TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 19, No. 6, December 2021, pp. 1892~1901
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v19i6.20409

Metaheuristic optimization in neural network model for seasonal data

Budi Warsito, Rukun Santoso, Hasbi Yasin
Department of Statistics, Faculty of Science and Mathematics, Diponegoro University, Semarang, Indonesia

Article Info

Article history:

Received Aug 15, 2020
Revised Sep 5, 2021
Accepted Sep 15, 2021

Keywords:

Metaheuristic
Neural network
Optimization
Rainfall
Time series

ABSTRACT

The use of metaheuristic optimization techniques in obtaining the optimal weights of neural network model for the time series was the main part of this research. The three optimization methods used as experiments were genetic algorithm (GA), particle swarm optimization (PSO), and modified bee colony (MBC). Feed forward neural network (FFNN) was the neural network (NN) architecture chosen in this research. The limitations and weaknesses of gradient-based methods for learning algorithm inspired some researchers to use other techniques. A reasonable choice is non-gradient based method. Neural network is inspired by the characteristics of creatures. Therefore, the optimization techniques which are also resemble the patterns of life in nature will be appropriate. In this study, various scenarios on the three metaheuristic optimization methods were applied to get the best one. The proposed procedure was applied to the rainfall data. The experimental study showed that GA and PSO were recommended as optimization methods at FFNN model for the rainfall data.

Corresponding Author:

Budi Warsito
Department of Statistics Faculty of Science and Mathematics
Diponegoro University
Prof. Soedarto SH. Street, Tembalang, Semarang, Indonesia
Email: budiwrst2@gmail.com

1. INTRODUCTION

The development of neural network modeling applied to time series data has grown rapidly. Some interesting problems to be investigated include the determination technique for the optimal input, the number of units in the hidden layer [1]-[3], the activation function in the hidden layer [4], [5], and the selection of optimization methods to get the optimal weight [6]-[8]. Modeling procedures to obtain the optimal architecture have also been developed, in terms of theoretical, application and computational studies [9]-[11]. In terms of finding the optimal weights, the optimization method is one of the main focuses of the discussion. Gradient-based optimization techniques have become the standard method for this problem. As a consequence, the activation function used must meet the continuous and differentiable conditions. The idea to develop non-gradient based metaheuristic optimization methods for the optimization of a function has also progressed a lot [12], [13]. Support from advances in computational aspects also facilitates the development of new non-gradient methods. Advances in statistical and mathematical modeling aspects have also generated various alternative models to get better predictions. The complexity of these models is getting higher. As a consequence, appropriate optimization techniques are needed to obtain parameter estimates.

The idea of developing metaheuristic algorithms as optimization techniques is a new chapter in statistical modeling. These non-gradient methods are useful in terms of parameter estimation of alternative models. Several optimization methods that fall into this category include genetic algorithm [14]-[16], ant colony [17], [18], simulated annealing [19], and particle swarm optimization [20], [21]. The use of these methods for parameter estimation of statistical models is still limited. Previous studies have demonstrated the superiority of metaheuristic optimization techniques in terms of estimation results. However, the drawbacks have also been found in terms of the iteration time required to reach convergence, which is longer than the gradient-based method. The purpose of this study is to find the best optimization method of the three metaheuristic algorithms, namely genetic algorithm (GA), particle swarm optimization (PSO), and modified artificial bee colony (MABC). This procedure is applied to rainfall, a data type that is known to have a seasonal pattern. In this case, the proposed procedure is used to measure the accuracy of in-sample predictions and out-sample predictions.

2. RESEARCH METHOD

This section describes the modeling stages carried out. The explanation is divided into two main parts. The first part describes the algorithm of neural network modeling and its architecture in the case of time series. The second part discusses the metaheuristic optimization methods used to obtain the optimal weights of the neural network model. There are three methods discussed, namely GA, PSO, and MABC. The proposed procedure is applied to the ten daily rainfall data which is a time series containing seasonal aspects. At the data processing stage, predictions are made of the training and testing data, then the accuracy of the three optimization methods is compared. A brief explanation of each part is presented as follows:

2.1. Feed forward neural networks

Neural network (NN) is a modeling algorithm inspired by the human brain, mimicking the way of signaling between biological neurons. The main class of neural network model is feed forward neural network (FFNN). Backpropagation algorithm is the most popular learning method. In FFNN, hidden layer(s) are inserted between input and output. FFNN is one type of neural network which is most often used in various applications, including time series [22], [23]. Architecture of this class contains a number of processing units such as simple neurons arranged in layers. The units in each layer are called neurons or nodes. Each neuron is connected to each neuron in the next layer. The strength of the relationship between the layer units is expressed as weights. The weights can vary depending on how strong the connection between the neurons is.

Complexity of the neural network model is determined by how many units are in each layer. The more complex the network, the more weights should be estimated. The type and number of units in the input layer is largely determined by the purpose of the model. In the application for classification problems, the network input is determined first. Meanwhile, in the application for time series problems, the input is influenced by how strong the relationship between lagged variables is with the current data as the output target. Meanwhile, the determination of the number of units in the output layer is based on univariate or multivariate analysis. In terms of the application of FFNN for univariate time series modeling, various modeling procedures continue to develop. Therefore, the number of neurons in output layer is one and the output is (1).

$$y = f^o\left(\sum_{j=1}^{l} w_j^o f_j^h\left(\sum_{i=1}^{n} w_{ji}^h x_i\right)\right) \quad (1)$$

Where f^o is the activation function in output layer, w_j^o is the weight from hidden unit j to output, f_j^h is is the activation function in hidden unit j, w_{ji}^h is the weight from input i to hidden unit j. In (1) has no bias, it can be added as input. The model also accommodates various activation functions for each unit in the hidden layer. If all the same, i.e. f^h, then (1) becomes:

$$y = f^o\left(w^b + \sum_{j=1}^{l} w_j^o f^h\left(w_j^b + \sum_{i=1}^{n} w_{ji}^h x_i\right)\right) \quad (2)$$

where w^b is the weights from bias to output and w_j^b is the weights from bias to hidden layer. In the perspective of time series modeling, past data series is input of the model, while the present data x_t is the output. Hence, if input is lagged values of 1 until p, or $x_{t-1}, \ldots, x_{t-p}$, in (2) becomes:

$$x_t = f^o\left(w^b + \sum_{j=1}^{l} w_j^o f^h\left(w_j^b + \sum_{i=1}^{n} w_{ji}^h x_{t-i}\right)\right) \quad (3)$$

configuration of FFNN architecture for time series is as follows. The input network consists of x_{t-1} up to x_{t-p} and a bias. The hidden layer consists of n neurons while the output contains one neuron. This architecture is represented in Figure 1.

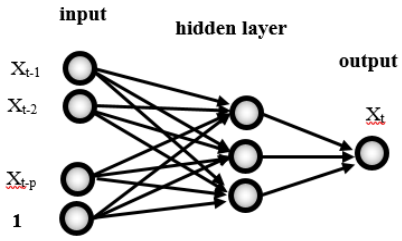


Figure 1. FFNN architecture for time series modeling

Three steps of backpropagation algorithm in neural network modeling sequentially consist of feedforward, error calculation and adjusting weights. In the feedforward stage, each input unit sends a signal to all neurons in the hidden layer, as well as from the hidden layer to the output. Furthermore, the differencing between target and output will produce errors. Updating the weights is carried out with certain optimization methods to reduce errors. With the new weights obtained, the feedforward stage is then repeated again. This procedure continues until the stopping criterion is reached, either the maximum epoch or the minimum error. In this study, metaheuristic optimization is used as an update of the weights. The main objective of metaheuristic optimisation in feed forward neural network modeling is to get the best prediction resulting from the optimal weights.

2.2. Genetic algorithm (GA)

GA is a metaheuristic search that is routinely used to obtain beneficial solutions for optimization and search problems [24]. This algorithm does its job by imitating natural genetic mechanisms, which search for the best gene structures in a creature's body. The theory of evolution is the basis for the emergence of genetic algorithms. In this theory, species with better adaptability will have a greater chance of survival. The algorithm begins with the initialization of a set of potential solutions. Adjustments through iterations are made to obtain the best solution. The set of potential solutions is called a chromosome and is predetermined. Chromosomes are formed in the form of a binary alphabet. The set of chromosomes represents a population. Iteration stages in the process of chromosome evolution are called generations. The evolution process consists of selection, crossover and mutation. In each generation, the process of evolution will produce a new generation or offspring. Genetic algorithms have several characteristics such as: the search is carried out on a population of points, not only from single point; work is performed on a set of encoding parameters, not directly on the parameters themselves; based on information on the fitness function or objective function, not the derivative function; and random operations are performed on each iteration with probabilistic adjustments, not with derivative rules.

The candidate solutions are encoded in the form of chromosomes. Each chromosome contains the genes and is the same length. The element in each gene is a binary alphabet. In the regeneration process, each x chromosome corresponds to the fitness function f(x). Determination of the length of chromosome, encoding and alphabet which is a mapping from a group with a certain universe of discourse (Ω) to a set of chromosomes is called a representation scheme. Selection operation is the first step of genetic algorithm after initial population P(0). At this stage, a set of mating pools M(k) is formed whose number of elements is identical to the number of elements in P(k). Every point m(k) in M(k) is taken from the points x(k) in P(k). The next stage is evolution process. At this stage, crossing over is carried out by taking a pair of chromosomes as the parent which gives birth to a new pair of chromosomes called offspring. The probability of selecting a parent pair of M(k) is random with probability p_c. Furthermore, the mutation process is applied with a small probability, $p_m << 1$, by changing or reversing the value of one or more genes in a chromosome. Good chromosomes will be preserved to survive in the next generation. Elitism is an assistance strategy that is applied by saving good chromosomes in the previous generation so that they are still preserved in the next generation. Linear fitness ranking (LFR) is another useful strategy for measuring fitness scores based on individual evaluations. This strategy is carried out to reduce the effect of a large variance on the fitness value obtained. This can be useful to avoid the tendency to converge to a local optimum solution. To get the chromosomes with the best fitness, this procedure is repeated until certain stopping criteria are fulfilled.

The step by step of genetic algorithm can be summarized as follows: i) shape the initial population or P(0), ii) evaluate P(k), iii) if stopping criteria is fullfilled then stop, iv) select M(k) from P(k), v) arrange M(k) to the form P(k + 1), and vi) go to step 2 (set: k = k + 1). In this research, the specifications of genetic algorithm were: population size=20, the number of generations=10000, probability of crossover p_c=0.7, probability of mutation p_m=0.1. Roulette wheel was used as the parents couple selection method.

2.3. Particle swarm optimization

Particle swarm optimization (PSO) is motivated by intelligent collective behavior of some animals such as flocks of birds or schools of fish [25]. Among the various algorithms in swarm intelligence, PSO is considered the most important one [26]. It is a population-based stochastic optimization algorithm. In weight optimization with PSO, the initial position of the particles is generated randomly. At this initial position, all particles do not move so that the initial velocity is given a value of zero. Selection of the fitness value in this initial position is very important because it will determine the best global position (gBest) and the best individual position (pBest). In this study, mean square error (MSE) is used as a fitness value. PSO stages for optimizing FFNN weights are described systematically as follows: i) Determine the initial values including the number of particles, the coefficient of acceleration, maximum number of iterations, and the weight of inertia; ii) In randomly, determine the initial velocity and initial position of each particle; iii) Calculate the output based on the weights in the initial position; iv) Calculate the fitness values of each particle, then select the minimum MSE as the optimal fitness; v) Choose the best position (*pBest*) based on the fitness value. The best position chosen from each iteration becomes the best global (*gBest*); vi) Update velocity and particle position; vii) Determine the new position; viii) Calculate the network output by using the weights at the new position; and ix) Go to 4.

These stages continue up to the stopping criteria are met. In this research, the maximum number of iterations was 500, and the population size (swarm size) was 10. These are the PSO parameters used in this research: inertia weight =1, personal learning coefficient=1.5, inertia weight damping ratio=0.99, and global learning coefficient=2.0.

2.4. Modified artificial bee colony

In the search of optimal solution, artificial bee colony (ABC) adopts the habit of a swarm of bees in search of food. It was developed by Karaboga and Basturk [27]. The performance of ABC has better quality or is equivalent to other swarm algorithms such as genetic algorithm, particle swarm optimization, differential evolution, and evolution strategies with the advantage of using fewer control parameters [28]. The stages of ABC algorithm were:

1) Initialization of population using $x_{ij} = x_{minj} + rand(0,1)(x_{maxj} - x_{minj})$, where x_{ij} is position of population i and parameter j.
2) Evaluation of population
3) Iteration = 1
4) Replication
   a. For the stage of worker bee:
   – Generate a new solution: $v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$, where v_{ij} is new position of population i and parameter j whereas ϕ_{ij} is a uniformly distributed random number in the range of -1 and 1.
   – Determine fitness of the solution using $fitness_i = \frac{1}{MSE}$
   – Compare v_i and x_i
   – Determine the probability using $p_i = \frac{fitness_i}{\sum_{n=1}^{SN} fitness_i}$
   b. For the stage of guardian bee:
   – Choose the x_{ij} solution based on p_i
   – Generate a new v_i
   – Determine fitness of the solution using $fitness_i = \frac{1}{MSE}$
   – Compare v_{ij} and x_{ij}
   c. For the stage of surveillance bee:
   If there is a sowlution left behind, replace it by generating a new solution x_i randomly using $v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$
   d. Save the best solution
   e. Add iteration with 1, iteration = iteration + 1
5) Until the requirements are met or iteration = maximum iteration

The modification of the ABC algorithm provides better convergence performance when compared to the ordinary ABC algorithm [29]. Modifications made are on the ABC formula as follows:

$$v_{ij} = x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right) + \varphi_{ij}\left(y_j - x_{ij}\right)$$

where φ_{ij} is a uniformly distributed random number in the range of 0 to 1.5 while y_j is the jth element of the best global solution. The formula is inspired by the search mechanism of the PSO algorithm and is used to improve the level of convergence of the PSO algorithm. In addition, the solution probability formula was also changed to:

$$p_i = \exp(-\frac{1}{\rho} fitness_i)$$

where $\rho = 2.5$. Research conducted by Shahrudin and Mahmuddin [29] showed that no matter what value is used in the simulation, it concludes that MABC is better than ABC.

3. RESULTS AND DISCUSSION

In this study, we used the ten-daily of rainfall from ZOM 138 Bawak Klaten Central Java from January 2010 until July 2018 with the length of 309. This data was taken from Meteorology, Climatology and Geophysics Agency (BMKG). The data is divided into two parts. The first part is the training data used for in-sample prediction while the second part is for testing data. The actual data is 309 which is divided into 273 training data and the remaining 36 as testing data. The proportion of training and testing data is close to the 90:10 composition, but not exactly. The argument is that the type of data used is ten-daily data and contains seasonal aspects, so that in one year there are 36 observation points. Of course, it would be wise to use the last year's data as testing data, instead of an incomplete year. Since the input variable is up to lag 18, the training data becomes 255. In this research, each optimization method was tried for several architectures. To match the number of inputs used, the number of hidden units used is from one to two times the input. In this case, there are three input variables i.e. lags of 1, 2, and 18 so the number of hidden unit is from one to six. The activation functions used in hidden layer and output layer are logistic sigmoid and linear, respectively. Table 1 shows the results of each experiment.

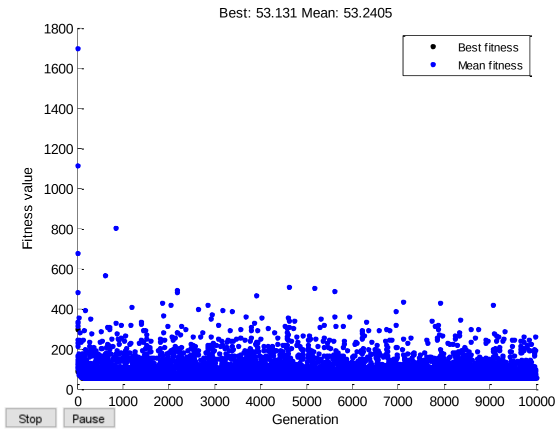
Table 1 shows that optimization of neural network model by using GA given the best value of MSE based on the testing data for out-sample prediction. This happens in experiments with three hidden units which are equal to the number of input variables. This is appropriate with the rule of thumb that the units in input correspond to the units in hidden layer. Meanwhile, if the results of the out-sample prediction with testing data are used as a basis for selection, the best method is PSO with five hidden units. However, it appears that the results of in-sample predictions from the GA method with four to six hidden units are able to approach the best results from the PSO. This provides sufficient reason to choose GA as the best method. This is consistent with the result that the out-sample prediction of the PSO method with five hidden units is very poor. Thus, PSO is less successful in guaranteeing that good in-sample predictions will also produce good out-sample predictions. The results of the MABC method also indicate not to choose this method as the recommended technique. Plot of the convergence graph of the GA and PSO shown at Figure 2 indicate the effectiveness of the performance due to increasing number of iterations.

In Figure 2 (a), it can be seen that there is a rapid decline at the beginning of the iteration or generation of the GA optimization. The calculated points displayed are the average fitness value of the population which is a set of chromosomes and the best fitness. Fluctuations depicted show the rise and fall of the average fitness. The results of these mean fitness indicate that in each generation, there are very bad and very good chromosomes. Chromosome that produces the best fitness values is preserved in the next generation to ensure convergence. The process towards convergence requires relatively long iterations. This indicates that there is a decrease in the accuracy value which is rather slow but continues. Similarly, the PSO optimization in Figure 2 (b) also shows an extreme slowdown at the beginning of the iteration. The displayed figure is the best value for each iteration. The iteration process towards convergence in PSO is very fast. In fact, convergence has appeared after 200 iterations. In terms of the number of iterations, PSO is more efficient. It requires fewer iterations than GA. However, the time required for one iteration in PSO is relatively longer than in GA. Overall, both methods give convergent results at the optimal MSE values.

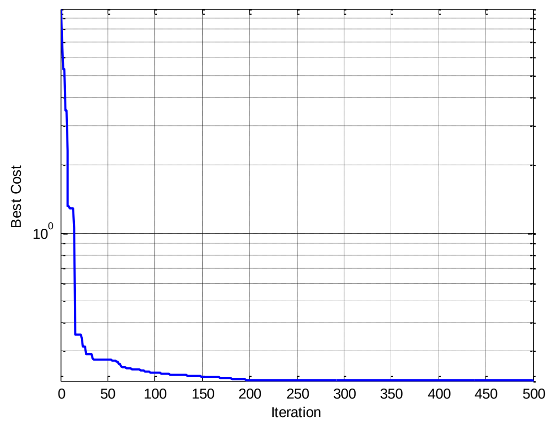
More interesting discussion arises when looking at the results of the averages and variances of each method. Taking into account the mean value, GA appears to be superior to both in-sample and out-sample predictions. The results obtained indicate that the GA method provides the lowest of MSEs average. PSO provides out-sample prediction results that are similar to MABC but produces better in-sample predictions on average. By paying attention to variance, it strengthens to choose GA because it has the smallest variance value in both training and testing data. These results indicate that GA is the best in terms of the stability of the results. Meanwhile, MABC is more stable in the out-sample prediction than PSO and vice versa, PSO is more stable in the in-sample prediction than MABC. Based on the discussion, GA is recommended to be chosen as the top priority for the optimization method of the neural network model. Of course, this applies to the case of rainfall data containing seasonal patterns as restricted in this study. As an illustration, plots of the results of in-sample predictions versus actual and out-sample predictions versus actual and the from the GA method can be seen in the Figure 3 and Figure 4. The in-sample predictions are obtained from modeling using actual training data whereas out-sample predictions are obtained from that using actual testing data. The blue line indicate the actual and the blue one represent the predictions.

Table 1. Results of the experiments

Metaheuristic Optimization Method	Number of hidden units	MSE (x10^3)	
		Training	Testing
Genetic Algorithm	1	2.2735	2.5435
	2	2.2196	2.7110
	3	2.2263	2.4919
	4	2.1658	2.7055
	5	2.1707	2.4946
	6	2.1703	2.5784
	average	2.2044	2.5875
	variance	0.0018	0.0098
Particle Swarm Optimization	1	2.2170	2.7121
	2	2.3849	2.8685
	3	2.2541	2.6390
	4	2.2944	2.5171
	5	2.1551	3.1108
	6	2.2587	2.5004
	average	2.2607	2.72465
	variance	0.0059	0.0541
Modified Artificial Bee Colony	1	2.4027	2.6378
	2	2.3953	2.6468
	3	2.5396	2.8032
	4	2.2446	2.5753
	5	2.4242	2.8645
	6	2.3840	2.8003
	average	2.3984	2.7213
	variance	0.0089	0.0135



(a)



(b)

Figure 2. Convergence graph of the (a) GA and (b) PSO

Plots in Figure 3 and Figure 4 indicate that seasonal patterns in rainfall data can be approached and predicted well by the model. However, the results obtained from the proposed modeling procedure seem unable to detect data with extreme values. If there is data with extreme high values, the prediction results are not able to predict well. This is the biggest source of error of the model used. Seasonal patterns from rainfall data have indeed been approached successfully by the model, but not with these extreme data. This becomes a weakness of the proposed modeling procedure and provides an open space for finding models or methods that are more suitable for this type of data.

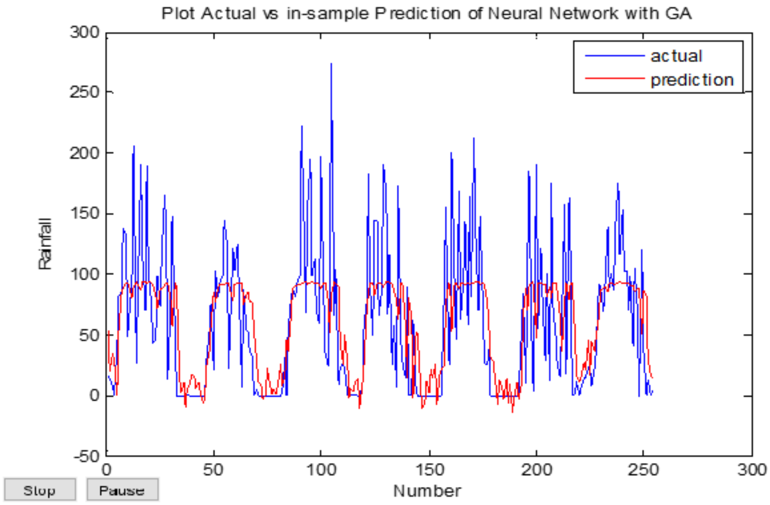


Figure 3. Plot of actual vs in-sample prediction of neural network with genetic algorithm

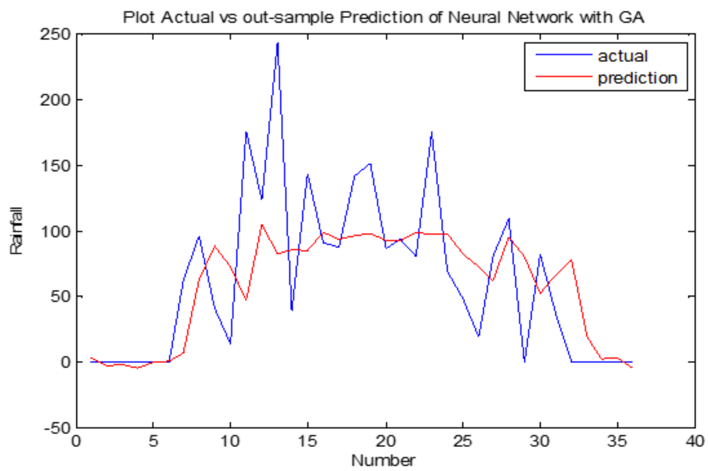


Figure 4. Plot of actual vs out-sample prediction of neural network with genetic algorithm

A review of the previously reported studies was carried out to evaluate the results obtained from this research. Mostly, the results obtained are in accordance with previous studies. The metaheuristic optimization method has also been proven successful in optimizing the adaptive neuro-fuzzy inference system (ANFIS) model for rainfall data [30]. In this report, PSO, GA and the hybrid are suitable for optimizing models and better than classical one. In [31], GA appears to be superior to PSO for optimizing Job Shop Scheduling Problems. Similarly, as reported in [32], GA also produced the highest performance in estimating the heating load of building energy efficiency for smart city planning, superior to PSO, imperial competitive algorithm (ICA), and ABC. Likewise, the three metaheuristic methods have been used for optimizing support vector machine (SVM) in the case of classification [33]. The GA had the highest average overall accuracy, followed by ABC and PSO. Whereas, in [34] ABC is better than or similar to GA and PSO with the advantage of employing fewer control parameters. Research involving all three methods with equally good results was obtained in [35]. In this report, is more successful in evolving larger networks and the PSO is more successful on smaller networks. Ramdania *et al.* [36] has reported that the PSO fitness value outperforms the genetic algorithm, but the genetic algorithm execution time is faster than the PSO algorithm. PSO involves less overall computation effort than GA but shown to outperform the GA for smaller population sizes [37]. Slightly different results were obtained in [38] and [39]. In the two results, PSO gave better results than GA and ABC. It should be noted that most of the research done has been applied in addition to the seasonal time series problem. Therefore, the characteristics of the data are very influential on the results obtained.

4. CONCLUSION

Three metaheuristic based optimization methods have been used to determine the optimal weight of the neural network model in the rainfall data. With a variety of architectures determined, optimization with genetic algorithms is recommended for use in models and data types of this type. This technique is more stable and provides better predictions than the other two techniques. The problem of detecting data with extreme values may be solved in the future by using appropriate preprocessing such as data normalization. For the purposes of validating the results of predictions and comparisons more broadly, the using of some other optimization techniques can also be investigated. Furthermore, it can also be combined between metaheuristic optimization methods with various gradient-based optimization methods to get better prediction results. The use of hybrid methods between metaheuristic optimization techniques can also be one possible solution to better predict extreme data. The various optimization techniques proposed can also be applied in searching of weights on other types of neural networks such as recurrent neural networks, convolutional neural networks and cascade forward neural networks and even for machine learning and other deep learning models. Further modeling procedures can also be applied to various types of data that are not included in the seasonal category.

ACKNOWLEDGEMENTS

This research was funded by the Institute for Research and Community Service, Diponegoro University via the scheme of International Publication Research with the contract number: 831.1-08/UN7.P4.3/PP/2019.

REFERENCES

[1] S. Karsoliya, “Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture,” *International Journal of Engineering Trends and Technology*, vol. 3, no. 6, pp. 714-717, 2012. [Online]. Available: http://www.ijettjournal.org/volume-3/issue-6/IJETT-V3I6P206.pdf

[2] F. S. Panchal and M. Panchal, “Review on methods of selecting number of hidden nodes in artificial neural network,” *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 11, pp. 455-464, 2014. [Online]. Available: https://www.ijcsmc.com/docs/papers/November2014/V3I11201499a19.pdf

[3] D. P. Pitambare, “Survey on optimization of number of hidden layers in neural networks,” *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 11, 2016. [Online]. Available: https://www.ijarcce.com/upload/2016/november-16/IJARCCE%20115.pdf

[4] B. Karlik and A.V. Olgac, “Performance analysis of various activation functions in generalized MLP architectures of neural networks,” *International Journal of Artificial Intelligence and Expert Systems (IJAE)*, vol. 1, no. 4, pp. 111-122, 2011. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.740.9413 &rep=rep1&type=pdf

[5] E. Sevinç, “Activation functions in single hidden layer feed-forward neural networks,” *Journal of Selçuk-Teknik*, pp. 1-13, 2018. [Online]. Available: https://core.ac.uk/download/pdf/267827237.pdf

[6] A. M. Schweidtmann and A. Mitsos, “Global deterministic optimization with artificial neural networks embedded,” *Journal of Optimization Theory and Applications*, vol. 180, pp. 925-948, 2019, doi: 10.1007/s10957-018-1396-0. [Online]. Available: https://arxiv.org/pdf/1801.07114.pdf

[7] G. Nowakowski, Y. Dorogyy, and O. D.-Ivaniuk, “Neural network structure optimization algorithm,” *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. 12, no. 1, 2018, doi: 10.14313/JAMRIS_1-2018/1.

[8] D. Devikanniga, K. Vetrivel, and N. Badrinath, “Review of meta-heuristic optimization based artificial neural networks and its applications,” *Journal of Physics: Conference Series*, vol. 1362, no. 1, 2019, doi: 10.1088/1742-6596/1362/1/012074. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1362/1/012074/pdf

[9] R. Luo, F. Tian, T. Qin, E. Chen, and T.-Y. Liu, “Neural architecture optimization,” *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, pp. 7816-7827, 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/933670f1ac8ba969f32989c312faba75-Paper.pdf

[10] T. Elsken, J. H. Metzen, and F. Hutter, “Neural architecture search: a survey,” *Automated Machine Learning*, pp. 63-77, 2019, doi: 10.1007/978-3-030-05318-5_3.

[11] S. Hu *et al*., "DSNAS: Direct Neural Architecture Search without Parameter Retraining," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 2020, pp. 12081-12089, doi: 10.1109/CVPR42600.2020.01210.

[12] E. K. Nyarko, R. Cupec, and D. Filko, "A comparison of several heuristic algorithms for solving high dimensional optimization problems," *International Journal of Electrical and Computer Engineering Systems*, vol. 5, no. 1, pp. 1-8, 2014. [Online]. Available: file:///C:/Users/Asus/Dropbox/TELK/ijeces_vol_5_no_1_01.pdf

[13] A. Shabani, B. Asgarian, S. A. Gharebaghi, M. A. Salido, and A. Giret, "A new optimization algorithm based on search and rescue operations," *Mathematical Problems in Engineering*, vol. 2019, 2019, doi: 10.1155/2019/2482543.

[14] E. Tatlow, "Hybridization of neural networks and genetic algorithms," *Network Architectures and Services*, pp. 53-60, 2018, doi: 10.2313/NET-2018-03-1_08. [Online]. Available: https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2018-03-1/NET-2018-03-1_08.pdf

[15] H. Chung and K. S. Shin, "Genetic algorithm-optimized long short-term memory network for stock market prediction," *Sustainability*, vol. 10, no. 10, 3765, 2018, doi: 10.3390/su10103765.

[16] J. Revathi, J. Anitha, and D. J. Hemanth, "Training feedforward neural network using genetic algorithm to diagnose left ventricular hypertrophy," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 18, no. 3, pp. 1285-1291, 2020, doi: 10.12928/telkomnika.v18i3.15225.

[17] V. K. Ojha, P. Dutta, A. Chaudhuri, and H. Saha, "Understating continuous ant colony optimization for neural network training: a case study on intelligent sensing of manhole gas components," *International Journal of Hybrid Intelligent Systems*, vol. 12, no. 4, pp. 185-202, 2015, doi: 10.3233/HIS-160215.

[18] A. ElSaid, F. E. Jamiy, J. Higgins, B. Wild, and T. Desell, "Using ant colony optimization to optimize long short-term memory recurrent neural networks," *GECCO'18: Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 13-20, July 2018, doi: 10.1145/3205455.3205637.

[19] W.-C. Hong, "Traffic flow forecasting by seasonal svr with chaotic simulated annealing algorithm," *Neurocomputing*, vol. 74, no. 12-13, pp. 2096-2107, 2011, doi: 10.1016/j.neucom.2010.12.032.

[20] L. Jin, Y. Huang, and H. Zhao, "Ensemble prediction of monthly mean rainfall with a Particle Swarm Optimization-neural network model," *2012 IEEE 13th International Conference on Information Reuse & Integration (IRI),* 2012, pp. 287-294, doi: 10.1109/IRI.2012.6303022.

[21] B. Warsito, H. Yasin, and A. Prahutama, "Particle swarm optimization versus gradient based methods in optimizing neural network," *Journal of Physics: Conference Series*, vol. 1217, May, 2019, doi: 10.1088/1742-6596/1217/1/012101.

[22] F. Saâdaoui, "A seasonal feedforward neural network to forecast electricity prices," *Neural Computing and Applications*, vol. 28, no. 4, pp. 835-847, 2017, doi: 10.1007/s00521-016-2356-y.

[23] S. Scher and G. Messori, "Weather and climate forecasting with neural networks: using General Circulation Models (GCMs) with different complexity as a study ground," *Geoscientific Model Development*, vol. 12, no. 7, pp. 2797-2809, 2019, doi: 10.5194/gmd-12-2797-2019.

[24] P. Bajpai and M. Kumar, "Genetic algorithm-an approach to solve global optimization problems," *Indian Journal of Computer Science and Engineering*, vol. 1, no. 3, pp. 199-206, 2010.

[25] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, no. 2, pp. 387-408, 2018, doi: 10.1007/s00500-016-2474-6.

[26] M. A. El-Shorbagy and A. E. Hassanien, "Particle swarm optimization from theory to applications," *International Journal of Rough Sets and Data Analysis (IJRSDA)*, vol. 5, no. 2, pp. 1-24, 2018, doi: 10.4018/IJRSDA.2018040101.

[27] D. Karaboga and B. Basturk, "Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems," In *Foundations of Fuzzy Logic and Soft Computing. IFSA 2007*, Lecture Notes in Computer Science, P. Melin, O. Castillo, L. T. Aguilar, J. Kacprzyk, W. Pedrycz, Eds., Springer, Berlin, Heidelberg, vol 4529, doi: 10.1007/978-3-540-72950-1_77.

[28] D. Karaboga and B. Akay, "Artificial Bee Colony (ABC), harmony search and bee algorithms on numerical optimization," *Innovative production machines and systems virtual conference,* March 2009.

[29] M. S. Shahrudin and M. Mahmuddin, "Experiment on modified artificial bee colony for better global optimisation," *Advances in Computer Science and its Applications*, pp. 735-741, 2014, doi: 10.1007/978-3-642-41674-3_105.

[30] Z. M. Yaseen *et al.*, "Novel hybrid data-intelligence model for forecasting monthly rainfall with uncertainty analysis," *Water*, vol. 11, no. 3, 2019, doi: 10.3390/w11030502.

[31] A. Muthiah and R. Rajkumar, "A comparison of artificial bee colony algorithm and genetic algorithm to minimize the makespan for job shop scheduling," *Procedia Engineering*, vol. 97, pp. 1745-1754, 2014, doi: 10.1016/j.proeng.2014.12.326.

[32] L.T. Le, H. Nguyen, J. Dou, and J. Zhou, "A comparative study of PSO-ANN, GA-ANN, ICA-ANN, and ABC-ANN in estimating the heating load of buildings' energy efficiency for smart city planning," *Applied Sciences*, vol. 9, no. 13, 2019, doi: 10.3390/app9132630.

[33] X. Zhu, N. Li, and Y. Pan, "Optimization performance comparison of three different group intelligence algorithms on a SVM for hyperspectral imagery classification," *Remote Sensing*, vol. 11, no. 6, 2019, doi: 10.3390/rs11060734.

[34] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108-132, 2009, doi: 10.1016/j.amc.2009.03.090.

[35] M. Settles, B. Rodebaugh, and T. Soule, "Comparison of genetic algorithm and particle swarm optimizer when evolving a recurrent neural network," In *Genetic and Evolutionary Computation-GECCO 2003. GECCO 2003. Lecture Notes in Computer Science,* Cantú-Paz E. *et al*. Eds., vol. 2723, 2003, doi: 10.1007/3-540-45105-6_17.

[36] D. R. Ramdania, M. Irfan, F. Alfarisi, and D. Nuraiman, “Comparison of genetic algorithms and Particle Swarm Optimization (PSO) algorithms in course scheduling,” *Journal of Physics: Conference Series*, vol. 1402, no. 2, 2019, doi: 10.1088/1742-6596/1402/2/022079.
[37] Z. Wang, T. J. McCarthy, and M. N. Sheikh, “A comparison of genetic algorithm and particle swarm optimisation for theoretical and structural applications,” *Proceedings of the Eleventh International Conference on Computational Structures Technology*, pp. 1-18, September, 2012, doi: 10.4203/ccp.99.68.
[38] M. N. Ab Wahab, S. Nefti-Meziani, and A. Atyabi, “A comprehensive review of swarm optimization algorithms,” *PloS one*, vol. 10, no. 5, 2015, doi: 10.1371/journal.pone.0122827.
[39] B. S. Khehra and A. P. S Pharwaha, “Comparison of genetic algorithm, particle swarm optimization and biogeography-based optimization for feature selection to classify clusters of microcalcifications,” *Journal of The Institution of Engineers (India): Series B*, vol. 98, no. 2, pp. 189-202, 2017, doi: 10.1007/s40031-016-0226-8.

BIOGRAPHIES OF AUTHORS

Budi Warsito received Bachelor Degree, Master Degree and Doctoral Degree from Gadjah Mada University, Jogjakarta, Indonesia in the field of Statistics. He currently works as lecturer at the Department of Statistics, Faculty of Science and Mathematics, Diponegoro University, Semarang Indonesia and Master Program of Information Systems, School of Postgraduate Studies, Diponegoro University. His current research interests include neural networks, wavelet, heuristic optimization and time series modelling. He has filed two patents and six Intellectual Property Rights in the fields of computation program at national level. Currently, he is also a lecturer at the Information Systems Doctoral program, School of Postgraduate Studies, Diponegoro University.

Rukun Santoso received Bachelor Degree at Department of Statistics, Faculty of Science and Mathematics, Diponegoro University, Semarang Indonesia. whereas Master Degree and Doctoral Degree were accepted from Gadjah Mada University, Jogjakarta, Indonesia in the field of Statistics. Currently, he is an associate professor at the Department of Statistics, Faculty of Science and Mathematics, Diponegoro University, Semarang Indonesia. His subject interests are about wavelet for statistics and computational statistics.

Hasbi Yasin received Bachelor Degree at Department of Statistics, Faculty of Science and Mathematics, Diponegoro University, Semarang Indonesia and Master Degree at Sepuluh Nopember Institute of Technology Surabaya Indonesia. He currently works as senior lecturer at the Department of Statistics, Faculty of Science and Mathematics, Diponegoro University, Semarang Indonesia. His current research interests include Computational Statistics and Neural Networks modelling. He has received many research grants from the Ministry of Research, Republic of Indonesia. He has filed two patents and seven Intellectual Property Rights. His current project is Spatial Analysis.