Classification of Bidikmisi Scholarship Acceptance using Neural Network Based on Hybrid Method of Genetic Algorithm

N Cahyani[1*], S S Pangastuti[2*], K Fithriasari[3], Irhamah[4], N Iriawan[5]

[1]Departemen Statistika, Universitas Nahdlatul Ulama Sunan Giri, Bojonegoro, Indonesia
[2]Departemen Statistika, Universitas Padjadjaran, Sumedang, Indonesia
[3,4,5]Departemen Statistika, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

[1*]nitacahyani@unugiri.ac.id, [2*]sinta.septi@unpad.ac.id

Abstract. A Neural network is a series of algorithms that endeavours to recognize underlying relationships in a set of data through processes that mimic the way human brains operate. In the case of classification, this method can provide a fit model through various factors, such as the variety of the optimal number of hidden nodes, the variety of relevant input variables, and the selection of optimal connection weights. One popular method to achieve the optimal selection of connection weights is using a Genetic Algorithm (GA), the basic concept is to iterate over Darwin's evolution. This research presents the Neural Network method with the Backpropagation Neural Network (BPNN) and the combined method of BPNN with GA, where GA is used to initialize and optimize the connection weight of BPNN. Based on accuracy value, the BPNN method combined with GA provides better classification, which is 90.51%, in the case of Bidikmisi Scholarship classification in East Java.

1. Introduction

Neural network and *genetic algorithms* are two different methods used for learning and optimization. By imitating the way human brains operate, a *neural network* (NN) is a series of algorithms that endeavour to recognize the essential relationship in a set of data [1]. A neural network algorithm refers to a network of neurons which is a mathematical function used to collect and to classify data from a given model [2]. Proper training of a neural network is the most important aspect of making a reliable model. One of the training methods for neural network is "*Back-propagation*", or we can call it *Back-propagation Network* (BPN).

Back-propagation network uses the gradient based approach that trains slowly with many local minimum [3]. Therefore, instead of using gradient-based learning technique for optimization, one may apply the commonly used optimization method such as genetic algorithm. *Genetic algorithm* (GA) is an optimization technique based on the principles of genetics and natural selection. The population in genetic algorithm is made up of many individuals who develop accordance with particular selection rules by maximizing fitness [4]. A series of genetic algorithm starts with a preliminary population, then this algorithm creates a new population using selection, crossover, and mutation operations. This algorithm takes the preliminary population as input and selects a fitness function. It helps the algorithm to generate an optimal solution. Genetic algorithm continues and develops the population through selection, crossover and mutation operations. It generates various populations until it satisfies the optimization constraints. As genetic algorithm is a stochastic general search method, capable of effectively exploring large search spaces, and has been used with back-propagation network for determining the various parameters such as number of hidden layers and hidden nodes [5], the possible combinations of the two methods are basically three: the first one is back-propagation network based genetic algorithm (BPN-GA) [6,7,8], the second is genetic algorithm based back-propagation network (GA-BPN) [9], and the third one uses the genetic algorithm for best training set generation for the back-propagation network [10].

Based on previous description, this study will use the back-propagation network method and then use the genetic algorithm on it to optimize the weight of back-propagation network training. The goal is to compare the two methods based on their accuracy. The results obtained are expected to explain the accuracy of the classification of Bidikmisi scholarship recipients in East Java. This paper structure as follows. In section two, a brief explanation about the data and the algorithms uses, also the performance evaluation is discussed. The results and discussion presented in section three, followed by conclusion in section four.

2. Methodology

2.1 Data Preparation and Pre-processing

This study was performed using the Bidikmisi data set in East Java with 12 variables and the data structure presented in Table 1. Using 10-fold cross validation to divide the data [11], the data is divided into 10 equal sized folds, hence we have 10 data subsets to evaluate the performance of the algorithms. For each of 10 data subsets, cross validation will use 9 folds for training and 1-fold for testing.

The pre-processing methods were actually changing the X-predictor variable which is categorized with the *m* category, the X-scale variable of the order is changed into a sequential value, with a interval of [0,1]. Another variable X with a nominal scale is changed to a dummy *m-1*.

Table 1. Data structure

Name	School	Predictor variables			Respon variables (Y)
		X_1	$\cdots$	X_{12}	Y
1	1	$x_{1.1}$	$\cdots$	$x_{12.1}$	y_1
2	2	$x_{1.2}$	$\cdots$	$x_{12.2}$	y_2
3	3	$x_{1.3}$	$\cdots$	$x_{12.3}$	y_3
4	4	$x_{1.4}$	$\cdots$	$x_{12.4}$	y_4
5	5	$x_{1.5}$	$\cdots$	$x_{12.5}$	y_5
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
52420	52420	$x_{1.52420}$	$\cdots$	$x_{12.52420}$	y_{52420}

2.2 Back-propagation Network (BPN)

After previous pre-processing the data, we were going to classify the Bidikmisi data set using BPN with the following steps [12]:

a. Build model BPN model by determining the number in the input layer, hidden layer and output layer. The input layer used consist of 33 neurons, the hidden layer used consists of one hidden layer. Then the number of neurons used in the hidden layer is obtained from the results of trial and error. (in this study using R software with *Neuralnet* package).
b. Determine the initial weights at the input level and the training process. In this study uses a default starting weight on the input level and hidden level (starting weight) on the *Neuralnet* package which is given a random value that are normally distributed.

c. Determine activation function to hidden layer and output layer. In this study, the sigmoid function or the activation logistic function was used, because the output form was binary data set, namely 0 (not accepted) and 1 (accepted).
d. Carry out a learning process on training data set with the BPN training method to calculate weights and bias in predicting the status of the Bidikmisi data set.
e. Calculate the classification accuracy from data training and data testing.
f. Calculate the average classification accuracy.

2.3 A Hybrid of Backpropagation and Genetic Algorithm (BPNGA)
First, the genetic algorithm is used to determine the optimal initial weight, then these weights are used in the BPN process. Here are the following steps of genetic algorithms [13]:

a. Population encoding and initialization scheme.
The main step of the GA is representation of the chromosome. For the BPN with single hidden layer with *m* nodes, *n* input nodes, and *p* output nodes the number of weights to be computed is given by *(n+p) *m*. each chromosome is made up of *(n+p) *m* number of genes.
b. Genes are represented by real number encoding method. The original population is a set of *N* chromosome, which is generated randomly. Fitness of each chromosome is computed by AUC (*area under the curve*) value.
c. Once fitness is computed for all the chromosome, the best-fit chromosomes replace the worst-fit chromosomes. Further crossover step is experimented using single point crossover, two -point crossover, and multi point crossover.
d. Finally, mutation is applied to generate the new population. The new population is given as input to compute the fitness of each chromosome, followed by process selection, reproduction, cross over and mutations to generate the next population.
e. This process is repeat until more or less all the chromosomes converge to the same fitness value.
f. The weights represented by the chromosomes in the final converged population are the optimized connection weights of the BPN.

The working of hybrid BPNGA for optimizing connection weights is shown in Figure 1.

2.4 Performance Evaluation
The final steps are to compare the BPN and BPN-GA methods and then choose the best method for Bidikmisi data set. The best method with a model that has a high average classification accuracy for training data and testing data [14].

3. Result and Discussion

In this section, we performed the Bidikmisi data set and split it into new train and test data sets using 10-fold cross validation. Build model prediction using Backpropagation Network (BPN) and Backpropagation Network with Genetic Algorithm (BPN-GA), after the model building, we calculate the performance accuracy for each model.

3.1. Backpropagation Network (BPN)

In order to create a BPN predictive model consisting of an input layer, hidden layer, and output layer, the number of neurons in each layer is determined. The number of neurons used in the input layer consists of 12 variables which are variables from the Bidikmisi data set. In this study we used one hidden layer. In determining the correct number of neurons in the hidden layer, trial and error is carried out using the binary sigmoid function in the hidden layer and the output layer. As shown in Table 3, the evaluation of the performance of the Bidikmisi classification data set using several criteria to support decision making resulted in the classification accuracy with the best performance on the number of neurons 4. Table 4 presents the average classification accuracy for training data sets 10 times in the BPN model. It is known that the average result of the classification accuracy made by the model is 78.02%.

Table 3. Performance criteria of AUC, G-Mean, dan accuracy for test data set and train data set

Number of neurons	Data Testing			Data Training		
	AUC	G-mean	Accuracy	AUC	G-mean	Accuracy
4	**0,51**	**0,01**	**78,00**	**0,51**	**0,00**	**78,02**
2	0,51	0,02	68,33	0,50	0,03	68,34
6	0,50	0,02	96,13	0,50	0,04	95,53
12	0,50	0,01	96,41	0,50	0,02	96,37
8	0,50	0,00	96,65	0,50	0,00	96,65
24	0,50	0,00	96,65	0,50	0,00	96,65

Table 4. The average classification accuracy result for the training dataset is 10 times in the BPN model

Actual Classification	Model Classification		Accuracy
	Unaccepted (0)	Accepted (1)	
Unaccepted (0)	3155	12658	78,02%
	19,95%	80,05%	
Accepted (1)	91023	364944	
	19,96%	80,04%	

3.2. Optimization of Backpropagation Neural Network Parameters Using Genetic Algorithms (BPNGA)

Steps to optimize the initial weights and bias in Backpropagation, the first thing that is done is to form an initial population consisting of 50 chromosomes, which contains the chromosome genes with 100 iteration limits, 0.8 is used in cross-crossing and a probability of 0.1 is used in the mutation. The selected chromosomes are as many as the weight used according to the number of inputs, the number of neurons in the number of hidden layers used and the number of outputs.

The second step is chromosome initialization by optimizing the Neural Network weight parameter, namely coding initialization using real value. The third step of the fitness function used is the value of AUC. The fourth step is to choose the chromosomes with the optimum fitness value which are used as parents. The fifth stage is the process of crossing over, two parents are randomly selected and used to form two new chromosomes, then elitism is carried out, namely the individual copying procedure so that the individual with the highest fitness value is not lost during the evolution process. The sixth step carries out mutations to introduce new gene elements on the chromosomes at random. The seventh stage, the stage is repeated from stage 2 to the chromosome that provides the most optimum fitness value.

After obtaining the optimal parameters for the Backpropagation training, then a model is formed to determine the classification performance of the Bidikmisi acceptance status data. Table 5 is the 10-fold cross-validation of the classification performance results from the Bidikmisi scholarship acceptance status data. Following are the results of weight optimization with 4 neurons in 1 hidden layer.

Tabel 5. Performance Results of BPN-GA Classification with 4 Neurons at 1 Hidden Layer

CV	Data Testing			Data Training		
	AUC	G-mean	Accuracy	AUC	G-mean	Accuracy
1	0,50	0,00	96,62	0,51	0,00	96,63
2	0,51	0,13	96,05	0,50	0,08	96,09
3	0,52	0,00	96,60	0,50	0,00	96,65
4	0,53	0,11	96,66	0,52	0,04	96,59
5	0,48	0,00	96,62	0,53	0,00	96,64
6	0,50	0,00	96,64	0,51	0,00	96,64
7	0,52	0,00	96,43	0,52	0,00	96,65
8	0,50	0,18	93,34	0,51	0,21	93,28
9	0,50	0,00	96,62	0,51	0,00	96,63
10	0,52	0,50	39,11	0,51	0,50	39,31
Average	0,51	0,10	90,47	0,51	0,10	90,51

The results of the analysis in Table 5 show that the classification accuracy of the training data is 90.51% and the testing data is 90.47%, it is known that the classification accuracy of the Neural Network Backpropagation model formed in terms of accuracy is 90.47%, in terms of AUC of 0.51 and viewed from the G-mean of 0.10. In this case, the results of the training data and testing data produce almost the same difference, so it can be said that the model is quite good.

3.3. Performance Level of Backpropagation Neural Network Classification Without and With Optimization of Weights and Backpropagation Bias

This section describes the comparison of the two methods that have been carried out, namely the classification analysis with BPN and BPN-GA. Table 5 presents the Performance of the Classification Results for Neural Network Backpropagation Without and with Optimization of Weights and Backpropagation Bias.

Based on Table 5, it can be seen that the initial weights and biases optimized using the Genetic Algorithm are able to improve the classification performance results for AUC, G-mean and accuracy. It can be seen the results of the Backpropagation performance with the NN structure (23-4-1) before optimization or manual random weight and bias parameters on the testing data resulting in an accuracy value of 78.00%, a G-mean value of 0.01 and a value AUC 0.51. After optimization using the Genetic Algorithm, an accuracy value of 90.47%, a G-mean value of 0.09, an AUC value of 0.51, although not significantly increased, it can be said that optimization of weights and initial bias using genetic algorithms can improve performance classification.

Table 5. Performance of Classification Results Without and with Optimization of Weights and Bias in Backpropagation Neural Networks

NN structure	Data Testing			Data Training		
	AUC	G-mean	Accuracy (%)	AUC	G-mean	Accuracy (%)
23-4-1 (With Optimization)	0,51	0,01	78,00	0,51	0,01	78,02
23-4-1 (Without Optimization)	**0,51**	**0,09**	**90,47**	**0,51**	**0,08**	**90,51**

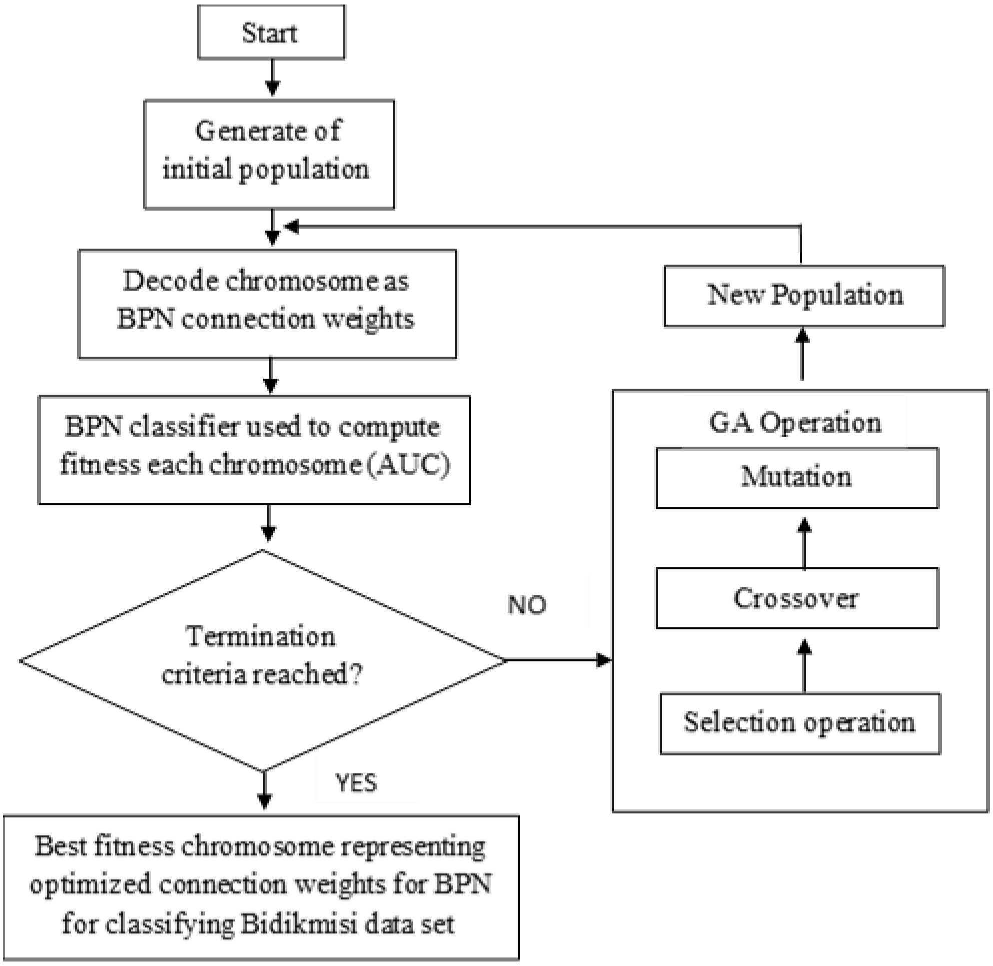
Start
Generate of initial population
Decode chromosome as BPN connection weights
BPN classifier used to compute fitness each chromosome (AUC)
Termination criteria reached?
NO
YES
Best fitness chromosome representing optimized connection weights for BPN for classifying Bidikmisi data set
New Population
GA Operation
Mutation
Crossover
Selection operation

Figure 1. Working of hybrid BPNGA

4. Conclusion

As we can see in the results, we can say that optimization of backpropagation can improve the neural network training process by increasing the backpropagation classification performance. The backpropagation performance with NN architecture (23-4-1) before optimization are randomly carried out manually on data testing, its accuracy value is 78%. After optimization using the genetic algorithm, the accuracy value has increased by 90.47%, a G-mean value of 0.09 and the AUC value is 0.51. Although it does not increase significantly, it can be said that the initial weight optimization and bias using genetic algorithms can improve classification performance.

5. Acknowledgments

This research was supported by DPRM-DIKTI under scheme PUPT, project No. 1049/PKS/ITS/2018. The author thanks the Kementerian Riset, Teknologi, dan Pendidikan Tinggi for funding and to anonymous references for their useful suggestions.

.

6. References

[1] Haykin S 1999 Neural Network: A Comprehensive Foundation 2nd Edition Prentice Hall New Jersey.

[2] Cahyani, N, Fithriasari, K, Irhamah, and Iriawan, N (2018). On the Comparison of Deep Learning Neural Network and Binary Logistic Regression for Classifying the Acceptance Status of Bidikmisi Scholarship Applicants in East Java. Malaysian Jurnal Of Industrial and Applied Mathematics.

[3] Rumelhart D E, Hinton G E and Williams R J 1986 Learning Representations by Back-Propagating Errors Nature 323, pages 533 – 536.

[4] Haupt R L and Haupt S E 2004 Practical Genetic Algorithms New Jersey: A John Wiley and Sons Inc.

[5] Karegowda A G, Manjunath A S, and Jayaram M A 2011 Application of Genetic Algorithm Optimized Neural Network Connection Weights for Medical Diagnosis of PIMA Indians Diabetes International Journal on Soft Computing (IJSC) 2(2).

[6] Lu C and Shi B 2000 Hybrid Back-Propagation/ Genetic Algorithm for Feedforward Neural Networks ICSP.

[7] Zhang M and Ciesielki V 1998 using Back Propagation Algorithm and Genetic Algorithms to Train and Refine Neural Networks for Object Detection Computer Science Postgraduate Student Conference Melbourne.

[8] Wahyu, W, Rahman, S.A, and Nita, C. (2019). Multilevel Logistic Regression and Neural Network-Genetic Algorithm for Modeling Internet Access. 5th International Conference, SCDS 2019 Iizuka, Japan, August 28–29, 2019 Proceedings.

[9] Khan A U, Bandopadhyaya T K, and Sharma S 2008 Genetic Algorithm Based on Back-propagation Neural Network Performs Better than Back-propagation Neural Network in Stock Rates Prediction International Journal of Computer Science and Network Security 8(7), pages 162 – 166.

[10] Sadeq A M, Wahdan A M A, and Mahdi H M K 2000 Genetic Algorithm and Its Use With back-propagation Network AIN Shams University Scientific Bullettin 35(3), pages 337 – 348.

[11] Last, M. (2006). The Uncertainty Priciple of Cross-Validation. *IEEE Conference Publications*, 275-280.

[12] Kusumadewi, S. (2004). Membangun Jaringan Saraf Tiruan Menggunakan MATLAB & EXCEL LINK. Yogyakarta: Graha Ilmu.

[13] Trevino, V., & Falciani, F. (2006). An R Package for Multivariate Variable Selection Using Genetic Algorithms. Bioinformatics, 1154-1156.

[14] Han, J., & Kamber, M. (2006). *Data mining: Concepts and Techniques.* California: Second Edition, Morgan Kaufmann.