DOI: 10.35382/TVUJS.13.4.2023.2844

A NOVEL SYSTEM FOR PREDICTING THE DAMAGE OF RICE DISEASES IN AN GIANG PROVINCE, VIETNAM

Thanh-Nghi Doan[1]*, Phuoc-Hai Huynh[2]

Abstract – *Many researchers have recently considered information technology applications to support smart agriculture. Numerous solutions and information technology systems have been implemented to benefit farmers, such as an e-commerce website for exchanging agricultural products, an agricultural information system, a livestock information management system, and an agricultural environment monitoring system. This paper presents a novel system that combines artificial back-propagation neural networks with genetic algorithms for predicting the damage of rice diseases in An Giang Province. The system predicts rice disease damage in the coming weeks in 11 districts in An Giang based on input parameters such as climate conditions and previous rice disease damage. The approach was evaluated on a dataset collected by the An Giang Plant Protection Department over 22 years. The rice diseases forecasting system produced promising results and received positive feedback from forecast department specialists, with an RMSE of 19.31 on the test dataset (ha).*

Keywords: artificial neural network, genetic algorithms, rice diseases.

I. INTRODUCTION

An Giang is a province in the Mekong Delta where the economy is still heavily dependent on agricultural production. The two main spearheads of this agricultural production are aquaculture and rice cultivation. Rice cultivation has now achieved remarkable productivity gains, propelling the province's annual rice output to the top of the country (the total annual rice cultivation area is about 640,000 ha with an output of about 3.8 million tons). Despite the substantial production yield, farmers do not generate significant profits since production expenses are too expensive. One of the stages in each season that has cost farmers a lot of effort and production costs is disease prevention for rice fields. The number of crops has been expanded in the expanding trend of agricultural production, to increase the number of crops and income, and the land has been fully utilized. Furthermore, environmental pollution and climate change have eased the advanced circumstances for some viruses to spread on a large scale. As a result, the prevalence of rice disease is increasing. Therefore, developing an early warning system for rice diseases is critical and essential. Artificial neural networks (ANNs) [1] have always been the focus of many researchers whose concerns relate to the operating mechanisms of artificial neurons and biological neural networks. In recent years, ANNs have become popular and successfully applied in various fields, such as finance, health, geology, physics, and agriculture. ANN models will be applicable and produce results that are trustworthy wherever there are issues with forecasting, classifying, and controlling. Besides, in the plant protection field, geographic information systems (GIS) and remote sensing (RS) are widely applied to agricultural forecasting [2–5]. In Vietnam, many provinces and cities have applied GIS to forecast agricultural pests, such as Dong Thap (2010), Quang Ninh (2011), Bac Ninh (2014), and An Giang (2014). The software, which is used by agriculture for forecasting in An Giang, is provided by Technical Science Service Consultant Joint Stock Company (Scitec Company). This software incorporates multiple statis-

[1,2]An Giang University, VNUHCM, Vietnam
*Corresponding author: dtnghi@agu.edu.vn
Received date: 06th July 2023; Revised date: 03rd October 2023; Accepted date: 04th December 2023

tical algorithms, serving as a supportive tool for forecasting purposes. Prediction methods that are used in this kind of software are based mainly on traditional statistical models, which must know beforehand the mathematical models. Therefore, the forecast is made based on the linear regression model. However, the majority of forecasting issues are complex and nonlinear. The use of a linear form in mapping can reduce the accuracy of forecast models [6–8]. To overcome this major drawback, many researchers often perform some transformations on the independent and dependent variables before building ANN models. This process is called data linearization. The difficulty in developing a linear regression model consists of determining linear mapping coefficients and data linearization. However, this difficulty can be solved by building a direct nonlinear model on the dataset, and the artificial neural network learning approach is a solution. Therefore, this research proposes an appropriate model of an artificial neural network for forecasting the situation of rice diseases in An Giang Province. The overall process of the proposed approach is shown in Figure 1. The research findings make several contributions: (i) The creation of a new dataset of rice diseases in An Giang Province for 22 years with 330 samples; (ii) The development of An efficient ANN model to forecast the rice disease level of An Giang Province; and (iii) The creation of a GIS map for visualizing the situation of rice diseases.

II. MATERIALS AND METHODS

A. Artificial neural network

An artificial neural network is a computer program that simulates the information processing of a biological neural network. It is made from different elements (called artificial neurons or processing elements) connected through links (each link has a weight value called a link weight) that work as a unity to solve a specific issue. In practice, many ANNs are good tools for modeling nonlinear statistical data and are used to model complex relationships between input and output data. An ANN is configured for a specific application (pattern recognition, data classification, etc.) through a learning process from a set of training samples. Learning is, in substance, the process of calibrating the link weights between neurons. There are three common machine learning methods: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning is the most popular method, which is typically the back-propagation algorithm.

B. Back-propagation algorithm

The back-propagation algorithm (BP) [9] is often applied to multi-layer feed-forward neural networks consisting of processing elements with activated and continuous functions. Given the training set consisting of p samples, the input-output pairs are as follows:

$$x^{(k)},\quad d^{(k)},\quad k = 1, 2, \ldots,\quad p \tag{1}$$

The back-propagation algorithm provides procedures for altering the weight values in back-propagation neural networks to properly classify given input samples. The basis of this algorithm is the reduced gradient method [10]. For a given input-output pair, the back-propagation algorithm performs two phases of the data stream. Firstly, the input pattern x(k) is propagated from the input layer to the output layer and it generates the actual output signal as the forward output. Next, the error, the difference between d(k) and y(k), will be backpropagated from the output layer to the input layers to help them adjust their weight values. For example, consider a three-layer network as shown in Figure 2, with n nodes in the input layer, n nodes in the hidden layer, and m nodes in the output layer. Solid lines indicate the forward spread of the signals; the dashed lines indicate the backward propagation of the errors.

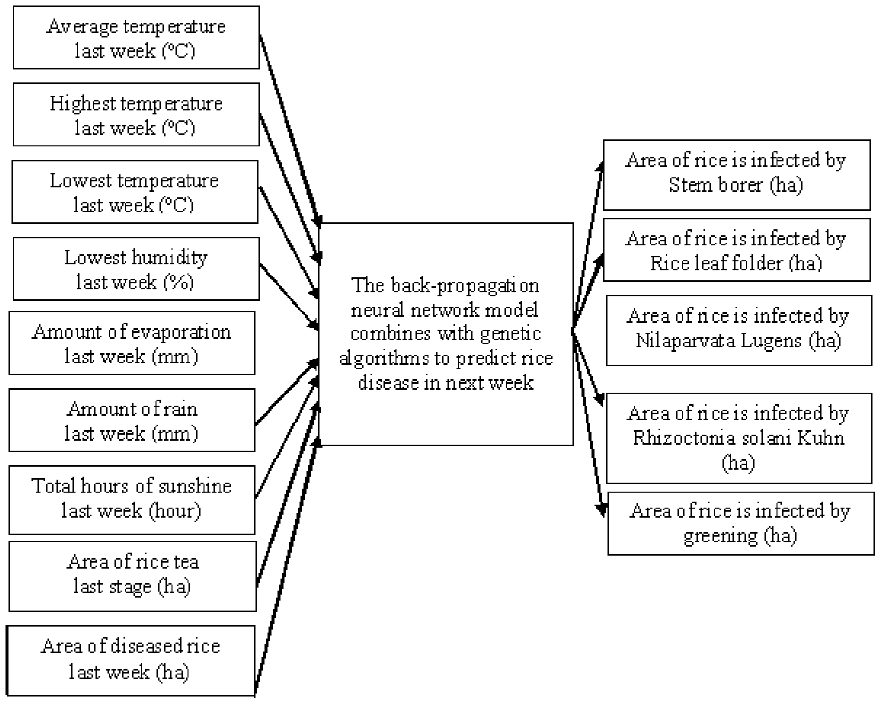

Fig. 1: An artificial neural network model for predicting rice disease degrees

Consider a pair of training data points (x, d). With the given input pattern x, the node q in the hidden layer receives the network's input:

$$net_q = \sum_{j=1}^{m} v_{qj} x_j \qquad (2)$$

And generates the output:

$$Z_q = a(net_q) = a\left(\sum_{j=1}^{m} v_{qj} x_j\right) \quad (3)$$

The network's input for the ith node of the output layer is as follows:

$$net_i = \sum_{q=1}^{l} w_{iq} z_q = \sum_{q=1}^{l} w_{iq}\, a\left(\sum_{j=1}^{l} w_{qj} x_j\right) \quad (4)$$

And generates the output:

$$y_i = a(net_i) = a\left(\sum_{q=1}^{l} w_{iq} z_q\right) = \left(\sum_{q=1}^{l} w_{iq}\, a\left(\sum_{j=1}^{m} v_{qj} x_j\right)\right) \quad (5)$$

The above equations determine the forward spread of input signals through the layers.

The cost function is defined by Equation (6).

$$E(w) = \frac{1}{2}\sum_{i=1}^{n}(d_i - y_i)^2 = \frac{1}{2}\sum_{i=1}^{n}[d_i - a(net_i)]^2 = \frac{1}{2}\sum_{i=1}^{n}\left[d_i - a\left(\sum w_{iq} z_q\right)\right]^2 \quad (6)$$

By the reduced gradient method, the weights are updated as follows:

$$\Delta w_{iq} = -\eta \frac{\partial E}{\partial W_{iq}} \quad (7)$$

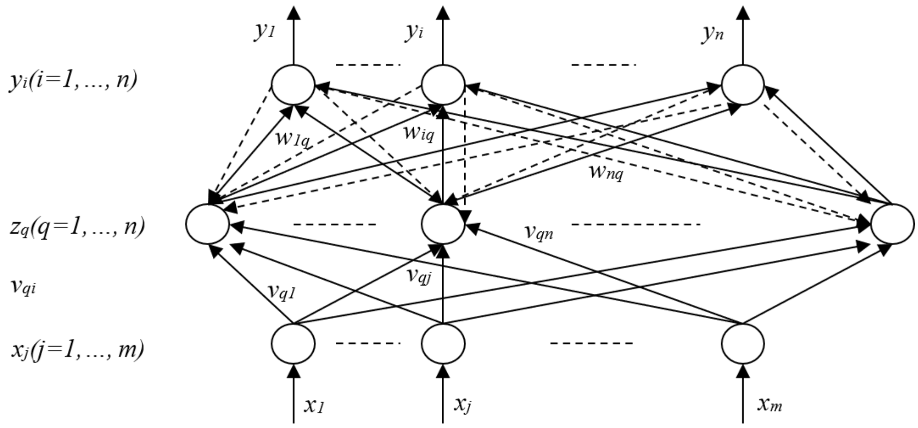

Fig. 2: A back-propagation neural network with three layers

Thus,

$$\Delta w_{iq} = -\eta \left[\frac{dE}{\partial y_i}\right]\left[\frac{\partial y_i}{\partial net_i}\right]\left[\frac{\partial net_i}{\partial w_{iq}}\right] \quad (8)$$
$$= \eta[d_i - y_i][a'(net)][z_q] = \eta\delta_{oi}z_q$$

Here, δ_{lq} is the error signal of the qth node in the hidden layer and is defined as:

$$\delta_{lq} \equiv -\frac{\partial E}{\partial net_q} = -\left[\frac{\partial E}{\partial z_q}\right]\left[\frac{\partial z_q}{\partial net_q}\right]$$
$$= a'(net_q)\sum_{i=1}^{n}\delta_{oi}w_{iq} \quad (9)$$

The net_i is the i^{th} node's input in the output layer.

And $a'(net_i) = \delta a(net_i)/\delta net_i$

The weights on input connections and hidden classes are updated as follows:

$$\Delta v_{qj} = -\eta\left[\frac{\partial E}{\partial v_{qj}}\right] = -\eta\left[\frac{\partial E}{\partial net_q}\right]\left[\frac{\partial net_q}{\partial v_{qj}}\right]$$
$$= -\eta\left[\frac{\partial E}{\partial z_q}\right]\left[\frac{\partial z_q}{\partial net_q}\right]\left[\frac{\partial}{\partial v_{qj}}\right] \quad (10)$$

Thus,

$$\Delta v_{qj} = \eta\sum\left[(d_i - y_i)a'(net_i)w_{iq}\right]a'(net_q)x_j \quad (11)$$

Deducing that

$$\Delta v_{qj} = \eta\sum\left[\delta_{oi}w_{iq}\right]a'(net_q)x_j$$
$$= \eta\delta_{lq}x_j \quad (12)$$

Here, δ_{lq} is the error signal of the q_{th} node in the hidden layer and is defined as:

$$\delta_{lq} \equiv -\frac{\partial E}{\partial net_q} = -\left[\frac{\partial E}{\partial z_q}\right]\left[\frac{\partial z_q}{\partial net_q}\right]$$
$$= a'(net_q)\sum_{i=1}^{n}\delta_{oi}w_{iq} \quad (13)$$

The net_q is the q_{th} node's hidden input.

In the case of general networks with arbitrary classes, back-propagation has the form as:

$$\Delta w_{ij} = \eta\delta_i x_j = \eta\delta_{output-i}x_{input-j} \quad (14)$$

Here, the 'output-i' and the 'input-j' determine two connections from the j_{th} node to the j_{th} node; x_j is the appropriate input, an active point from a hidden node or from external input, δ_i is the learning signal.

When the activated function is a sigmoid function, we have:

$$\delta_{oi} = \frac{1}{2}(l - y_i^2)[d_i - y_i] \quad (15)$$

And

$$\delta_{hq} = \frac{1}{2}(1 - z_q^2)\sum\delta_{oi}w_{iq} \quad (16)$$

C. Genetic algorithm

The genetic algorithm (GA) [11] is a technique of computer science that is capable of solving problems by imitating human or animal evolution under specified conditions of the environment. The means to solve this problem is a computer program consisting of several steps, from selecting a representative solution to a problem to choosing adaptive functions as well as transformational methods to create more appropriate solutions [12]. Therefore, GA does not focus on the unique and accurate solution as in the classical approach, but it considers all possible solutions and chooses the best or optimal one. The flowchart of a genetic algorithm can be generally described as shown in Figure 3.

D. Artificial neural network model for predicting rice diseases

Based on national technical regulations on methods of detecting plant pests [13, 14] and agricultural professionals' experience [15], the prediction of rice diseases can be based on the following factors, including climate, area of rice tea stages, and the situation of rice diseases in the last week. In particular, the environmental and climatic factors include the average temperature (^oC), the highest temperature (^oC), the lowest temperature (^oC), the lowest humidity (%), the amount of evaporation (mm), the amount of rain (mm), and the total hours of sunshine (h). The growth stages of rice include plating, tillering, flowering, and bud (ha).

E. Data mining techniques

The back-propagation neural network paired with genetic algorithms is the data mining approach used in this article to address the disease problem affecting rice. This combination model is generally depicted in Figure 1. ANN is a computational approach that involves the development of learnable mathematical structures. One of the most important advantages of ANN is the ability to create highly accurate prediction models that meet the requirements of data mining for the prediction of time-dependent events. The methods used in back-propagation neural networks are supervised learning methods. Its operation is to analyze past data to generate a prediction rule for data in the future. Samples extracted by ANNs are expressed at the outputs of the network. In back-propagation neural networks, each node is associated with a threshold, so the patterns (or rules) of a concept are the combinations of weights that are greater than the threshold.

F. Back-propagation neural network with delay and shift window

Typically, real-time data are used as both the output data and the input sample data for prediction problems. Real-time inference refers to data from both the present and the past. The 'before-after' relationship is the centerpiece of learning from real-time data. The most challenging aspect of learning these patterns is how to identify, define, and maintain those relationships. To address this issue, a back-propagation neural network is transformed into a time-delay neural network and implements a shift window concept, as illustrated in Figure 4.

If the input data has x bits and is delayed with m shift windows, there will be m*x input units to encode the input sample. When new data is inserted, it will be placed at the input node at a certain end of the network. The older data will be shifted by one unit on the network's entry nodes, the same as the shift register. Based on the data, how many times the lookback windows will shift by a corresponding amount is predicted. Correspondingly, the larger the prediction need, the larger the look-ahead windows. An example of a network's 'input-output' relation that performs prediction with real-time data can be formulated in the Equation (17).

$$Y_{(t)} = f_{nn}(X_{(t-1)}, X_{(t-2)}, \dots, X_{(t-m)}) \quad (17)$$

Where m is the input shift window and the number of the network's nodes, the output shift window is equal to 1 and the network's output nodes. The relationship shows that the network

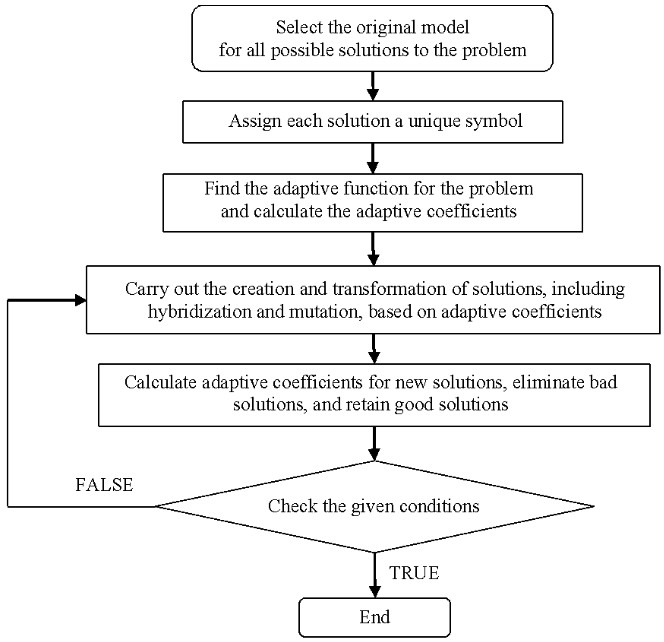

Fig. 3: General flowchart of genetic algorithm

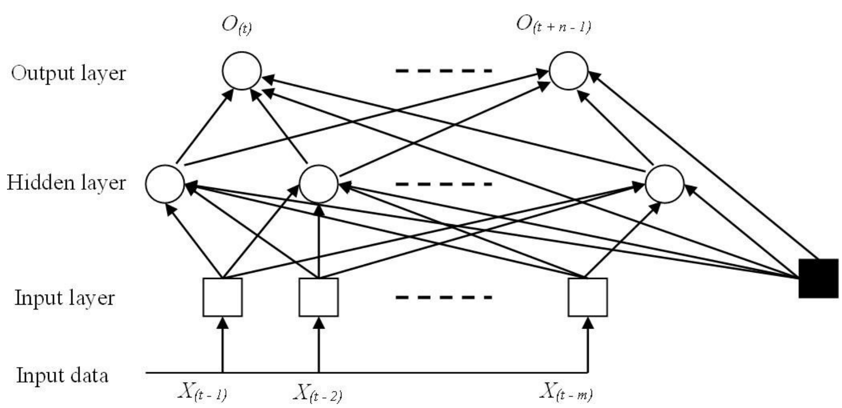

Fig. 4: Back-propagation neural network on real-time data with a shift window

will predict Y at the next time t, with the present time being t-1, knowing that its values and its dependent values are in the past and present time.

G. Applied genetic algorithms in finding the optimal set of weights for back-propagation neural networks

By using a local search method to minimize errors, back-propagation neural networks encounter the following difficulties: (i) it is not possible to find the optimal set of weights for a given network's structure, but only an acceptable set of weights; and (ii) the network may not converge or converge very slowly. These difficulties will affect data mining speed as well as the data mining results will not be as expected.

Genetic algorithms can quickly identify areas of extreme interest, yet finding extreme values in those areas is extremely difficult [16, 17]. This is an optimization algorithm. Thus, this paper uses genetic algorithms to support the optimization process of the weights set on neural networks. First, the hybridization process begins by initializing the first population of chromosomes (it is the network's weights encoding set) as the input for genetic algorithms. Then, genetic algorithms will produce a new generation. This generation is decoded and sent back to the neural network to evaluate the adaptability of everyone. This is the weight set. Before continuing evolution based on assessed adaptability, the system will retain some of the most adaptable individuals of the present generation. The process will be repeated until the system finds an optimal weight set. After the neural network is optimized, it will use a back-propagation algorithm to find local extremes in this optimized space. Finally, the network will be used to exploit the data and extract the prediction patterns.

III. RESULTS AND DISCUSSION

The rice disease prediction model using back-propagation neural networks and genetic algorithms was coded in the C++ programming language. All the experiments were conducted on a computer with an Intel Core i5 CPU, 8GB of RAM, and the Microsoft Windows 10 operating system.

Training data: Our method was tested on a dataset of rice diseases in the winter-spring crop, provided by the An Giang Plant Protection Department. Winter-spring crops are usually from January to the end of March, so this department usually collects data from districts for 15 weeks. In order to synchronize with the neural network's input data, we collected data on the climatic environment for 15 weeks which was provided by An Giang National Centre for Hydro-Meteorological Forecasting. To ensure adequate data for the training set and to enhance the reliability of the forecasting model, the research used collected for 22 years (from 1999 to 2021). A total of 330 samples were collected in the winter-spring crop by multiplying 15 weeks by the 22 years of data. Out of these samples, 300 were utilized for model training, while the remaining 30 were reserved for model testing. The 16 inputs of the network included data relating to average temperature (^{o}C), highest temperature (^{o}C), lowest temperature (^{o}C), lowest humidity (%), evaporation (mm), rainfall (mm), total sunshine hours (hours), area of rice tea at the seedling stage, tillering, bud, flowering, and situation of 5 types of diseases sick in the last week. The five outputs of the network included infected area in the next week (in hectares) of stem borer, leaf roller, brown planthopper, leaf spot disease, and leaf yellowing disease. Due to challenges in data collection, our dataset was limited. To address this limitation, we employed the bootstrap method during training. Bootstrap sampling, a widely used resampling technique in statistics, involves repeatedly drawing samples with replacements from the available data to estimate the sampling distribution of a statistic. In this research, we utilized bootstrap sampling to enhance the robustness and generalization capabilities of our model. The fundamental concept of bootstrap sampling lies in mitigating overfitting by exposing the model to diverse subsets of the data in each iteration. Additionally, it offers a means to gauge the variability in the model's performance.

Creating the network training file: After putting all of the necessary data into the storage files, a network training file is created from this data. A sample training file is shown in Figure 5.

Forecasting accuracy measurement: Predictions are always subject to error because the situation with tourism is complicated and dependent on many factors. Forecasting error will be a measure of how close the predicted value is to the actual value. This error is the difference between the actual value (d_t) and the predicted value (y_t) and is calculated as follows:

$$e_t = |d_t - y_t| \qquad (18)$$

A forecasting model is considered good if the forecasting error on testing data is quite small. If the model was built properly, the fluctuations

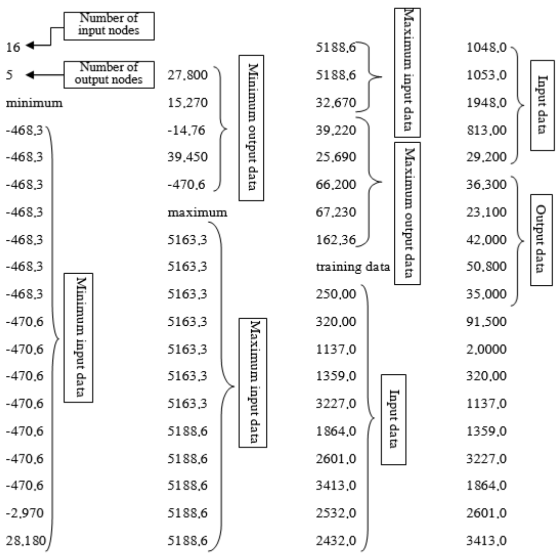

Fig. 5: An example file that stores network training data

of the prediction error would not go in any direction. The fluctuations in prediction errors are frequently caused by unforeseeable external phenomena. This means that the random oscillations of e_t in each period are purely random oscillations around the predicted value y_t, so the total prediction error will be zero. At present, the two most popular methods of calculating forecasting errors are absolute prediction error and relative prediction error. The calculation of forecasting errors is quite good in cases where the problem is complex, and the forecasting model has a large error. Therefore, to measure the prediction accuracy of the rice disease prediction model, we use the relative error, namely the Root Mean Square Error (RMSE).

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n} e_t^2}{n}} \qquad (19)$$

where e_t is the prediction error in t stage, n is the number of forecasting observations.

Neural network training: To build the forecasting system, 11 neural networks were trained for 11 districts in An Giang, respectively. The training of these networks is the same. Therefore, this paper only presents the training process of a neural network model of Long Xuyen City, in the winter-spring crop, with the stem borer. The data sample set is taken from 1994 to 2015, so we have 330 samples, of which 30 samples are used to test the network. This process is divided into two main steps as follows:

Step 1: Find global optimization weights: This step is referred to as the 'warm-up' phase, where the network will be trained by genetic algorithms using randomly generated weights and configured as in Table 1.

Table 1: Configuring neural networks when training by genetic algorithm

Parameters	Value
Population size	40
Chromosome length	8
Generation number	500
pick_fittest	TRUE
Hybrid probability	0.6
Mutation probability	1.0 / population_size
Fault tolerance	0.01
Learning type	Genetic algorithm

Table 2: Configuring neural networks with back-propagation algorithm

Parameters	Value
Learning type	Back-propagation algorithm
Lowest error	0.001
Alpha	0.9
Learing speed	1.0
Iterations number	50,000
Sigmoid Output Layer	No

After 500 generations of network training using genetic algorithms, the RMSE error on the testing data set was 609.409, as shown in Table 3. As shown in Figure 6, the ANN output (blue line) is already relatively close to the actual line of the training data (red line). This proves that at the 'warm-up' stage, the ANN model has learned and generalized the data. In the test region (yellow), the network output curve has a relative value that matches the test data, which means that the network model can predict future data, that has not been learned before. However, the error

RMSE is still too large because, in this phase, the network is looking for a global optimization weights space. To reach the local extremity in weight space, we continue to Step 2 for training with a time-delay back-propagation network and shift window.

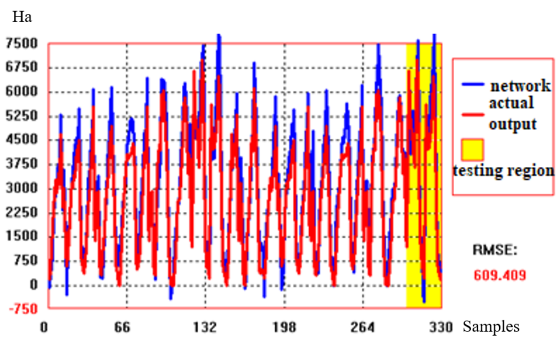

Fig. 6: Graph of comparison between network output and actual output using GA

Step 2: To locate the local extremes in the global optimization weights space, after completing Step 1, a global optimization weight space was employed. However, the network model has not yet really reached its extreme value. Therefore, this step continues to train the network with the existing weights set, but this time the learning model is the back-propagation algorithm. In this step, the training process is carried out in the following manner to reduce the network's training time: (i) the network is first trained with an error threshold of 0.75 or 0.5; (ii) then, to reach the 0.001 error threshold, the network is trained with the network error parameter reduced to 0.005, 0.0025, and finally 0.001. The specific configuration for the training in Step 2 is shown in Table 3.

After 50,000 generations of training time-delay back-propagation neural networks with a shift window, we have the following results:

Table 3: Configuring neural networks when training by genetic algorithm

No.	Area infected with pests (in hectares)		Error ($e_t=d_t-y_t$)
	Actual output (d_t)	Network output (y_t)	
1	-91.2588	207.4850	298.7438
2	455.3049	524.2260	68.9211
3	533.9167	1128.0900	594.1732
4	1026.5219	1849.7700	823.2482
5	2239.1890	2420.0000	180.8110
6	2909.8696	3318.4299	408.5603
7	3046.4023	2906.6399	-139.7625
8	3598.2600	3098.5400	-499.7200
9	3820.0103	3534.3301	-285.6802
10	3794.6565	3176.2100	-618.4465
11	5317.1084	4729.9199	-587.1885
12	2352.9365	2464.6399	111.7034
13	3618.9238	3037.8601	-581.0637
14	2524.4556	2261.7200	-262.7356
15	1540.5690	1133.6000	-406.9690
16	-308.0587	608.1020	916.1606
17	1356.8221	968.4780	-388.3441
18	2057.8721	2074.3501	16.4780
19	2784.4365	2843.4099	58.9734
20	3251.2764	3202.3101	-48.9663
21	3503.1768	3059.3000	-443.8767
22	4007.9775	3454.7500	-553.2275
23	4307.4590	3352.4500	-955.0090
24	4631.1777	4259.7598	-371.4180
25	4367.7407	4179.7798	-187.9609
26	5509.8184	4524.8198	-984.9985
27	3587.9702	3176.6699	-411.3003
28	1366.9540	927.3100	-439.6440
29	859.1884	558.9320	-300.2563
30	932.7178	380.9310	-551.7869

- With the training data set (300 samples): As shown in Figure 7, the network's output (blue color) almost coincides with the actual output (red color). This shows that the network has modeled the entire set of rice disease situations.

- With the testing data set (30 samples), we can see that the network was able to predict unlearned data, the neural network's output line is nearly closest to the actual output line, the RMSE prediction error on the testing data set is 10.649 (Table 4).

Table 3: Configuring neural networks when training by genetic algorithm (cont.)

No.	Area infected with pests (in hectares)		Error (e_t=d_t-y_t)
	Actual output (d_t)	Network output (y_t)	
309	6653.0684	7018.1899	365.1216
310	7648.4180	6307.2798	-1341.1382
311	4533.7095	3865.1399	-668.5696
312	1690.7820	1538.7200	-152.0620
313	1072.6321	626.2660	-446.3661
314	-200.0077	520.1650	720.1727
315	-508.2415	242.9690	751.2104
316	3615.5625	3525.5801	-89.9824
317	5541.3818	5642.0698	100.6880
318	4415.6631	4246.7100	-168.9531
319	4951.5244	4276.0601	-675.4644
320	5700.4868	4616.2798	-1084.2070
321	6118.1045	4838.2500	-1279.8545
322	6385.0801	5446.8599	-938.2202
323	6999.5483	6538.1201	-461.4282
324	7870.9580	6048.3501	-1822.6079
325	2851.4417	2745.7800	-105.6616
326	1194.8302	761.2940	-433.5362
327	826.5892	448.2700	-378.3192
328	661.6802	398.3690	-263.3112
329	415.0981	206.2930	-208.8051
		RMSE	609.409

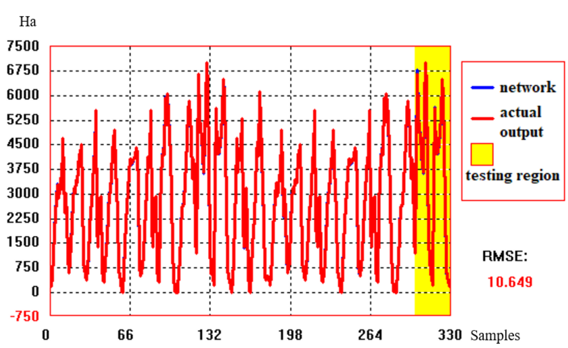

Fig. 7: A comparison between the neural network output and actual output on the training dataset using back propagation algorithm (BP)

This shows that the network has generalized quite well to the data that was not learned during the training.

Table 4: The results of trained neural networks on a testing data set using BP

No.	Area infected with pests (in hectares)		Error (e_t=d_t-y_t)
	Actual output (dt)	Network output (y_t)	
1	208.0557	207.4850	-0.5707
2	524.8705	524.2260	-0.6445
3	1134.4835	1128.0900	-6.3936
4	1864.0269	1849.7700	-14.2568
5	2434.9619	2420.0000	-14.9619
6	3322.1338	3318.4299	-3.7039
7	2912.4121	2906.6399	-5.7722
8	3101.3596	3098.5400	-2.8196
9	3534.7561	3534.3301	-0.4260
10	3178.9893	3176.2100	-2.7793
11	4724.8398	4729.9199	5.0801
12	2466.9551	2464.6399	-2.3152
13	3033.2791	3037.8601	4.5811
14	2254.6340	2261.7200	7.0859
15	1120.4385	1133.6000	13.1615
16	610.1624	608.1020	-2.0604
17	974.7417	968.4780	-6.2637
18	2088.7920	2074.3501	-14.4419
19	2853.0112	2843.4099	-9.6013
20	3210.1338	3202.3101	-7.8237
21	3064.3601	3059.3000	-5.0601
22	3451.8816	3454.7500	2.8684
23	3357.2061	3352.4500	-4.7561
24	4252.9980	4259.7598	6.7617
25	4173.1909	4179.7798	6.5889
26	4521.8301	4524.8198	2.9897
27	3175.9929	3176.6699	0.6770
28	924.6970	927.3100	2.6130
29	557.2224	558.9320	1.7097
30	380.3488	380.9310	0.5822

The correctness of the network model is verified through Figure 8, where each circular dot represents a sample in the testing pattern. These samples are not randomly distributed around the diagonal. They tend to be on the slant in some areas and below in others. The deviation from the diagonal of the samples is not too large, indicating that the network is highly effective. Figure 9 shows a GIS map that predicts the situation of rice diseases in the coming week in

Table 4: The results of trained neural networks on a testing data set using BP (cont.)

No.	Area infected with pests (in hectares)		Error (e_t=d_t-y_t)
	Actual output (dt)	Network output (y_t)	
309	7013.4478	7018.1899	4.7422
310	6324.8945	6307.2798	-17.6147
311	3849.1008	3865.1399	16.0391
312	1536.9921	1538.7200	1.7279
313	631.7750	626.2660	-5.5090
314	552.5786	520.1650	-32.4136
315	242.9010	242.9690	0.0680
316	3513.7822	3525.5801	11.7979
317	5668.2588	5642.0698	-26.1890
318	4242.8413	4246.7100	3.8687
319	4266.2949	4276.0601	9.7651
320	4604.6484	4616.2798	11.6313
321	4831.8579	4838.2500	6.3921
322	5450.6797	5446.8599	-3.8198
323	6520.3916	6538.1201	17.7285
324	6056.1436	6048.3501	-7.7935
325	2742.1736	2745.7800	3.6064
326	757.4758	761.2940	3.8182
327	448.6403	448.2700	-0.3703
328	400.1339	398.3690	-1.7649
329	196.9795	206.2930	9.3135
		RMSE	**10.649**

An Giang Province. It allows users to observe directly on the GIS map the pest situation for the next week in 11 districts in the Province and the infected area (in hectares) of each pest in each province. Each color on the pie chart will represent the percentage of infected area of each pest in each district.

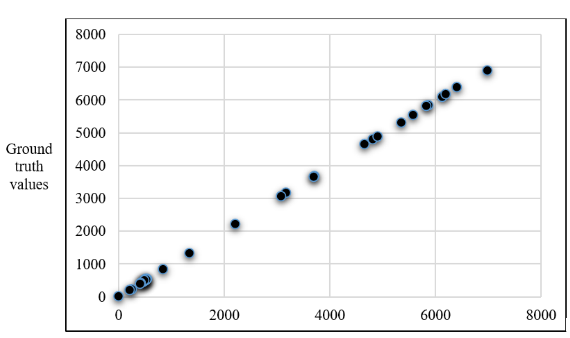

Fig. 8: Verify the correctness of the model

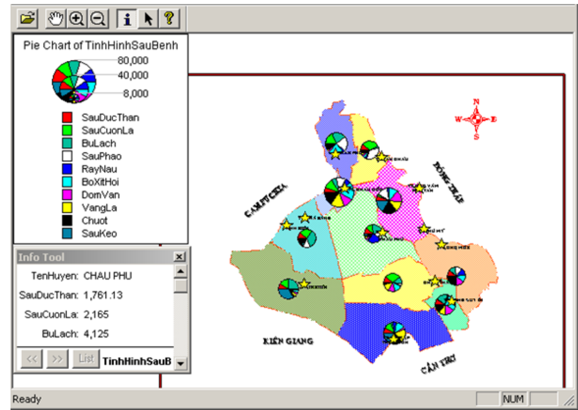

Fig. 9: GIS map predicting rice disease in An Giang Province in the coming week

IV. CONCLUSION AND RECOMMENDATIONS

The forecasting method has yielded certain results, and it will provide many benefits to farmers in the future, thanks to the enthusiastic support of An Giang Province's Plant Protection Department and An Giang University's Faculty of Agriculture and Natural Resources. The system has been tested in the rice fields of 11 districts in An Giang Province. Experimental results show that our system is currently producing promising results. Due to the increasing complexity of rice diseases, this system needs more time to research and improve. This paper has presented a combination of an artificial neural network model and a genetic algorithm to predict the rice disease situation in An Giang Province. By applying this prediction model, the prediction has achieved better advantages compared to other traditional statistical methods. Our next task is to perfect this system and research other influential factors that can help model predict rice yields, diagnose rice diseases, and propose appropriate prevention and treatments.

ACKNOWLEDGMENTS

This research was supported by the An Giang University, Vietnam under the research grant 23.01.CT, and the technical staff and agricultural experts from An Giang University, and Vietnam National University in Ho Chi Minh City, Vietnam.

REFERENCES

[1] McCulloch W, Pitts W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*. 1990;52(1–2): 99–115. https://doi.org/10.1007/BF02459570.

[2] Zhao G, Yu H, Wang G, Sui Y, Zhang L. Study on the detection and warning system of rice disease based on the GIS and IOT in Jilin Province. In: *Computer and Computing Technologies in Agriculture VIII: 8th IFIP WG 5.14 International Conference, CCTA 2014, Beijing, China*. Springer International Publishing; 2015. p.337–343. https://dx.doi.org/10.1007/978-3-319-19620-6_38.

[3] Nguyen NBP, Nguyen NH, Tran DT, Tran PTM, Pham TG, Nguyen TM. Assessing damages of agricultural land due to flooding in a lagoon region based on remote sensing and GIS: case study of the Quang Dien District, Thua Thien Hue Province, central Vietnam. *Journal of Vietnamese Environment*. 2020;12(2): 100–107. https://doi.org/10.13141/jve.vol12.no2.pp100-107.

[4] Al-Ismaili AM. GIS and remote sensing techniques in Controlled Environment Agriculture: A review. *Journal of Agricultural and Marine Sciences [JAMS]*. 2021;26(2): 10–23. https://doi.org/10.53541/jams.vol26iss2pp10-23.

[5] Plata IT, Panganiban EB, Alado DB, Taracatac AC, Bartolome BB, Labuanan FRE. Drone-based Geographical Information System (GIS) Mapping of Cassava Pythoplasma Disease (CPD) for Precision Agriculture. *International Journal of Emerging Technology and Advanced Engineering*. 2022;12(2): 1–9. https://doi.org/10.46338/ijetae0222_01.

[6] Vedaldi A, Zisserman A. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2012;34(3): 480–492. http://dx.doi.org/10.1109/TPAMI.2011.153.

[7] Visa S, Ralescu A. Issues in mining imbalanced data sets – a review paper. In: *Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference, MAICS-2005, Dayton, 16th – 17th April 2005. p.67–73*. https://www.researchgate.net/publication/228386653_Issues_in_Mining_Imbalanced_Data_Sets_-_A_Review_Paper [Accessed 15th January 2023].

[8] Wang C, Yan S, Zhang HJ. Large scale natural image classification by sparsity exploration. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE; 2009. p.3709–3712. http://dx.doi.org/10.1109/ICASSP.2009.4960432.

[9] Dreyfus SE. Artificial neural networks, back propagation, and the Kelley-Bryson gradient procedure. *Journal of Guidance, Control, and Dynamics*. 1990;13: 926–928. https://doi.org/10.2514/3.25422.

[10] Polak E. *Optimization: Algorithms and Consistent Approximations*. New York: Springer. 1997.

[11] Turing AM. Computing Machinery and Intelligence. In: Epstein R, Roberts G, Beber G. (eds) *Parsing the Turing Test*. Dordrecht: Springer; 2009. https://doi.org/10.1007/978-1-4020-6710-5_3.

[12] Back T. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. New York: Oxford University Press; 1996.

[13] Ministry of Agriculture and Rural Development, Vietnam. *National technical regulation on surveillance method of rice pests. QCVN 01-38: 2010/BNNPTNT*. Ministry of Agriculture & Rural Development; 2010.

[14] Nguyen Manh Chinh. *Plant Doctor's Handbook. [Cẩm nang bác sĩ cây trồng]*. Hanoi, Vietnam: Agriculture Publishing House; 2012.

[15] An Giang Plant Protection Joint Stock. *Handbook of rice farmers need to know. [Sổ tay người nông dân trồng rau cần biết]*. An Giang, Vietnam: An Giang Plant Protection Joint Stock; 2020.

[16] Montana DJ, Davis L. Training feedforward neural networks using genetic algorithms. In: *Proceedings of the 11th International Joint Conference on Artificial Intelligence, San Mateo*. United States: Morgan Kaufmann Publishers; 1989. p.762–767.

[17] Whitley D, Starkweather T, Bogart C. Genetic algorithms and neural networks: Optimizing connections and connectivity. *Parallel Computing*. 1990;14(3): 347–361. https://doi.org/10.1016/0167-8191(90)90086-O.