W_{ij} \right). \quad (7)$$

The binary state of all hidden layer units was calculated from equation (6). After the state of all hidden layer units was determined, the i th visible unit v_i value of 1 probability according to equation (7) was determined, and a reconstruction of the visible layer was created

- (3) The parameter updated formula in the data training process was as follows:

$$\begin{aligned} \Delta W_{ij} &= \epsilon \left(\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{recon}} \right), \\ \Delta a_i &= \epsilon \left(\langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{recon}} \right), \\ \Delta b_j &= \epsilon \left(\langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{recon}} \right), \end{aligned} \quad (8)$$

where ϵ was the learning rate, and $\langle \bullet \rangle_{\text{recon}}$ was the distribution that represented the reconstructed model definition.

3.2. The Softmax Classifier. In the above process, what is finally obtained was the feature values of $x^{(i)}$. However, in the prediction of input varieties, it was necessary to categorize output and to add a softmax classifier [22] to the output layer to organize the learned feature values. The diagrammatic drawing of the softmax classifiers was presented in Figure 6. The marked training set $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$, among $y^{(j)} \in \{1, 2, \dots, k\}$ representative training sample $x^{(i)}$ is k . The given test input $x^{(i)}$, which is the classification model calculates the probability that it belongs to each category.

Thus, to a sample set with k types, output k -dimensional vector to represent the probability vector. The j th element in the probability vector represents the probability of belonging to the j th category, and the sum of values of elements is 1. Specifically, our hypothesis function $h_\theta(x)$ is shown as

$$\begin{aligned} h_\theta(x^{(i)}) &= \begin{bmatrix} p(y^{(i)} = 1 | x^{(i)}; \theta) \\ p(y^{(i)} = 2 | x^{(i)}; \theta) \\ \vdots \\ p(y^{(i)} = k | x^{(i)}; \theta) \end{bmatrix} \\ &= \frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_k \end{bmatrix}. \end{aligned} \quad (9)$$

Among $\theta_1, \theta_2, \theta_3, \dots, \theta_k$ are network model parameters, as

$$\theta = \begin{bmatrix} -\theta_1^T - \\ -\theta_2^T - \\ \vdots \\ -\theta_k^T - \end{bmatrix}, \frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}}, \quad (10)$$

this item mainly restricts the probability values from 0 to 1, which the sum of the probability values is 1.

FIGURE 6: The softmax classifier.

In equation (9), the probability of the sample $x^{(i)}$ output by the classifier belonging to class j is ($1\{\text{true}\} = 1, 1\{\text{false}\} = 0$):

$$p(y^{(i)} = j | x^{(i)}; \theta) = \frac{e^{\theta_j^T x^{(i)}}}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}} = p_1 = \prod_{j=1}^k p_j^{1\{y^{(i)}=j\}}. \quad (11)$$

The likelihood function corresponding to training samples is

$$\begin{aligned} L(\theta) &= \prod_{i=1}^m p(y^{(i)} = j | x^{(i)}; \theta) = \prod_{i=1}^m \prod_{j=1}^k p_j^{1\{y^{(i)}=j\}}, \\ l(\theta) &= \log L(\theta) = \sum_{i=1}^m \sum_{j=1}^k 1\{y^{(i)} = j\} \log(p_j) \\ &= \sum_{i=1}^m \sum_{j=1}^k 1\{y^{(i)} = j\} \log\left(\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}}}\right). \end{aligned} \quad (12)$$

The parameter θ that maximizes the likelihood function as the optimal parameter of the softmax classifier. The cost function of the softmax regression model is

$$J(\theta) = -\frac{1}{m} \left[\sum_{i=1}^m \sum_{j=1}^k 1\{y^{(i)} = j\} \log\left(\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}}}\right) \right]. \quad (13)$$

The cost function is minimized by the gradient descent method; the gradient function is as follows:

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m} \left[\sum_{i=1}^m x^{(i)} \left(1\{y^{(i)} = j\} - p(y^{(i)} = j | x^{(i)}; \theta) \right) \right]. \quad (14)$$

The softmax classifier has an unusual feature: it has a “redundant” set of parameters [23]. To illustrate the feature, if the vector μ was subtracted from the parameter

vector θ_j , each θ_j becomes $\theta_j - \mu$ ($j = 1, 2, \dots, k$). The function is shown as

$$\begin{aligned} p(y^{(i)} = j | x^{(i)}; \theta) &= \frac{e^{(\theta_j - \mu)^T x^{(i)}}}{\sum_{l=1}^k e^{(\theta_l - \mu)^T x^{(i)}}} = \frac{e^{\theta_j^T x^{(i)}} e^{-\mu^T x^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}} e^{-\mu^T x^{(i)}}} \\ &= \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}}}. \end{aligned} \quad (15)$$

In equation (15), we can see that the parameters $\theta_j - \mu$ and θ_j can both gain the same result. In other words, when θ_j is the optimal parameter, $\theta_j - \mu$ can also have the same effect. It is the disadvantage of having redundant parameters in the softmax classifier. The loss function of the softmax classifier is distinctly nonconvex. Although there is a minimum point, the minimum value is in “flat” space and not at a single point. That is, all points in the area can get a minimum value. To make the cost function a strictly convex function, we need to add a weight attenuation term, as follows:

$$J(\theta) = J(\theta) + \frac{\lambda}{2} \sum_{i=1}^m \sum_{j=1}^k \theta_{ij}^2, \quad (16)$$

$$\nabla_{\theta_j} J(\theta) = \nabla_{\theta_j} J(\theta) + \lambda \theta_j.$$

3.3. Modeling. In establishing the model, the collected data were first normalized, and then DBN was used for unsupervised training to extract features. However, these features were not directly applicable to classification [24–26], so the softmax classifier was added to the output to perform supervised classification training. The flow diagram is shown in Figure 7.

4. Experiment Design

In this experiment, we used six agricultural inputs, including phosphate (P_2O_5 , SinoChem, China), potassium (K_2O , SinoChem, China), compound fertilizer (carbamide, nitrogen phosphorus potassium, SouthRanch, China), Podol pesticide (TaoChun, China), imidacloprid (Bayer, Germany), and oxamoxime (HeYi, China), purchased from local stores in Guangzhou, China. The first three of these inputs were chemical fertilizers, the latter three were pesticides, and their aqueous solutions were placed in dilution ratios (500:1) for use. Eighteen pots of soil-filled bottom drainable basins were prepared and set in an open-air environment. The EC sensors, pH sensors, and moisture sensors were inserted into the soil, and the power was turned on to enable to collect the sensor data in real time. During the period from October 2016 to March 2017, 200 ml of each inputs aqueous solution was sprayed into three pots of soil. Over 50 experiments, the soil parameter data before and after the input, including moisture proportion (before input), conductivity (before input), the pH value (before input), moisture ratio (after input), conductivity (after input), and the pH value (after input) were recorded. 150 data were collected for

FIGURE 7: The flow diagram of modeling.

FIGURE 8: Changes in pH before and after input.

FIGURE 9: Changes in EC/moisture before and after input.

each type of input product, and the total number of data was 900.

5. Results and Discussion

5.1. Data of Sensors. The sensor data before input were not the same in each experiment; the collected sensor data after input minus before input could better explain the characteristics of the input. Six agricultural inputs were sprayed into the soil, respectively, and pH, conductivity, and moisture data were collected before and after the input. In this paper, 20 times of experimental data were randomly selected for observation.

Observing the collected pH values, before input they were close to 7, which was neutral. After applying six agricultural inputs, the pH value decreased. As shown in Figure 8, the changes of pH at different inputs were similar and overlapped

with each other. It indicates that there was no significant difference in the power of hydrogen during several agricultural inputs applied.

Further observation of changes in electrical conductivity (EC), since the EC value was very sensitive to moisture content, there was a significant error in the shift in the EC value observed separately. The EC value divided the moisture content, and the obtained ratio was counted as shown in Figure 9. Observation shows that in general, EC changes in pesticides, including podol pesticide, imidacloprid, and oxamoxime were smaller, while fertilizers were comparatively larger. Changes of potassium had the most considerable EC value, and its value was more significant than the three; however, the EC value changes of pesticide were less than 1.5.

5.2. Modeling and Analysis. Sensors collected the trained and relevant experimental data of the agricultural inputs

TABLE 1: The characteristic data table.

y^a	Feature data					
1	0.261774	0.51821	0.363377	0.333239	0.40386	0.748109
1	0.249594	0.569245	0.360975	0.317917	0.442481	0.728785
1	0.25111	0.560419	0.361024	0.32006	0.437188	0.730672
2	0.811001	0.796877	0.348194	0.270687	0.189169	0.231068
2	0.839929	0.777112	0.698331	0.380023	0.302937	0.187739
2	0.822922	0.811853	0.321081	0.268463	0.183998	0.220862
3	0.397166	0.347973	0.591175	0.642555	0.696136	0.526804
3	0.441901	0.3287	0.630434	0.696979	0.742546	0.464949
3	0.417766	0.325697	0.646081	0.692987	0.75344	0.477638
4	0.716266	0.526859	0.587048	0.715965	0.590975	0.389491
4	0.764408	0.593614	0.599644	0.594801	0.447832	0.317517
4	0.735743	0.50656	0.56267	0.718762	0.570059	0.368529
5	0.221322	0.570325	0.261664	0.264869	0.386502	0.750886
5	0.226302	0.555085	0.258739	0.26812	0.375292	0.751574
5	0.23323	0.578087	0.251799	0.265587	0.376804	0.743913
6	0.760376	0.252841	0.753006	0.841325	0.298932	0.814017
6	0.752798	0.194803	0.781346	0.848471	0.26095	0.846151
6	0.760472	0.275192	0.74997	0.839426	0.310604	0.809583

^a y is for the types of agricultural inputs. 1: potassium fertilizer; 2: compound fertilizer; 3: imidacloprid; 4: Podol liquid; 5: oxamoxime; 6: phosphate fertilizer.

FIGURE 10: Three-dimensional distribution of feature values.

prediction models. The main content of each data sample is input product category, the moisture proportion (before input), conductivity (before input), the pH value (before input), moisture ratio (after input), conductivity (after input), and pH value (after input). In establishing the model, the leave-one-out method [27] was used for the cross-

FIGURE 11: Three-dimensional distribution of original values.

FIGURE 12: The predicted results of DBN-SOFTMAX for test sets. In the ordinate, 1: potassium fertilizer; 2: compound fertilizer; 3: imidacloprid; 4: Podol liquid; 5: oxamoxime; and 6: phosphate fertilizer.

validation to test the model's performance. Each of the 900 samples was taken separately, and then the remaining 899 samples were used to build the model. The model independently tested each sample, and the results were averaged to obtain the average performance of the method.

When using DBN for feature extraction, a four-layer neural network was established, the number of neurons in each layer was 300, 100, 20, and 6, respectively. The activation function of the hidden layer was "logsig", the training method was the L-BFGS algorithm [28], and the output layer function was a softmax function. In DBN feature extraction,

TABLE 2: The forecast accuracy comparison table.

Training data	Model	Input layer (neuron)	Hidden layer (neuron)	Output layer (neurons)	Accuracy
Raw data	BP	6	10	6	84.7%
Raw data	DBN-BP	6	300-100-20-6	6	97.8%
Raw data	DBN-SOFTMAX	6	300-100-20-6	6	98.5%

RBM used an unsupervised learning method to train each layer of RBM networks separately, ensuring that the feature information was preserved as much as possible during the mapping process. After the training was completed, a classifier was set at the last level of the DBN; using supervised fine-tuning, the best training results were obtained. Since each layer of RBM network only adjusted the weights in its own layer, it did not guarantee that the feature vector mapping of the entire DBN was optimal. After supervised fine-tuning, the process of RBM network training could be regarded as the initialization process of weight parameters of a deep neural network, which enabled the DBN network to overcome the disadvantages of the traditional BP network due to random initialization weight parameters and easy to fall into local optimum and long training time.

During the training, the number of iterations was 400, the learning rate was 0.1, and the training error target was set to 0.001. After the training, the extracted feature data were shown in Table 1:

Through unsupervised training of DBN and nonlinear mappings, the features were obtained from the input data, such as pH, moisture, and conductivity. After extracting features, the cohesion of the same types of agricultural inputs and the variances of different farm inputs could be better demonstrated. After dimensional reduction by the principal component analysis (PCA) method [29], the three-dimensional distribution of feature values and the three-dimensional distribution of original values were shown in Figures 10 and 11. It could be observed that the feature values extracted by DBN can be separated, but the initial input values without feature extraction were scattered and there were some confusion. Therefore, it was evident that using the extracted feature values for classification could achieve better prediction results.

We used this model for predicting the agricultural inputs. First, by applying the RBM-based DBN model, unsupervised training on raw data was carried out to improve the robustness of the network. Second, the feature data were obtained, and the softmax classifier was added to the back of DBN, the feature data were taken as the input, and the categories of inputs were taken as the output. Thirdly, the feature data and the tagged samples were combined to fine-tune the softmax classifier, and finally, the model was established to predict the accuracy. The result was shown in Figure 12. When 900 samples were predicted, thirteen samples were wrongly predicted with the model accuracy of 98.5%.

To evaluate the performance of the DBN-softmax model, BP-neural network and DBN-BP model were also established and the prediction accuracy of the input products was compared.

TABLE 3: DBN model performance comparison with BP-NN and SVM.

Models	R^2_{cal}	RMSEC	R^2_{CV}	RMSECV
DBN	0.99	0.03	0.99	0.15
BP-NN	0.99	0.09	0.94	0.40
SVM	0.99	0.09	0.98	0.21

As shown in Table 2, the DBN-BP accuracy was higher than that of the BP neural network, because DBN adopted the unsupervised layer-wise [16] mechanism training mode. The weights gained through DBN were obtained by learning the structure of input data, which were close to the optimal global values. However, BP neural network, whose initial values were randomly set, was prone to problems such as local optimal and gradient diffusion, so it required manual adjustment parameters [30]. When DBN-BP was compared with DBN-SOFTMAX, we used the same DBN structure to extract features with different classifiers, so the prediction accuracy was the same. It indicated that the prediction accuracy depended mainly on the quality of feature extraction, and the results obtained by different classifiers were not very different.

Further research on the determination coefficient (R-Square) and root mean square error (RMSE) when testing model performance. This paper compared conventional modeling methods such as BP-NN and support vector machine (SVM) [31]. When establishing the SVM model, the nu-SVM algorithm was selected, and the radial basis function (RBF) was selected as the kernel function. The error penalty coefficient γ and the kernel function parameter nu were 0.255 and 1, which were determined by grid searching technique [32]. In the process of modeling, the calibration sets were used to build the model, and the leave-one-out method was used for cross-validation to test the robustness and adaptability of the model further.

As shown in Table 3, when the BP-NN model was observed, the performance of the calibration sets was the same as SVM. However, the determination coefficient of cross-validation was smaller than SVM, and the root mean square error was more extensive than SVM, revealing that the model was not as accurate and stable as the SVM model. When the DBN model was observed, it could be seen that after feature extraction, the determination coefficients of calibration sets and cross-validation were the largest compared to BP-NN and SVM, reaching 0.99 and 0.99, respectively. Meanwhile, the RMSE of calibration sets and cross-validation in the DBN model were the smallest, which were 0.03 and 0.15, respectively. So in general, compared to SVM

and BP-NN, DBN was still considered to be the optimal modeling method.

6. Conclusions

Based on the self-developed monitoring equipment and DBN-SOFTMAX model, we have developed a platform for intelligent monitoring of agricultural inputs; perform online and real-time monitoring on farms. When the agricultural inputs were applied in farms, we could compare the types of inputs and application time with the data entered by the administrators in the traceability system. Once the producers do not record the traceability information or input the wrong information, our system can capture related data timely and accurately, then automatically provides safety warning to the producers, to ensure that the traceability information is true and accurate. The intelligent monitoring platform will pave a new way for the development of traceability systems.

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, Ting Wu, and Li Lin conceived and designed the experiments; Ling Yang performed the experiments; Ting Wu, Juan Zhou, Xu Can Cai, and V Sarath Babu analyzed the data; Ling Yang, Ting Wu, and Li Lin wrote and finalized the manuscript.

Acknowledgments

This study was jointly supported by the National Natural Science Fund (61501531); fund for Science and Technology from Guangdong Province (2015A020209173, 2017A020225007); fund from Guangzhou Science and Technology Bureau (201704020030, 201803020033); and "Innovation and Strong Universities" special funds (KA170500G) from the Department of Education of Guangdong Province. V. Sarath Babu was supported by Chinese Postdoctoral Science Foundation.

References

- [1] A. Regattieri, M. Gamberi, and R. Manzini, "Traceability of food products: general framework and experimental evidence," *Journal of Food Engineering*, vol. 81, no. 2, pp. 347–356, 2007.
- [2] T. Bosona and G. Gebresenbet, "Food traceability as an integral part of logistics management in food and agricultural supply chain," *Food Control*, vol. 33, no. 1, pp. 32–48, 2013.
- [3] H. Y. Deng, W. Xie, Z. Q. Zhou, B. Li, and Q. T. Jiang, "Determination of 11 benzoylurea insecticides residues in vegetable by LC-MS/MS," *Journal of Instrumental Analysis*, vol. 28, no. 8, pp. 970–974, 2009.
- [4] J. H. Zheng, G. F. Pang, C. L. Fan, and M. L. Wang, "Simultaneous determination of 128 pesticide residues in milk by liquid chromatography-tandem electrospray mass-spectrometry," *Chinese Journal of Chromatography*, vol. 27, no. 3, pp. 254–263, 2009.
- [5] M. Y. Selisker, D. P. Herzog, R. D. Erber, J. R. Fleeker, and J. Itak, "Determination of Paraquat in fruits and vegetables by a magnetic particle based enzyme-linked immunosorbent assay," *Journal of Agricultural and Food Chemistry*, vol. 43, no. 2, pp. 544–547, 1995.
- [6] M. J. Alcocer, P. P. Dillon, B. M. Manning et al., "Use of phosphonic acid as a generic hapten in the production of broad specificity anti-organophosphate pesticide antibody," *Journal of Agricultural and Food Chemistry*, vol. 48, no. 6, pp. 2228–2233, 2000.
- [7] S. Kumaran and C. Tran-Minh, "Insecticide determination with enzyme electrodes using different enzyme immobilization techniques," *Electroanalysis*, vol. 4, no. 10, pp. 949–954, 2010.
- [8] 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, 2015.
- [9] J. Burrell, T. Brooke, and R. Beckwith, "Vineyard computing: sensor networks in agricultural production," *IEEE Pervasive Computing*, vol. 3, no. 1, pp. 38–45, 2004.
- [10] M. Srbinska, C. Gavrovski, V. Dimcev, A. Krkoleva, and V. Borozan, "Environmental parameters monitoring in precision agriculture using wireless sensor networks," *Journal of Cleaner Production*, vol. 88, pp. 297–307, 2015.
- [11] D. D. Chaudhary, S. P. Nayse, and L. M. Waghmare, "Application of wireless sensor networks for greenhouse parameter control in precision agriculture," *International Journal of Wireless & Mobile Networks*, vol. 3, no. 1, pp. 140–149, 2011.
- [12] H. J. Jia and Z. H. Guo, "Research on the technology of rs485 over ethernet," in *2010 International Conference on E-Product E-Service and E-Entertainment*, pp. 1–3, Henan, China, November 2010.
- [13] M. Aref and A. Sikora, "Free space range measurements with Semtech Lora™ Technology," in *2014 2nd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems*, pp. 19–23, Offenburg, Germany, September 2014.
- [14] N. Azmi, S. Sudin, L. M. Kamarudin et al., "Design and development of multi-transceiver Lorafi board consisting LoRa and ESP8266-Wifi communication module," *IOP Conference Series: Materials Science and Engineering*, vol. 318, article 012051, 2018.
- [15] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*, G. Montavon, G. B. Orr, and K. R. Müller, Eds., vol. 7700 of Lecture Notes in Computer Science, pp. 599–619, Springer, Berlin Heidelberg, 2012.
- [16] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
- [17] B. Schölkopf, J. Platt, and T. Hofmann, "In Greedy layer-wise training of deep networks," *International Conference on Neural Information Processing Systems*, pp. 153–160, 2006.
- [18] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

- [19] N. Li, J. Shi, and M. Gong, "Change detection in synthetic aperture radar images based on fuzzy restricted Boltzmann machine," in *Bio-inspired Computing – Theories and Applications. BIC-TA 2016*, M. Gong, L. Pan, T. Song, and G. Zhang, Eds., vol. 681 of Communications in Computer and Information Science, pp. 438–444, Springer, Singapore, 2016.
- [20] G. V. Tulder and M. D. Bruijne, "Combining generative and discriminative representation learning for lung CT analysis with convolutional restricted Boltzmann machines," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1262–1272, 2016.
- [21] P. M. Long and R. A. Servedio, "Random classification noise defeats all convex potential boosters," *Machine Learning*, vol. 78, no. 3, pp. 287–304, 2010.
- [22] L. Chen, M. Zhou, W. Su, M. Wu, J. She, and K. Hirota, "Softmax regression based deep sparse autoencoder network for facial emotion recognition in human-robot interaction," *Information Sciences*, vol. 428, pp. 49–61, 2018.
- [23] K. Duan, S. S. Keerthi, W. Chu, S. K. Shevade, and A. N. Poo, "Multi-category classification by soft-max combination of binary classifiers," in *Multiple Classifier Systems. MCS 2003*, T. Windeatt and F. Roli, Eds., vol. 2709 of Lecture Notes in Computer Science, pp. 125–134, Springer, Berlin, Heidelberg, 2003.
- [24] B. Liao, J. Xu, J. Lv, and S. Zhou, "An image retrieval method for binary images based on DBN and Softmax classifier," *IETE Technical Review*, vol. 32, no. 4, pp. 294–303, 2015.
- [25] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 7559–7566, Brisbane, Australia, May 2018.
- [26] J. Yang, Y. Bai, F. Lin, M. Liu, Z. Hou, and X. Liu, "A novel electrocardiogram arrhythmia classification method based on stacked sparse auto-encoders and softmax regression," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 10, pp. 1733–1740, 2017.
- [27] M. Kearns and D. Ron, "Algorithmic stability and sanity-check bounds for leave-one-out cross-validation," *Neural Computation*, vol. 11, no. 6, pp. 1427–1453, 1999.
- [28] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1–3, pp. 503–528, 1989.
- [29] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1–3, pp. 37–52, 1987.
- [30] B. Yang, X. H. Su, and Y. D. Wang, "Bp neural network optimization based on an improved genetic algorithm," in *Proceedings. International Conference on Machine Learning and Cybernetics*, vol. 61, pp. 64–68, Beijing, China, November 2002.
- [31] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [32] L. I. Bing, Q. Z. Yao, Z. M. Luo, and Y. Tian, "Gird-pattern method for model selection of support vector machines," *Computer Engineering & Applications*, vol. 44, no. 15, pp. 136–138, 2008.