IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.17, No.3, July 2023, pp. 301~312
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: https://doi.org/10.22146/ijccs.85584

Predictive Analysis of Rice Pest Distribution in Bali Province Using Backpropagation Neural Network

I Kadek Agus Dwipayana[1], Putu Sugiartawan*[2]
[12]Program Studi Teknik Informatika, Institut Bisnis dan Teknologi Indonesia, Bali
e-mail: **[1]agoesdwipayana03@gmail.com**, [2] putu.sugiartawan@stiki-indonesia.ac.id

Abstract

The distribution of pests in rice plants results in significant losses in production and damage to rice plants for farmers, seen from data on the area of rice borer attacks in the province of Bali in Tabanan district. Therefore, by predicting the distribution of rice pests, we can know the pattern of pest attacks to anticipate them. Predicting can provide accuracy and error values through the test results. One of the prediction models is BPNN, where BPNN's advantages for solving complex problems are very suitable for use where large amounts of data are involved and many input/output variables. Backpropagation includes supervised learning, meaning it can learn from labeled examples and make accurate predictions on new, unlabeled data. Prediction of the distribution of rice pests in the future can be made with the predicted value of the Epoch value used so that the backpropagation method can determine the pattern of data on the distribution of borer pests in all districts of the Province of Bali by using the pattern of distribution of rice pests in Tabanan district. Second, optimal learning from the neural network architecture obtained a Learning Rate value 0.001. For the second, the results of the Hidden Layer unit testing experiment obtained a total of three units. The Momentum testing experiment got a Momentum number of 0.0001

Keywords: *Rice Pest Prediction, Backpropagation Analysis, Machine Learning*

1. INTRODUCTION

Diseases in rice plants are the beginning of difficulties for farmers because if they affect the plants, they will cause significant losses for farmers. Pests on paddy rice plants can indirectly cause damage to land seen from the area of attack and a significant decrease in production [1]. Especially in the case of Bali Province in Tabanan Regency, where seen from the data obtained, the extent of the rice borer attack has a significant impact on farmers' land and influences production. Indonesian farmers are still haunted by the rice borer pest, which is the pest that most often causes heavy damage to rice plants and attacks rice plants during the nursery, vegetative phase, and generative phase [2]. This threatens agricultural productivity, so we need a prediction of the area of attack by borer pests on rice plants to minimize the impact of crop failure.

To predict future conditions in this case, namely predicting borer attacks, data is needed, which will later be processed using a predictive method; the use of machine learning methods to predict is part of artificial intelligence (Artificial Intelligence), which is a method for exploring the construction of algorithms that can study and make predictions from data [3]. One branch of machine learning is the Artificial Neural Network (ANN) method, also known as Artificial Neural

Received June 12th,2023; Revised July 26th, 2023; Accepted July 29th, 2023

Network (ANN). ANN is widely used in data mining; this method can be used in clustering, regression, classification, time series forecasting, and visualization [4]. The backpropagation method is also included in the ANN model used for forecasting; this method is part of supervised learning which is usually used for layers to determine the weights connected to the neurons in the hidden layer [5].

From the description above, the background of this research is how artificial neural networks with the Backpropagation method can help predict the area of rice borer attack using BPNN. In addition to providing helpful information to other researchers, it is also expected to know the accuracy and error of the prediction case for the area of rice borer attack. To be able to help the farmers advise in the future.

2. METHODS

2.1 Borer Pests

The extent of pest attacks on rice farming is often detrimental, as seen through the harvest of farmers who find significant deficiencies; this is masterminded by OPT (Plant Disturbing Organisms). OPT has various plant pests, especially rice plants, which are staple foods in Indonesia [6]. One of the pests that often disturb rice plants is borer pests; these pests are destructive and are very detrimental to farmers when they are in the larval (caterpillar) stage. This rice stems borer attacks rice plants at any stage, from seedling to maturity. The rice borer attacks in two phases, namely the vegetative and generative phases, which cause the shoots to wither, turn brown, dry, and die. Moreover, after the panicles appear white and empty [7], figure 1 shows the pest at ricefield [8].

Figure 1 Borer pest at ricefield [8]

2.2 Machine Learning

Machine learning can be defined as computer applications and mathematical algorithms that are adopted using learning data derived from data and generating predictions in the future [8]. Machine learning is an AI approach extensively utilized to replicate or mimic human behavior, enabling problem-solving and automation. As its name suggests, machine learning attempts to emulate how humans or intelligent beings learn and generalize. There are at least two primary applications in machine learning: classification and prediction. The central feature of machine learning is the training or learning process, which relies on data to learn, commonly referred to as training data. Classification is a machine learning technique employed by machines to categorize and group objects based on specific characteristics, akin to how humans discern and differentiate between different objects.

2.3 Backpropagation Neural Network Architecture

Backpropagation is one of the training models of an Artificial Neural Network to get a balance between the network's ability to recognize the patterns used in the training stage and the network's ability to respond correctly to input patterns that are similar to the patterns used in the training stage [9]. ANN (Artificial Neural Network) Backpropagation architecture consists of three layers: the input layer, the hidden layer, and the output layer. At the input layer, the calculation process has not yet occurred, but at the input layer, the X input signal is sent to the hidden layer. In the hidden layer and output layer, a calculation process occurs depending on the weight and bias of each neuron. The result calculates the output value of the hidden and output layers based on the activation function used [10]. In the hidden layer and output layer, a calculation process occurs depending on the weight and bias of each neuron. The result calculates the output value of the hidden and output layers based on the activation function used.

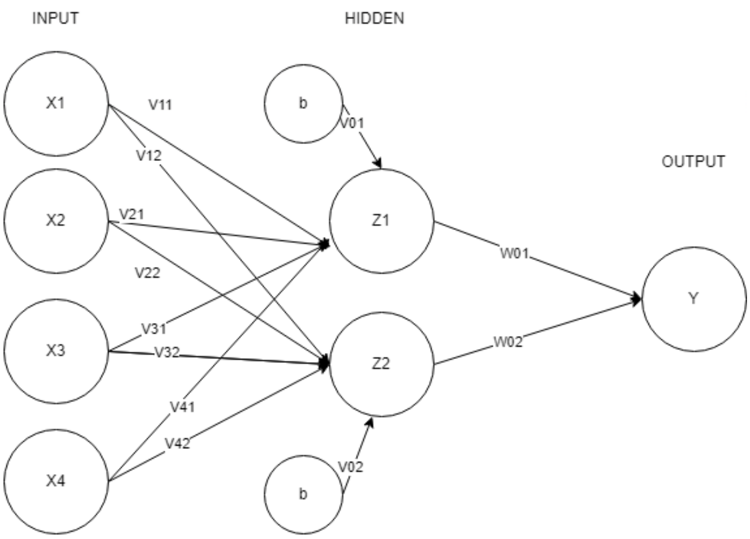

Figure 2 Backpropagation Neural *Network Architecture*

The description of each layer in the Backpropagation Neural Network (BPNN) is as follows [11] :

1. Input Layer
   The input layer is the link where the external environment provides a pattern into the neural network. Once a pattern is given to the input layer, the output layer will provide another pattern
2. Hidden Layer

This layer consists of several layers and at least one layer. The hidden layer consists of neurons from the first to the z-hidden neuron. Determining the number of neurons in the hidden layer is a significant part of the neural network architecture. Several rules can be used to determine the number of neurons to be used in the hidden layer, and the number of hidden neurons from 2 to 9 can produce good results in the network. The number of hidden neurons can be up to infinity (~). Several rules can be used to determine the number of neurons in the hidden layer:

a. The number of hidden neurons must be between the size of the input layer and the output layer.
b. The number of hidden neurons must be 2⁄3 of the size of the input layer, plus the size of the output layer.
c. The number of hidden neurons must be less than twice the number of the input layer.

3. Output Layer
This layer consists of several output neurons and consists of one layer. The output layer of a neural network is the actual pattern given by its external environment. Patterns given by the output layer can be traced back to the input layer. The number of output neurons depends on the type and performance of the neural network itself.

2.3.1 Backpropagation Algorithm

In the backpropagation method training, there are three phases. The first phase is the forward phase, which is when the network calculates the output data; the second phase is the reverse phase if there is an error (the difference between the desired output target and the output value obtained), and the third phase is weight modification to reduce network generated errors, Training algorithm for a network with one hidden layer (with a binary sigmoid activation function) is as follows:
Step 0. Initialize all weights with small random numbers
Step 1. If the termination conditions are not met, proceed to steps 2-9.
Step 2. For each pair of training data, perform steps 3-8.
Phase I: Forward Propagation.
Step 3. Each input layer (, i = 1,2,....,n) receives an input signal and forwards it to the hidden layer
Step 4. Calculate all the outputs in the hidden layer (j=1,2,…,p).

$$Z\_in_j = v_{0j} + \sum_{i=1}^{n} x_i \; v_{ij} \tag{1}$$

Apply the activation function to calculate the output signal

$$z_j = f\left(z_{in_j}\right) = \frac{1}{1 + e^{-z\_in_j}} \tag{2}$$

Send signal to all output layers

Step 5. Calculate all outputs in the output layer (k=1,2,…,m)

$$y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j \; w_{jk} \tag{3}$$

Apply the activation function to calculate the output signal

$$y_k = f(y_{in_k}) = \frac{1}{1 + e^{-y\_in_k}} \tag{4}$$

Phase II: Backpropagation
Step 6. Calculate the factor δ in the output layer based on the errors in each output layer (k=1,2,…,m).

$$\delta_k = (t_k - y_k)\, f' \;\; (y\_in_k) = (t_k - y_k)\, y_k \,(1 - y_k) \tag{5}$$

δ_k is the unit of error that will be used in changing the weight of the layer below (step 7). Calculate the weight change rate (which will be used to change the weight) with the acceleration rate α.

$$\Delta w_{kj}=\alpha\ \delta_k\ z_j; \quad (k=1,2,\ldots,m;\ j=0,1,\ldots,p) \tag{6}$$

Calculate the change in bias (which will be used to change the weight)

$$\Delta w_{0k}=\alpha\ \delta_k; \qquad (k=1,2,\ldots,m) \tag{7}$$

And send it δ_k to the layer below it

Step 7. Calculate the factor δ in the hidden layer based on the error in each hidden layer unit (j=1,2,…,p).

$$\delta\_net_j = \sum_{k=1}^{m} \delta_k\ w_{jk} \tag{8}$$

Factor δ hidden layer:

$$\delta_j = \delta\_net_j\ f'\ (z\_net_j) = \delta\_net_j\ z_j\ (1-z_j) \tag{9}$$

Calculate the change in weight (which is used later to change the weight).

$$\Delta v_{ij}=\alpha\ \delta_j\ x_i; \qquad (j=1,2,\ldots,p;\ i=0,1,\ldots,n) \tag{10}$$

Phase III: Change in Weight.

Step 8. Calculate all changes in weight. Change the weight of the line going to the output layer

$$w_{kj}\ (baru) = w_{kj}\ (lama) + \Delta\ w_{kj},\ (k=1,2,\ldots,m\ ;\ j=0,1,\ldots,p) \tag{11}$$

Change the weight of the line going to the hidden layer.

$$v_{ij}\ (baru) = v_{ij}\ (lama) + \Delta\ v_{ij} \qquad (j=1,\ldots,;\ i=1,2,\ldots,n) \tag{12}$$

Step 9. The training process stops.

After training is complete, the network can be used for pattern recognition. In this case, only forward propagation (steps 4 and 5) determines the network output. If the activation function is not binary sigmoid, then steps 4 and 5 must be adjusted. Likewise, the derivatives in steps 6 and 7.

In some cases the training required required many iterations, thus making the training process long. To speed up the iteration can be done with the parameter α or Learning Rate. The value of α lies between 0 and 1 ($0 \le \alpha \le 1$). If the price α is greater, then the number of iterations used is less. This causes the correct pattern to be broken so that understanding becomes slow.

2.4 Sliding Windows Technique

One of the approaches from the standard model of Neural Networks in doing prediction of time series data is by sliding windows. This technique drives the f function in the Neural Network and uses the feedforward function as in the MLP, RBF, or cascade correlation architecture, by using a single set N tuples as input and one output as network target value. In the sliding window illustration, in this case, three-time stages are used. The time span used is three days, where the previous three days of data are used to predict future data results [12].

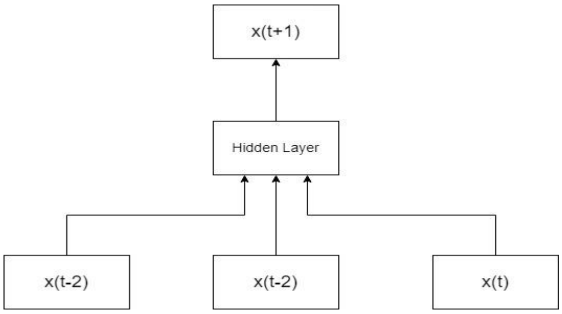

Figure 3 Illustration of Sliding Windows

2.5 Pre Processing Data

In this stage, from the data source obtained from the Faculty of Agriculture, Udayana University Pest Control Laboratory, it is known that the extent of rice borer attack from 2012-2021 Bali Province in Tabanan district will later be divided into two types of data, namely training data and training data. Test (testing) as input data. In order to be able to predict, this stage is essential, which aims to prepare data. After that, the normalization stage is because the hidden layer uses the binary sigmoid activation function in the hidden layer for its output layer. Normalization is essential to simplify calculations and get more accurate prediction results [13]. The following is an example of data on the area of attack by borer in Tabanan district for 2019 – 2021.

Table 1 Data on Pest Attack Area in Tabanan Regency 2019 - 2021

Area of attack per month (ha) Borer Pest District Tabanan 2019 - 2021			
Moon	**Attack area (ha) 2019**	**Attack area (ha) 2020**	**Attack area (ha) 2021**
01/01	8,00	2,00	10,00
01/02	10,00	11,00	17,00
01/03	9,00	13,00	18,00
01/04	8,00	18,00	27,00
01/05	9,00	19,00	147,00
01/06	11,00	66,00	46,00
01/07	10,00	45,00	17,00
01/08	7,00	68,00	5,00
01/09	7,00	4,00	1,00
01/10	5,00	2,00	31,00
01/11	11,00	13,00	15,00
01/12	3,00	10,00	20,00

2.6 Data Normalization

Data normalization is done to minimize errors, and the dataset, known in Table 1, will be normalized by changing the actual data to a range interval value [0,1]. The normalization technique used is min-max scaling. The formula for normalizing min-max scaling is:

$$X' = \frac{(x - min_x)}{max_x - min_x} \tag{13}$$

Keterangan:
X : Normalized data
X' : Data After normalized
min : The minimum value of the entire data
max : The maximum value of the entire data

In table 1 above, we input it into the normalization formula as follows:

Normalization on January 1, 2019:

$$X' = \frac{(x - min_x)}{max_x - min_x}$$

X = 8,00

max = 11,00

min = 3,00

$$X' = \frac{(8{,}00 - 3{,}00)}{11{,}00 - 3{,}00}$$

Jadi X' = 0,625.

Normalization of this data will be carried out onwards referring to the dataset in Table 1, here taking an example from January 1 2019 to December 1 2021, so it will produce the following table:

Table 2 Attack Area Data after Normalization

Area of attack per month (ha) Borer Pests in Tabanan Regency 2019 – 2021 after normalization			
Moon	**2019**	**2020**	**2021**
01/01	0,625	0	0,061
01/02	0,875	0,136	0,109
01/03	0,75	0,166	0,116
01/04	0,625	0,242	0,178
01/05	0,75	0,257	1
01/06	1	0,969	0,308
01/07	0,875	0,651	0,109
01/08	0,5	1	0,027
01/09	0,5	0,030	0
01/10	0,25	0	0,205
01/11	1	0,166	0,095
01/12	0	0,121	0,130

3. RESULTS AND DISCUSSION

Data for this study were retrieved from the Bali Province Agriculture Office 2012 – 2021; the data used is rice plant borer data. The coverage area taken for research covers all districts in the province of Bali. Prediction of the distribution of rice pests (borers) using the Artificial Neural Network approach with the Backpropagation Neural Network algorithm; this algorithm can recognize patterns of distribution of rice pests with a certain level of accuracy, improve the prediction accuracy of the backpropagation algorithm using several test options including optimal learning, momentum testing, testing the level of learning the amount of learning that is normalization is done to minimize errors, and the dataset, known in Table 1, will be normalized by changing the actual data to a range interval value [0,1]. The normalization technique used is min-max scaling. The formula for normalizing min-max scaling is.

3.1 Prepocesing Data

In Preprocessing this data changes the form of time series data to supervised data. Changes to the data using the Sliding Windows Technique, this technique uses a window size value of two. As shown in Table 6.

Table 3 Example of Sliding Windows Implementation

Waktu	Data	X1	X2	Y
01/01/2012	0	0	2	11
01/02/2012	2	2	11	3
01/03/2012	11	11	3	4
01/04/2012	3	3	4	20
01/05/2012	4	4	20	17
	…	…	…	…
n				

3.2 Data Normalization

Data Normalization aims to reduce the value of the data interval so that the range of test values becomes small. Data normalization uses the min max algorithm, as shown in table 7..

Table 4 Data Normalization Example

Data	X1	X2	Y	X1’	X2’	Y’
1	0	2	11	0	0,01	0,07
2	2	11	3	0,01	0,07	0,02
3	11	3	4	0,07	0,02	0,02
4	3	4	20	0,02	0,02	0,13
5	4	20	17	0,02	0,13	0,11
	…	…	…	…	…	…
n						

3.3 Testing with the Number of Units in the Hidden Layer

The backpropagation network architecture and the best parameters obtained from training with windows size two and used for testing are shown in Figure 3 and Table 8.

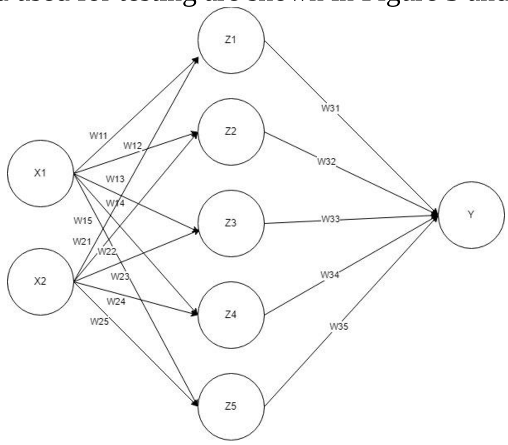

Figure 4 BPNN Architecture BPNN Test Windows Size Two

Table 5 Testing Process Parameters Using Number of Neurons With Windows Size Two

Parameter	**Value**	**Information**
Input	2	X_1 dan X_2
Target *Error* (SSE)	0,00001	The target process stops
Learning Rate	0,001	System learning speed
Momentum	0,001	Konstanta momentum
Amount *Hidden*	1	The number of hidden layers
Maximum *Epoch*	10.000	The number of iterations or repetitions

Parameter	Value	Information
Target Output	1	Y
Tolerance	0,1	Prediction Tolerance Value of Borer Pest Attack Area
Transfer Function	*Sigmoid Biner*	
Bobot	Random (0,1)	

The results of testing experiments with the number of units in the hidden layer found that with the unit value of the hidden layer three (3) the result was the fastest time.

Table 6 Hidden Layer Unit Value Experiment Results

Test	Unit Value	SSE	Time
Test 1	3	2.19191861152649	18 sec
Test 2	5	2.19749498367310	21 sec
Test 3	7	2.19462394714355	20 sec
Test 4	10	2.19679689407349	20 sec
Test 5	13	2.19752526283264	26 sec

3.4 Testing with Learning Rate Value

Testing the neural network backpropagation algorithm for the best parameters obtained from training with windows size two and used for testing is shown in Table 10.

Table 7 Parameters for Testing the Learning Rate Value

Parameter	Value	Information
Input	2	X_1 dan X_2
Target *Error* (SSE)	0,00001	The target process stops
Momentum	0,001	Konstanta momentum
Amount *Hidden*	1	The number of hidden layers
Number of Neurons	3	Number of Neurons in Hidden
Maximum *Epoch*	10.000	The number of iterations or repetitions
Target Output	1	Y
Tolerance	0,1	Prediction Tolerance Value of Borer Pest Attack Area
Transfer Function	*Sigmoid Biner*	
Bobot	Random (0,1)	

The results of testing experiments with a certain Learning Rate value get the result that with a Learning Rate value of 0.001 the result is the fastest time and the smallest SSE value.

Table 8 Learning Rate Value Experiment Results.

Test Time	Learning Rate	SSE	Test Time
Test 1	0,1	1.93761515617371	1 min 53 sec
Test 2	0,01	2.19216489791870	1 min 41 sec
Test 3	0,05	2.01736760139465	1 min 49 sec
Test 4	0,001	2.18856549263000	1 min 19 sec
Test 5	0,0001	2.20707249641418	1 min 49 sec

3.5 Testing with Momentum

Testing the backpropagation neural network algorithm and the best parameters obtained from training with windows size two and used for testing is shown in Table 12..

Table 9 Testing Parameters for Momentum Values

Parameter	**Value**	**Information**
Input	2	X_1 dan X_2
Target *Error* (SSE)	0,00001	The target process stops
Learning Rate	0,001	System learning speed
Amount *Hidden*	1	Number of Neurons in Hidden
Number of Neurons	3	The number of iterations or repetitions
Maximum *Epoch*	10.000	Y
Target Output	1	Prediction Tolerance Value of Borer Pest Attack Area
Tolerance	0,1	The number of hidden layers
Transfer Function	*Sigmoid Biner*	
Bobot	Random (0,1)	

The results of the testing experiment with a certain Momentum value found that with a Momentum value of 0.0001 the result was the fastest time and the smallest SSE value.Tabel 1 Hasil Percobaan nilai Momentum.

Table 10 Momentum value testing parameters

Test	**Momentum**	**SSE**	**Time**
Test 1	0,1	2.19757485389709	18 sec
Test 2	0,01	2.19420647621155	18 sec
Test 3	0,05	2.19285535812378	17 sec
Test 4	0,001	2.19754958152771	20 sec
Test 5	0,0001	2.19270682334900	17 sec

3.6 Data Testing Results Testing

After testing the parameters, namely: Hidden Layer, Learning Rate, and Momentum and using windows size two values, the best results were obtained from the results of each experiment. The best results from each of these experiments will be entered into the parameters to be tested again as shown in table 14.

Table 11 Test Parameters of each Best value

Parameter	**Value**	**Information**
Input	2	X_1 dan X_2
Target *Error* (SSE)	0,00001	The target process stops
Learning Rate	0,001	System learning speed
Momentum	0,0001	The number of hidden layers
Amount *Hidden*	1	Number of Neurons in Hidden
Number of Neurons	3	The number of iterations or repetitions
Maximum *Epoch*	1.000.000	Y
Target Output	1	Prediction Tolerance Value of Borer Pest Attack Area
Tolerance	0,1	The number of hidden layers
Transfer Function	*Sigmoid Biner*	
Bobot	Random (0,1)	

The results of testing the best value of each parameter from the Hidden Layer Neuron, Learning Rate, and Momentum and using an epoch of 1,000,000, found the following results:

Table 12 Best Value Test Results for each Parameter

Learning Rate	**Momentum**	**SSE**	**epoch**	**Time**
0,001	0,0001	1.93582129478455	1.000.000	30 min 39 sec

3.7 Prediction Results

The Backpropagation method can also be used to calculate predictions by calling the results of prediction testing data testing which will then be retrained to find prediction results in the next few months as follows.

The prediction results for Tabanan Regency are as follows:

Table 13 Tabanan Regency Prediction Results

X1	X2	Yout
0.3129252	0.1156463	0.09357321
0.1156463	0.0340136	0.07755776
0.0340136	0.0068027	0.07291491
0.0068027	0.2108844	0.1234654
0.2108844	0.1020408	0.09160128

4. CONCLUSIONS

Several conclusions can be drawn from this research. The first prediction of the distribution of rice pests in the future can be done with the predicted value of the Epoch value used. So that the backpropagation method can determine the pattern of data on the distribution of borer pests in all districts of the Province of Bali by using the pattern of distribution of rice pests in the Tabanan district. Second, optimal learning from the neural network architecture obtained a Learning Rate value of 0.001. For the second, the results of the Hidden Layer unit testing experiment obtained a total of three units. The fourth after doing the Momentum testing experiment got a Momentum number of 0.0001. Suggestions that can be given can use the RNN (Recurrent Neural Network) method to compare the results of accuracy.

REFERENCES

[1] M. Sarumaha and M. Pracaya, “Identifikasi serangga hama pada tanaman padi di desa bawolowalani,” *J. Educ. Dev.*, vol. 8, no. 3, pp. 86–91, 2020.

[2] M. Syamsiah and A. F. Dikri, “PENGGUNAAN BEBERAPA PERANGKAP UNTUK MENGENDALIKAN HAMA PENGGEREK BATANG PADI PANDANWANGI (Oryza sativa var. aromatic) PADA FASE GENERATIF,” *Pro-STek*, vol. 1, no. 1, p. 51, 2020, doi: 10.35194/prs.v1i1.821.

[3] P. Ongsulee, “Artificial intelligence, machine learning and deep learning,” *Int. Conf. ICT Knowl. Eng.*, pp. 1–6, Jan. 2018, doi: 10.1109/ICTKE.2017.8259629.

[4] S. A. Salloum, M. Alshurideh, A. Elnagar, and K. Shaalan, “Mining in Educational Data: Review and Future Directions,” *Adv. Intell. Syst. Comput.*, vol. 1153 AISC, pp. 92–102, 2020, doi: 10.1007/978-3-030-44289-7_9.

[5] P. Wijaya, R. W. Sembiring, and S. S, “Analisis Metode Backpropagation Memprediksi Penerimaan Santri/Wati di Pondok Pesantren Modern Al-Kautsar,” *Jurasik (Jurnal Ris. Sist. Inf. dan Tek. Inform.*, vol. 6, no. 1, p. 140, 2021, doi: 10.30645/jurasik.v6i1.278.

[6] S. Sudewi, A. Ala, B. Baharuddin, and M. F. BDR, “Keragaman Organisme Pengganggu Tanaman (OPT) pada Tanaman Padi Varietas Unggul Baru (VUB) dan Varietas Lokal

pada Percobaan Semi Lapangan," *Agrikultura*, vol. 31, no. 1, p. 15, 2020, doi: 10.24198/agrikultura.v31i1.25046.

[7] M. Suarsana, P. Parmila, P. S. Wahyuni, and I. G. M. Suarmika, "Pengaruh Serangan Hama Penggerek Batang dan Penyakit Tungro terhadap Produktivitas Sembilan Varietas Padi di Lokapaksa, Bali," *Agro Bali Agric. J.*, vol. 3, no. 1, pp. 84–90, 2020, doi: 10.37637/ab.v3i1.461.

[8] A. Roihan, P. A. Sunarya, and A. S. Rafika, "Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper," *IJCIT (Indonesian J. Comput. Inf. Technol.*, vol. 5, no. 1, pp. 75–82, 2020, doi: 10.31294/ijcit.v5i1.7951.

[9] M. N. Fadilah, A. Yusuf, and N. Huda, "Prediksi Beban Listrik Di Kota Banjarbaru Menggunakan Jaringan Syaraf Tiruan Backpropagation," *J. Mat. Murni Dan Terap. Epsil.*, vol. 14, no. 2, p. 81, 2021, doi: 10.20527/epsilon.v14i2.2961.

[10] G. Dewantoro and J. N. Sukamto, "Implementasi Kendali PID Menggunakan Jaringan Syaraf Tiruan Backpropagation," *Elkha*, vol. 11, no. 1, p. 12, 2019, doi: 10.26418/elkha.v11i1.29959.

[11] J. R. Prabowo, R. Santoso, and H. Yasin, "IMPLEMENTASI JARINGAN SYARAF TIRUAN BACKPROPAGATION DENGAN ALGORITMA CONJUGATE GRADIENT UNTUK KLASIFIKASI KONDISI RUMAH (Studi Kasus di Kabupaten Cilacap Tahun 2018)," *J. Gaussian*, vol. 9, no. 1, pp. 41–49, 2020, doi: 10.14710/j.gauss.v9i1.27522.

[12] A. E. Radho, P. Sugiartawan, and G. A. Santiago, "Prediksi Jumlah Kasus COVID-19 Menggunakan Teknik Sliding Wondows dengan Metode BPNN," *J. Sist. Inf. dan Komput. Terap. Indones.*, vol. 4, no. 1, pp. 12–23, 2022, doi: 10.33173/jsikti.123.

[13] H. Putra and N. Ulfa Walmi, "Penerapan Prediksi Produksi Padi Menggunakan Artificial Neural Network Algoritma Backpropagation," *J. Nas. Teknol. dan Sist. Inf.*, vol. 6, no. 2, pp. 100–107, 2020, doi: 10.25077/teknosi.v6i2.2020.100-107.