Hybrid Artificial Neural Network with Meta-Heuristics for Grid-Connected Photovoltaic System Output Prediction

Norfarizani Nordin, Shahril Irwan Sulaiman, Ahmad Maliki Omar
Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia

Article Info

Article history:

Received Jan 4, 2018
Revised Mar 15, 2018
Accepted Apr 5, 2018

Keywords:

Artificial neural network (ANN)
Cuckoo search algorithm (CSA)
Firefly algorithm (FA)
Evolutionary programming (EP)
Grid-connecetd photovoltaic (GCPV)

ABSTRACT

This paper presents the performance evaluation of hybrid Artificial Neural Network (ANN) model with selected meta-heuristics for predicting the AC output power fof a Grid-Connected Photovoltaic (GCPV). The ANN has been hybridized with three meta-heuristics, i.e. Cuckoo Search Algorithm (CSA), Evolutionary Programming (EP) and Firefly Algorithm (FA) separately. These meta-heuristics were used to optimize the number of neurons, learning rate and momentum rate such that the Root Mean Square Error (RMSE) of the prediction was minimized during the ANN training process. The results showed that CSA had outperformed EP and FA in producing the lowest RMSE. Later, Mutated Cuckoo Search Algorithm (MCSA) was introduced by incorporating Gaussian mutation operator in the conventional CSA. Further investigation showed that MSCA performed better prediction when compared with the conventional CSA in terms of RMSE and computation time.

Corresponding Author:

Norfarizani Nordin,
Faculty of Electrical Engineering,
Universiti Teknologi MARA,
40450 Shah Alam, Selangor, Malaysia.
Email: norfanordin1907@gmail.com

1. INTRODUCTION

In a Grid-Connected Photovoltaic (GCPV) system, one major question among consumers is how much output energy can be obtained from the system throughout the operation. In many cases, customers would want to know the performance of the associated GCPV system under different climatic conditions. Therefore, many studies have been conducted in order to predict the output of the system. One of the popular techniques used for prediction is by using Artificial Neural Networks (ANN) [1]-[4]. While this study had discovered many findings in PV output prediction using ANN, the ANN design was heavily dependent on past experience with the similar application and also subjected to trial and error process [5]. The trial and error process during ANN training can be very complex and time consuming. Therefore, meta-heuristic algorithms are presented in this paper to facilitate the ANN training such that the prediction of the AC output from the GCPV system is optimized.

In order to solve many types of optimization problems, meta-heuristics are used as they have better potential in solving complex optimization problems when compared to traditional algorithms [6]. Examples of meta-heuristics are Genetic Algorithms (GA), Simulated Annealing (SA), Ant Colonies Optimization (ACO), Particle Swarm Optimization (PSO), Bee Colonies Optimization (BCO), Harmony Search Algorithm (HS), Firefly Algorithm (FA), Bat Algorithm (BA) and Krill Herd (KH) [7]. Besides that, several attempts were made to facilitate the ANN design by combining the ANN models together with an meta-heuristic algorithms, such as Evolutionary Programming-Artificial Neural Network (EP-ANN) [8]-[10], Artificial Bee Colony [11], Harmony Search-Artificial Neural Network (HS-ANN) [12] and Particle Swarm Optimization-

Artificial Neural Network (PSO-ANN) [13]. Most of them are inspired by the nature, by mimicking the successful characteristics of the physical, biological and sociological systems.

Nevertheless, some of these algorithms can give better solutions to some particular problems. There are no specific algorithms in order to solve all kind optimization problems [6]. If two algorithms produce similar results but one is significantly simpler than the other, then the simpler of the two is a superior algorithm. Algorithms with a low degree of complexity have a number of advantages, including being simple to implement in an industrial setting, being simple to re-implement by researchers, and being simpler to explain and analyze [14]. The algorithmic structure of a meta-heuristic algorithm is desired to be simple enough to allow for its easy adaptation to different problems. Also, it is desired that the meta-heuristic algorithm has no algorithmic control parameters or very few algorithmic control parameters excluding the general ones, i.e. size of population, total number of iterations, problem dimension of the population based optimization algorithms. If a meta-heuristic algorithm has algorithmic control parameters, the related algorithm must not be too dependent on the initial values of the mentioned algorithmic control parameters [7].

In this paper, selected meta-heuristics, i.e. Cuckoo Search algorithm (CSA), Firefly algorithm (FA) and Evolutionary Programming (EP) were used to optimize the ANN training for predicting the power output of a GCPV system.

2. RESEARCH METHOD

This study was implemented in several stages. Firstly, a hybrid ANN models were developed for predicting the AC power output of a GCPV system. A multi-layer feedforward neural network was proposed as the architecture of the ANN. In addition, the inputs to the ANN were Solar Irradiance (SI), Ambient Temperature (AT) and Module Temperature (MT) and the output of the ANN was the AC power output. These input and out data were obtained from a GCPV system located at Green Energy Research Centre (GERC), Universiti Telknologi MARA, Malaysia. The selected meta-heuristics, i.e. Cuckoo Search Algorithm (CSA), Evolutionary Programming (EP) and Firefly Algorithm (FA) were then used separately to determine the optimal number of neurons in hidden layer, learning rate and momentum rate during the training of the ANN such that the RMSE of the prediction was minimized. Upon completion of the training process, testing process was subsequently performed to confirm the training process. The performance of these hybrid ANN models using different meta-heuristics was compared based on RMSE and computation time. Later, Mutated Cuckoo Search Algorithm (MCSA) was introduced with the aim of improving the prediction performance of hybrid ANN using CSA. Gaussian mutation was introduced as part of the optimization process durng the ANN training. The implementation of Cuckoo Search (CSA), Firefly Algorithm (FA) and Evolutionary Programming (EP) for the prediction were briefly explained in the following sections.

2.1. Cuckoo Search Algorithm (CSA)

Cuckoo Search Algorithm (CSA) is inspired by the way of laying eggs from cuckoo species [15]. A cuckoo normally lays eggs in the nest of a bird from other species. The basic principles of CSA and the conceptual implementation of the ANN using CSA are described as follows:

(i) The female cuckoo bird lays egg one at a time, and put it randomly in a host nest. Thus, each nest initially contains an egg from the host bird and an egg from the cuckoo bird. In this study, the cuckoo egg was represented by a set of decision variables that need to be optimized in predicting the AC power from the GCPV system. The decision variables used were the learning rate, momentum rate and number of neurons in hidden layer of the ANN model.

(ii) The nest with the most quality egg will survive without failure. The quality of the cuckoo egg is compared with the quality of the host egg in a particular nest. In this study, each egg also carried information on quality, i.e. the RMSE of the prediction using ANN. However, RMSE can only be obtained after simulating the ANN with the set of decision variables for that particular nest. If the quality of the cuckoo egg was better than the quality of the host egg, the cuckoo egg was set to survive in the host nest, and vice versa.

(iii) The number of host nests is fixed with the probability of cuckoo's egg being discovered by the host bird, P_a is from 0 to 1[16]. If the cuckoo egg is discovered by the host bird, the cuckoo egg is destroyed or thrown away by the host bird. This event was expected to occur by chance with probability P_a. If the cuckoo egg was found to be destroyed, new nest locations are identified using

$$x_t^{t+1} = x_t^{(t)} + \alpha \oplus Levy(\lambda) \quad (1)$$

where x_t^{t+1} is the new egg at a new nest location for a cuckoo i and α denoted as a step size. The product ⊕ means entry wise multiplications [17]. The L´evy flight essentially provides a random walk while the random step length is drawn from a L´evy distribution:

$$Levy{\sim}u = t^{-\lambda}, \tag{2}$$

where λ denoted as the random walk and the value was set between 1 to 2.

The flowchart of CSA was illustrated in Figure 1. First, cuckoo search parameters and the initial host nest are defined. Next, the fitness of each cuckoo is evaluated before it was ranked. After evaluation, the host nest is modified using Levy flight equation as shown in Equation 1 and the fitness for each modified cuckoo is evaluated. Next, if the condition is not satisfied, the cuckoos are moved towards the best nest environment and the process will be repeated from the beginning. On the other hand, if the condition is satisfied, choose the current best nest as the best cuckoo until the population exceeds the maximum generation, and lastly, the evolution process will be stopped.

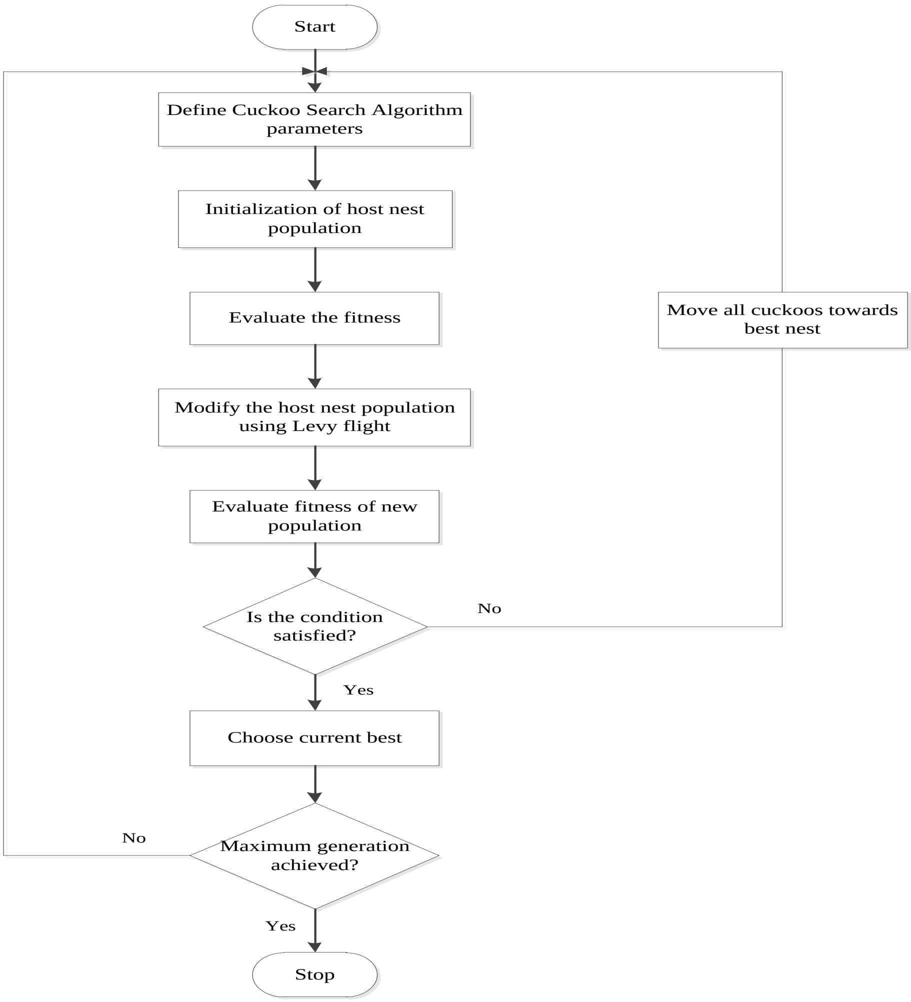
Start
Define Cuckoo Search Algorithm parameters
Initialization of host nest population
Evaluate the fitness
Move all cuckoos towards best nest
Modify the host nest population using Levy flight
Evaluate fitness of new population
Is the condition satisfied?
No
Yes
Choose current best
No
Maximum generation achieved?
Yes
Stop

Figure 1. Flowchart of Cuckoo Search Algorithm

2.2. Evolutionary Programming (EP)

EP is one of meta-heuristic technique which is used to perform random search in optimizing an objective function. It had been used in many numerical and combinatorial optimization problems in recent years. It is a combination of several main processes namely initialization of parents, evaluation of the fitness value, mutation process to produce an offspring from their parents, evaluation of fitness for the offspring, a combination process, selection and lastly the convergence test. In brief, there are two major steps to be summarized in optimization by EP [18]:

(i) Mutate parents in the current population. The mutation of a parent was conducted using Gaussian mutation as below:

$$q^{'}_{1j} = q_j + \sigma . N_j(0,1) \tag{3}$$

where q'_{1j} were the offsprings generated from each parent by Gaussian mutation. $N_j(0,1)$ represents a normal Gaussian random variable with mean 0 and standard deviation 1. In this study, each parent contains information of the decision variables that need to be optimized, i.e. learning rate, momentum rate and number of neurons in hidden layer.

(ii) Select the next generation from the parents and mutated parents (offspring). At this stage, each parent is mutated to produce an offspring. Then, the next generation of candidates for potential solution is selected by first ranking the pool of parents and offspring according to their fitness values. Subsequently, a population of candidates is transcribed to the next generation for the next evolution. In this study, the fitness value was RMSE.

The flowchart of EP was illustrated in Figure 2. First, initialize the parent population of the generation. Second, the fitness of each parent generated is evaluated using the selected equation or function. Next, perform the mutation process by using Gaussian distribution operator as shown in Equation 3 to produce the new population. The new population is known as the offspring. After that, combine all parents and offsprings in order to find the best result by undergoing the selection process. The process is stopped when the convergence is achieved. If not, the process will be repeated by performing the mutation process again.

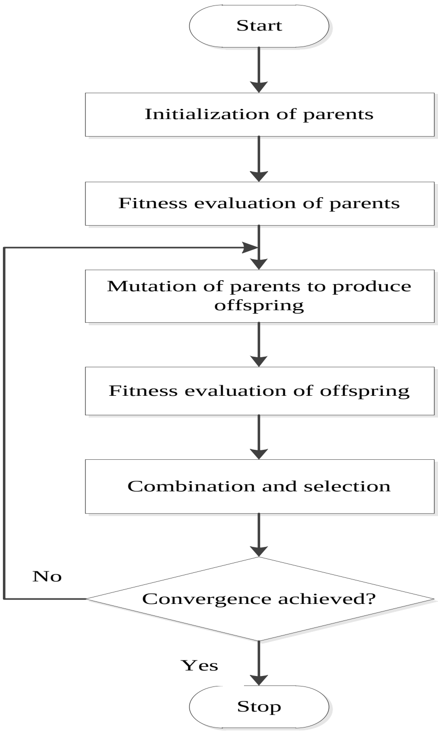
Start
Initialization of parents
Fitness evaluation of parents
Mutation of parents to produce offspring
Fitness evaluation of offspring
Combination and selection
No
Convergence achieved?
Yes
Stop

Figure 2. Flowchart of Evolutionary Programming

2.3. Firefly Algorithm (FA)

Another type of meta-heuristic algorithm selected in this study was Firefly Algorithm (FA). FA is a meta-heuristic optimization algorithm which was introduced at Cambridge University in 2008 by Xin-She Yang [19]. The algorithm was inspired by the flashing characteristics of fireflies at night. The algorithm was formulated based on three idealized rules [15]:

(i) All fireflies are unisex. Thus, one firefly will be attracted to the other fireflies regardless of their sex. In this study, besides being unisex, every firefly generated contains a set of decision variables that need to be optimized in predicting the AC power from the GCPV system. The decision variables used were the learning rate, momentum rate and number of neurons in hidden layer of the ANN model.

(ii) Attractiveness is proportional to the brightness of fireflies. However, the brightness decreases as the distance increases. For any two flashing fireflies, firefly with less brightness will fly towards the brighter one and if there is no brighter one than a particular firefly, the firefly will fly randomly. Therefore, the lower the distance between two fireflies, better agreement on the solution is obtained. On the other hand, random flight by a firefly was implemented by randomly adjusting the value of each decision variables.

(iii) The brightness of firefly represents the quality of solution and it is associated to the fitness value of the objective function. In this study, the brightness of the firefly was represented by the fitness value, i.e. the RMSE. Therefore, the firefly with less brightness had higher RMSE while the firefly with more brightness had lower RMSE.

The flowchart of FA was illustrated in Figure 3. First, generate the initial population of the firefly and evaluate the fitness of each fireflies generated by using the selected equation or function. After that, update the light intensity of the firefly by using light intensity equation. Next, the fitness of all the fireflies is evaluated and ranked acoording to the corresponding fitness value. If the maximum iteration was reached, the process will proceed by finding the global best throughout all the generations of the firefly. The new population of firefly from the best generation will be used to represent the best population.

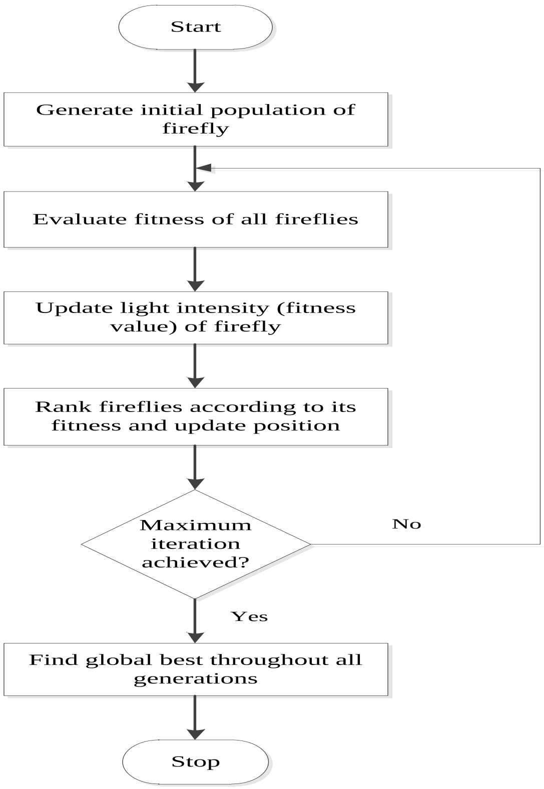
Start
Generate initial population of firefly
Evaluate fitness of all fireflies
Update light intensity (fitness value) of firefly
Rank fireflies according to its fitness and update position
Maximum iteration achieved?
No
Yes
Find global best throughout all generations
Stop

Figure 3. Flowchart of Firefly Algorithm

2.4. Mutated Cuckoo Search Algorithm (MCSA)

In this study, the performance of CSA was improved by modifying the optimization process in CSA. The conventional CSA commonly employs Levy flight to update the location of the nest during the evolution of cuckoo. However, the random walk of cuckoo using Levy flight is very random in nature and thus causing the cuckoo to divert far away from the possible optimal solutions during the search for optimal solution. This random walk can be improved by introducing a characteristic of inheritance such that the movement of cuckoo is still dependent on the group performance of cuckoos in the population.

As a result, mutation was introduced such that the cuckoo was mutated after a random walk using Levy flight. In MCSA, Levy flight will be combined with the Gaussian distribution operator to perform the mutation process. Meanwhile, the rest of the processes are similar with the process of CSA. Hence, the group performance of cuckoo was preserved during the evolution of cuckoo. Eventually, a better optimal solution can be found. In short, the overall process for MCSA is illustrated as in the following flowchart.

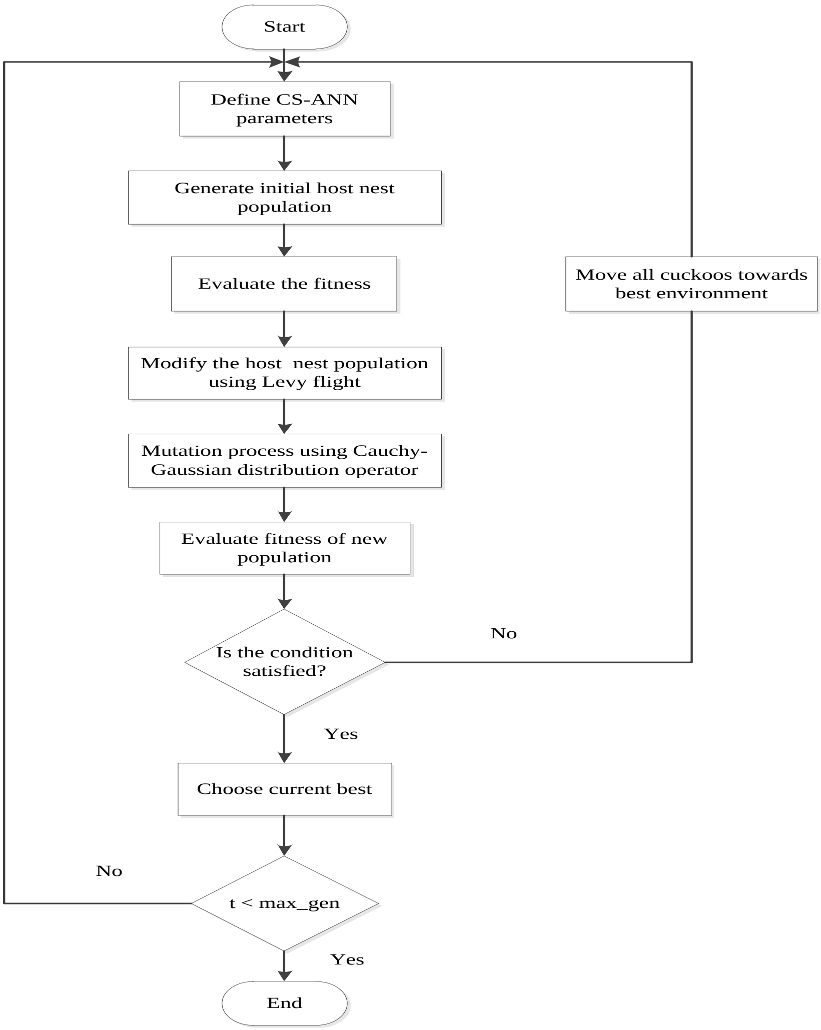
Start
Define CS-ANN parameters
Generate initial host nest population
Evaluate the fitness
Modify the host nest population using Levy flight
Mutation process using Cauchy-Gaussian distribution operator
Evaluate fitness of new population
Is the condition satisfied?
No
Move all cuckoos towards best environment
Yes
Choose current best
No
t < max_gen
Yes
End

Figure 4. Flowchart of Mutated Cuckoo Search algorithm

3. RESULTS AND DISCUSSIONS

This section presents the results and discussion of the study. The performance comparison of hybrid ANN models, i.e. CSA-ANN, EP-ANN and FA-ANN is shown in Section 3.1. After that, the performance of MCSA-ANN is shown in Section 3.2. The results achieved by MCSA-ANN was evaluated and benchmarked with CSA-ANN.

3.1. Performance Comparison of Hybrid Artificial Neural Network Models

After all the ANN training parameters had been determined for the models, the performance of the hybrid ANN models was compared in terms of prediction accuracy based on RMSE and R^2 as shown in Table 1. CSA-ANN was discovered to be the best model as it produces the lowest RMSE during both training and testing. The RMSE during training produced by CSA-ANN is approximately 43.60% lower than FA-ANN and 38.26% lower than EP-ANN. Meanwhile, the RMSE during testing produced by CSA-ANN is approximately 47.55% lower than FA-ANN and 42.86% lower than EP-ANN. Moreover, CSA-ANN also yields the highest R^2 during both training and testing when compared with FA-ANN and EP-ANN. Besides that, CSA-ANN was also approximately 5.3 and 5.6 times faster than EP-ANN and FA-ANN respectively.

Table 1. Performance comparison of Hybrid ANN models

Parameters	CSA-ANN	EP-ANN	FA-ANN
Optimal no. of neurons in hidden layer	5	20	9
Optimal population size	20	60	60
Optimal learning rate	0.2393	0.6973	0.5655
Optimal momentum rate	0.7244	0.7166	0.6904
Activation function in hidden layer	LOGSIG	LOGSIG	LOGSIG
Learning algorithm	TRAINLM	TRAINLM	TRAINLM
RMSE training, in Watt	12.3463	19.9988	21.8907
R^2 training	0.9988	0.9770	0.9755
RMSE testing, in Watt	14.2718	24.9776	27.2113
R^2 testing	0.9899	0.9734	0.9753
Computation time, in minutes	21.2568	112.7094	119.4736

3.2. Performance Comparison of Mutated Cuckoo Search Algorithm

Although CSA was found to be the best meta-heuristics for the hybrid ANN model, the performance of the conventional CSA depends on random walk using Levy flight which is very random in nature. Thus, it may cause the cuckoo to divert far away from the possible optimal solutions during the search for optimal solution. Therefore, a Mutated Cuckoo Search Algorithm-Artificial Neural Network (MCSA-ANN) was developed to improve the performance of the conventional CSA. The MCSA-ANN was later used to improve the hybrid ANN for the prediction task. The performance of MCSA-ANN and CSA-ANN was compared in Table 2. MCSA-ANN had outperformed CSA-ANN by producing lower RMSE and higher R^2 during both training and testing. The MCSA-ANN was also slightly faster than CSA-ANN. In short, the introduction of MCSA-ANN for the prediction was justified shown by the superior performance obtained during the testing and training process. The MCSA-ANN had exhibited the optimal number of neurons in hidden layer of 4, while the optimal values for learning rate and momentum rate are 0.6740 and 0.7965 respectively. This algorithm had also yielded lowest RMSE of 11.0534 W during training and 13.1348 W during testing. Other than that, it had produced maximum R^2 with 0.9994 and 0.9913 for both training and testing, with the least computational time of 17.9616 minutes when compared to other hybrid ANNs.

Table 2. Performance comparison of Mutated Cuckoo Search Algorithm

Parameters	CS-ANN	MCS-ANN
Optimal no. of neurons in hidden layer	5	4
Optimal learning rate	0.2393	0.6740
Optimal momentum rate	0.7244	0.7965
Activation function in hidden layer	LOGSIG	LOGSIG
Learning algorithm	TRAINLM	TRAINLM
RMSE training, in Watt	12.3463	**11.0534**
R^2 training	0.9988	**0.9994**
RMSE testing, in Watt	14.2718	**13.1348**
R^2 testing	0.9899	**0.9913**
Computation time, in minutes	21.2568	**17.9616**

4. CONCLUSION

The performance of selected hybrid ANN models was compared. The results showed that CSA-ANN had outperformed EP-ANN and FA-ANN by producing the lowest RMSE and computation time. Apart from that, the introduction of MCSA-ANN showed that MCSA-ANN had outperformed CSA-ANN by exhibiting the lowest RMSE during training and testing as well as the highest R^2 during training and testing. In terms of computation time, MCSA-ANN needed less computation time than CSA-ANN. As a conclusion, the cability of MCSA-ANN for the output prediction was justified.

ACKNOWLEDGEMENTS

This work was supported in part by the Fundamental Research Grant Scheme (FRGS), Ministry of Education (Ref: 600-RMI/FRGS 5/3 (120/2015) and Universiti Teknologi MARA (UiTM) Malaysia.

REFERENCES

[1] F. Hashim, *et al.*, "Prediction of rainfall based on weather parameter using artificial neural network," *Journal of Fundamental and Applied Sciences*, vol. 9, pp. 493-502, 2017.

[2] S. Sulaiman, *et al.*, “Performance Analysis of Evolutionary ANN for Output Prediction of a Grid-Connected Photovoltaic System,” *Proceedings of World Academy of Science: Engineering & Technology*, vol. 53, 2009.

[3] N. Nordin, *et al.*, “Prediction of AC power output in grid-connected photovoltaic system using Artificial Neural Network with multi-variable inputs,” in *Systems, Process and Control (ICSPC), 2016 IEEE Conference on*, pp. 192-195, 2016.

[4] S. Supian, *et al.*, “Mathematical Model for Dissolved Oxygen Prediction in Cirata Reservoir, West Java by Using Artificial Neural Network,” *Journal of Fundamental and Applied Sciences*, vol. 10, pp. 66-78, 2018.

[5] X. Yao, “Evolving artificial neural networks,” *Proceedings of the IEEE*, vol. 87, pp. 1423-1447, 1999.

[6] A. Ouaarab, *et al.*, “Discrete cuckoo search algorithm for the travelling salesman problem,” *Neural Computing and Applications*, pp. 1-11, 2013.

[7] P. Civicioglu and E. Besdok, “A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms,” *Artificial Intelligence Review*, vol. 39, pp. 315-346, 2013.

[8] S. I. Sulaiman, *et al.*, “Evolutionary programming versus artificial immune system in evolving neural network for grid-connected photovoltaic system output prediction,” *WSEAS Transactions on Systems and Control*, vol. 6, pp. 197-206, 2011.

[9] S. I. Sulaiman, *et al.*, “Hybridization of Meta-Evolutionary Programming and Artificial Neural Network for predicting grid-connected photovoltaic system output,” in *TENCON Spring Conference, 2013 IEEE*, pp. 445-449, 2013.

[10] S. A. Shaaya, *et al.*, “Immunized-evolutionary algorithm based technique for loss control in transmission system with multi-load increment,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 6, pp. 737-748, 2017.

[11] J. Shahrabi and S. M. Khameneh, “Development of a Hybrid System of Artificial Neural Networks and Artificial Bee Colony Algorithm for Prediction and Modeling of Customer Choice in the Market,” *Journal of Fundamental and Applied Sciences*, vol. 9, pp. 154-183, 2017.

[12] N. Kassim, *et al.*, “Harmony search-based optimization of artificial neural network for predicting AC power from a photovoltaic system,” in *Power Engineering and Optimization Conference (PEOCO), 2014 IEEE 8th International*, pp. 504-507, 2014.

[13] T. N. Hussain, *et al.*, “A hybrid artificial neural network for grid-connected photovoltaic system output prediction,” in *Computers & Informatics (ISCI), 2013 IEEE Symposium on*, pp. 108-111, 2013.

[14] J. Silberholz and B. Golden, “Comparison of metaheuristics,” in Handbook of Metaheuristics, ed: Springer, pp. 625-640, 2010.

[15] X. S. Yang, “Nature-inspired metaheuristic algorithms,” Luniver press, 2010.

[16] X. S. Yang and S. Deb, “Cuckoo search via Lévy flights,” in *Nature & Biologically Inspired Computing, NaBIC 2009. World Congress on*, pp. 210-214, 2009.

[17] X. S. Yang and S. Deb, “Engineering optimisation by cuckoo search,” *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, pp. 330-343, 2010.

[18] X. Yao, *et al.*, “Evolutionary programming made faster,” *IEEE Transactions on Evolutionary computation*, vol. 3, pp. 82-102, 1999.

[19] X. S. Yang, “Nature-inspired optimization algorithms,” Elsevier, 2014.