

Particle Swarm Optimization to Obtain Weights in Neural Network

Budi Warsito, Hasbi Yasin and Alan Prahutama

Department of Statistics, Faculty of Science and Mathematics
Diponegoro University, Jl. Prof. Soedarto, SH Tembalang Semarang, Indonesia

*Corresponding author: budiwrst2@gmail.com

Article history

Received: 14 May 2019

Received in revised form: 21 August 2019

Accepted: 28 August 2019

Published online: 1 December 2019

Abstract This research discusses the use of a class of heuristic optimization to obtain the weights in neural network model for time series prediction. In this case, Feed Forward Neural Network (FFNN) was chosen as the class of network architecture. The heuristic algorithm determined to obtain the weights in network was Particle Swarm Optimization (PSO). It is a non-gradient optimization technique. This method was used for optimizing the connection weights of network. The lags used as the input were selected based on the strong relationship with the current. The eight architectures were conducted to improve the accuracy of the neural network model. In each architecture, we repeated the running thirty times to get the statistics of mean and variance. The comparison of the performance of various architectures based on the minimum MSE and the stability of the results is presented in this paper. The optimal number of neurons in hidden layer was determined by these criteria. The proposed procedure was applied in air pollution data, i.e. Solid Particulate Matter (SPM). The results showed that the proposed procedure gave promising results in terms of prediction accuracy. A few neurons in hidden layer are strongly recommended in choosing the optimal architecture.

Keywords PSO; neural network; time series; air pollution prediction; SPM.

Mathematics Subject Classification 68T20, 65K10.

1 Introduction

In recent years neural network modeling has been an interesting open problem and widely applied in many yields because of its good generalization performance. It has an activation function or a certain kind of nonlinear transfer function that allows for running various data analysis, including time series data analysis. The modeling algorithm also enables to select many kinds of optimization methods to get outputs that minimize error. The construction of the network architecture is also very flexible, so there are many opportunities for advanced investigation. Such flexibility includes the procedure of selecting input, determining the number of hidden units, choosing the activation function and the optimization method to be used to

obtain the optimal weights. Various researches related to the procedure of modeling, selecting inputs, determining the number of hidden units and also the theoretical and convergence analysis have been developed. Furthermore, the question is whether the optimal weights obtained from a certain optimization method can reach the global optimum or only a local optimum.

Various optimization methods have been attempted to obtain the optimal weights in neural network. In many cases, a certain optimization method used oftentimes is incapable of getting the global optimum. The estimation results are often trapped in local optimum early [1]. Besides, it is also difficult to use iteration method to determine the initial values to get convergence. In conventional optimization methods i.e. gradient based methods, the results are often inconsistent. Hereafter, alternative methods to handle this problem have been rapidly developed, including genetic algorithm, simulated annealing, ant colony and particle swarm optimization (PSO). In a previous research, genetic algorithm has been successfully attempted for some typical time series data, taken the shine off from some gradient based methods [2]. However, the main problem of this algorithm is it requires high computational cost and much time to get stable results. In fact, particle swarm optimization requires no complicated evolutionary operators, fewer parameters to adjust, and it is easy to implement [3]. In this research, the use of particle swarm optimization to obtain the optimal weights in neural network modeling in time series prediction has been developed. Some previous researches have also used particle swarm optimization for neural network [4-7]. Nonetheless, many of them used trial and error method for selecting the best architecture and conducted based on the values of only one experiment. In this research, a simple architecture was developed first. The incremental of neuron was added one by one until reaching the desired number. In each architecture, we repeated the running several times to get the statistics of mean and variance. The stability of the results served as the basis of choosing the best architecture, i.e. the optimal number of neurons in hidden layer. A comparative analysis was also carried out using different activation functions at the hidden layer. We compared logistic sigmoid and bipolar sigmoid as activation functions to get the optimal architecture. The proposed procedure was applied in predicting the concentration of air pollution, i.e. Solid Particulate Matter (SPM). Several previous studies have also applied neural network in predicting the concentration of air pollutions in various fields ([8-12] and [13-15]).

Predicting air pollution data is important because environmental problem may be the most severe problem which greatly influences human health and ecosystems. Air quality evaluation is an important way to monitor and control air pollution [16]. The formation of air pollutants is a very complex and nonlinear phenomenon [17]. The impact of air pollution on health is very complex as there are different sources and their individual effects vary from one to another [18]. Air pollution degrades air quality and leads to several diseases, such as wheezing, eye irritation, bronchitis, and asthma. Premature death due to SPM is reported to be very high [19,20]. In recent years, rapid headway has been made in the prediction of the concentration of air pollution. However, it is still challenging to accurately predict the concentration of air pollution due to complex influential factors. This way, it is important to obtain more effective methods to accurately predict the concentration of air pollutions.

2 Feed Forward Neural Network

Artificial Neural Network (ANN) or often abbreviated as NN, is a modeling algorithm inspired by biological neural network. Feed Forward Neural Network (FFNN) is the main class and the most popular neural network modelling. It uses back propagation algorithms for the learning process. FFNN is also often called Multilayer Perceptron (MLP) because its network architecture consists of several layers. In this network architecture, hidden layer(s) are added between the input layer and the output layer. FFNN is the most widely used neural model in many practical applications [21]. It consists of a (possibly large) number of simple neuron-like processing units, organized in layers. Each unit (also often called neuron or node) in hidden layer is connected with all the units in the input layer, and every unit in the output layer is connected with all the units in the hidden layer. These connections represent the strength of relationship between each unit.

The strength of the connections is explained as weights. Each connection may have a different weight based on the strength of the connection itself. The number of units in each layer determines the complexity of the network. More complex network requires more weights to be obtained. Determining the number of units in input layer and output layer depends on the task of a network. For example, in the case of classification problem, the input is determined first. It is determined by the knowledge of the practical applications. Likewise, in the case of time series modeling the input unit is the lagged variables selected according to the strength of relationship to the output. Meanwhile, the fundamental of univariate or multivariate analysis is used to determine the number of units in output layer. In this case, the application of FFNN for univariate time series modeling evolves. Therefore, the number of neurons in output layer is one and the output is

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

where f^o is the activation function in output layer, and f is the activation function in output layer. If bias is added at the input layer and the activation function of each neuron at hidden layer is f^h then equation (1) becomes

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

where w^b is the weights from bias to neuron output and w_j^b is the weights from bias to hidden layer. In time series modeling, input is the past series for predicting the current value x_t . Hence, if input is lagged values of 1 until p , or x_{t-1}, \dots, x_{t-p} , equation (2) becomes:

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

Configuration of FFNN architecture for time series is as follows. The network consists of input neuron X_{t-1} until X_{t-p} , one bias unit, one hidden layer containing n neuron and one output neuron is represented in Figure 1.

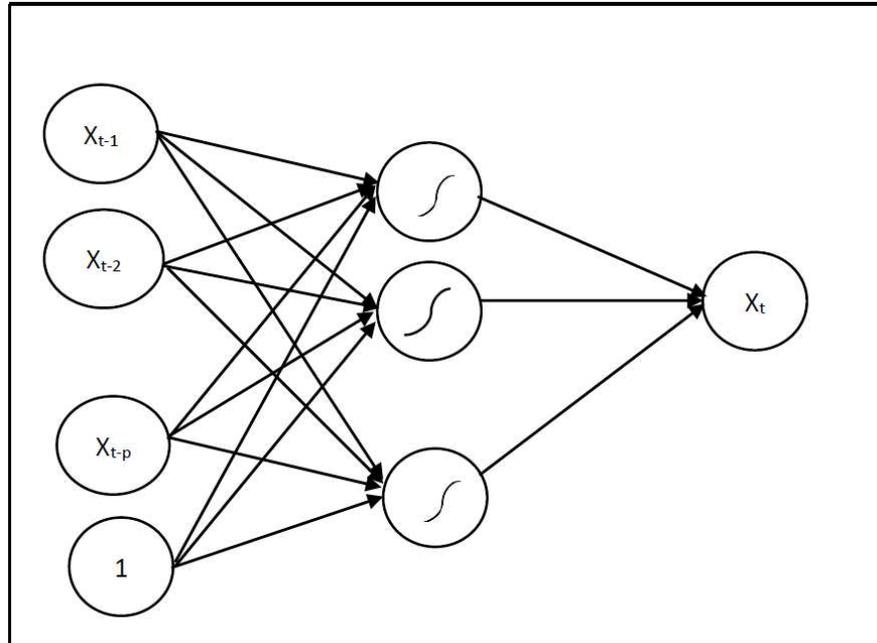


Figure 1: FFNN Architecture for Time Series Modeling

There are three stages of back propagation algorithm in neural network modeling, i.e. feed forward, error calculation and adjusting the weights. After the feed forward stage, calculating the network error is carried out by the differencing between output and target. Adjusting the weights is carried out by updating the weights using a certain optimization technique. The feed forward stage is then performed again by applying the new weights. This process continues until the stopping criteria are reached, either minimum error or maximum epochs. For example, if we have weights vector w , the length of p , and the objective e , then the updating process is carried out by the minimum of e as a goal. In this research, the weights updating is carried out by using particle swarm optimization algorithm. The main objective of PSO in Feed Forward Neural Network modeling is to get the best particle position from a group of particles which are either moving or trying to move towards the best solution [22].

3 Particle Swarm Optimization

Swarm Intelligence is a scientific discipline of intelligent systems related to natural and artificial systems, which consists of many individuals or population who coordinate using the concept of decentralization control or social intelligence in groups and self-organized (personal intelligence or independently organized). Swarm Intelligence logic then inspires many researchers to develop algorithms based on animal intelligence, especially to overcome optimization problems. Various algorithms regarding collective behavior have been proposed to overcome optimization problems. Optimization problems lead to research to determine the minimum or maximum value of a function. One of the swarm intelligence algorithms is Particle Swarm Optimization (PSO). It is one of the optimization techniques in metaheuristic group. Particle Swarm Optimization is an optimization technique inspired by social behavior and movement dynamics in flocks of insects, birds and fish that form certain formations without colliding with each

other. PSO is also well-known as population-based search method based on the behavior of elements in nature such as fish schooling and birds flocking [23]. It was originally proposed by James Kennedy and Russell Eberhart in 1995 for optimization of continuous nonlinear functions [24-25] and also has been applied in machine learning [26].

In PSO, the process starts with a randomly generated swarm of particles [27]. Each particle is then evaluated for its quality using the fitness function. Furthermore, the particle follows its leader which is affected by the best-position of each particle in the whole swarm. Each bird is described as a particle representing the solution to a problem that has a position and velocity. PSO has two main functions, namely velocity updates and position updates. The particles used to update the velocity and position will continue to accelerate near the position of the previous best particles, and the best globally until the minimum error conditions are reached. PSO starts by initializing a random set of particles as the solutions. PSO calculations are formulated as follows:

$$v_{i,j}^{t+1} = \rho v_{i,j}^t + c_1 r_1 (pBest_{g,j}^t - x_{i,j}^t) + c_2 r_2 (gBest_{g,j}^t - x_{i,j}^t) \quad (4)$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^t \quad (5)$$

where

- ρ : inertia weight
- c_1 & c_2 : acceleration coefficients
- r_1 & r_2 : random value from continuous uniform distribution
- $pBest_{g,j}^t$: best position from particle i which is dimension j at iteration t
- $gBest_{g,j}^t$: global optimum from particle g which is dimension j at iteration t
- $x_{i,j}^t$: position of particle i which is dimension j at iteration t

In the processing of updating the speed and position of the particles, inertia weights are always updated in each iteration. This weight update is referred to as Time Variant Inertia Weight (TVIW). The renewal of inertia weight is explained in the following equation:

$$\rho = \rho_{\max} - \frac{\rho_{\max} - \rho_{\min}}{t_{\max}} \times t \quad (6)$$

where

- ρ : inertia weight
- ρ_{\max} : upper bound of ρ , commonly $\rho_{\max} = 0.9$.
- ρ_{\min} : lower bound of ρ , commonly $\rho_{\min} = 0.4$.
- t_{\max} : the number of maximum iterations
- t : current iteration

Time Varying Acceleration Coefficients or TVAC, in this case are c_1 and c_2 , will strive to increase global search for optimization so that particles will move towards optimal global [18]. The values of c_1 and c_2 are usually equal to two, respectively, so the multiplication results of the $c_1 \times r_1$ and $c_2 \times r_2$ ensure that the particles will approach the target of about half the difference.

In this research, PSO optimization is done to provide the optimal FFNN weights so the prediction results can have a smaller error. These parameters will be optimized using the PSO iteration then selected based on the smallest MSE. The initial position of the particle is

randomly generated, while the initial velocity for all the particles is given a zero value because the position of the particle has no movement. Determining the fitness value at the initial position of the PSO particles is very important. This value is used in determining the best individual position ($pBest$) and the best global position ($gBest$). The fitness value used in this study was a function that minimized the value of Mean Square Error (MSE). Systematically, the stages of PSO algorithm for optimizing FFNN weights are as follows:

1. Determine the initial value including the number of particles, maximum iteration, acceleration coefficient and inertia weights of equation (6).
2. Determine the initial position and initial velocity of each particle randomly. Initial position includes all weights in the network.
3. Calculate the network output by using the weights in the initial position.
4. Calculate the fitness value of each particle and then choose the optimal fitness (the minimum MSE).
5. Choose the $pBest$ (best position) based on the fitness value. The chosen $pBest$ of each iteration becomes the $gBest$ (best global).
6. Update velocity and particle position by using equation (4) and equation (5).
7. Determine the new position after updating velocity and position.
8. Calculate the network output by using the weights at the new position.
9. Back to 4.

These stages were carried out until the stopping criteria were met.

4 Application in SPM Data

The data used in this research were the monthly data of Solid Particulate Matter (SPM) in Semarang City, Central Java Indonesia from January 2008 to December 2017. The length of the data was 120 and divided into two parts, the first 100 data as training and the remaining 20 as testing. The input model was identified by using the correlation between the past series and the current. The significant values of the correlation became the lagged values as candidates of the input. From the identification, the lagged values of 1 and 2 were selected as input. The number of neurons in hidden layer was determined by choosing the simple one, and then incremental of neuron was added one by one until reaching the desired number. In this research, we began with one neuron and continued until eight neurons. In each architecture, we repeated the running thirty times to get the statistics of mean and variance. The stability of the results served as the basis of choosing the best architecture, i.e. the optimal number of nodes in hidden layer. In order to obtain better result and for comparison purpose, logistic sigmoid and bipolar sigmoid were tried as activation function in the hidden layer. The procedure was applied in air pollution data, i.e. Solid Particulate Matter (SPM).

In each experiment, the maximum number of iterations was 500, and the population size (swarm size) was 10. The PSO parameters are determined as follows: Inertia Weight = 1, Inertia Weight Damping Ratio = 0.99, Personal Learning Coefficient $c_1 = 1.5$, and Global Learning Coefficient $c_2 = 2.0$. Plot of the Best Cost from 500 iterations in one experiment is seen in Figure 2.

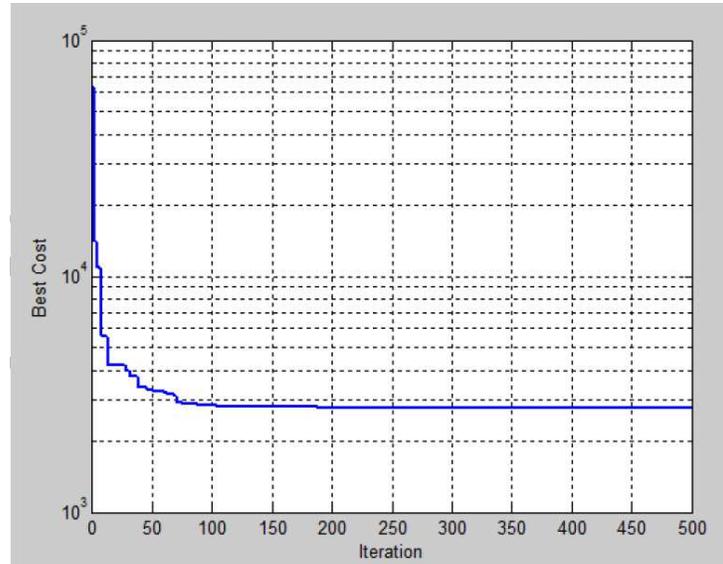


Figure 2: Plot of Iteration in One Experiment

In each experiment, the pattern resulted after 500 iterations gave a similar result with the plot in Figure 2. In the beginning of the iteration, the Best Cost was still very high, declined rapidly and after a few iterations the result reached a small value. After about 100 iterations, the result became stable. After 200 iterations, a more stable result was obtained until 500 iterations. This shows that 500 iterations are enough to reach the convergence and repeating with the same number of iterations is also reasonable. In each architecture, the repeating process was carried out 30 times. Statistics of mean and variance for the 30 results became the basis of choosing the best architecture. The experiment results are shown in Table 1.

Table 1: Experimental Results of FFNN-PSO for SPM Data

Hidden unit	Logistic sigmoid				Bipolar sigmoid			
	MSE train		MSE testing		MSE train		MSE testing	
	Mean ($\times 10^3$)	Var ($\times 10^4$)	Mean ($\times 10^3$)	Var ($\times 10^4$)	Mean ($\times 10^3$)	Var ($\times 10^4$)	Mean ($\times 10^3$)	Var ($\times 10^4$)
1	2.6516	1.0550	1.9040	0.1670	2.79141	0.0057	1.99125	0.1964
2	2.8226	1.2227	2.1746	0.3186	2.8196	1.0811	2.33447	7.3242
3	2.7190	2.6240	2.3119	10.464	2.7049	1.7884	2.2395	12.548
4	2.7877	2.9495	2.1537	3.9142	2.7831	0.9505	2.3223	4.7812
5	2.6616	3.2345	2.1999	2.7193	2.6344	1.3527	2.2219	1.7759
6	2.8275	17.896	2.5163	61.160	2.6284	3.4819	2.2376	5.2458
7	2.5981	3.3636	3.2543	28.216	2.5923	2.2019	2.2415	4.9229
8	2.6086	1.2571	2.3722	6.2759	2.6561	2.7933	2.3281	6.5105

The results of Table 1 show that the averages of MSE in all architectures were almost the same. The goodness result of the in-sample prediction in a certain architecture did not ensure the goodness in the out-sample prediction. This indicates that the use of various types of

architecture is of no difference. However, if we pay attention to the variance, the significant difference of each result was more visible. The network architecture with small enough hidden unit gave the best result, in both in-sample and out-sample predictions. This was found at the network with two different activation functions. Therefore, the network architecture with one hidden unit was chosen as the best architecture. Figure 3 shows the final architecture. There are two neurons in the input layer, one bias, one neuron in hidden layer and one neuron in output layer. Totally, there are five interconnection weights in the network. The final weights obtained by PSO from input to hidden layer were (0.6385, 0.1138), from bias to hidden layer was 0.7830, from bias to output was -0.6583 and from hidden layer to output was 1.1948.

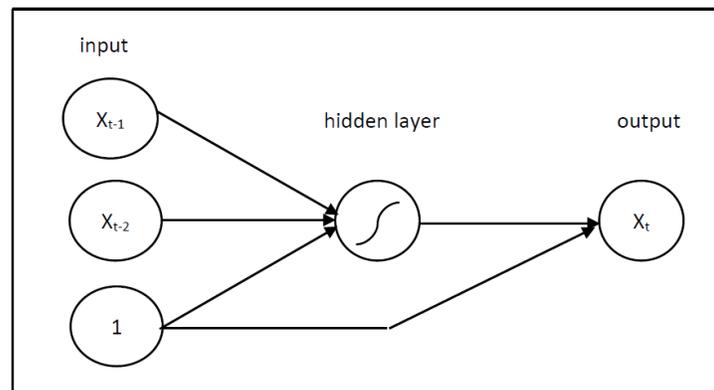


Figure 3: Final Architecture

Generally, the use of different activation function did not influence the result. For identifying the goodness of the prediction, comparison of the actual data and the prediction result in one of the experiments is reported in Figure 4.

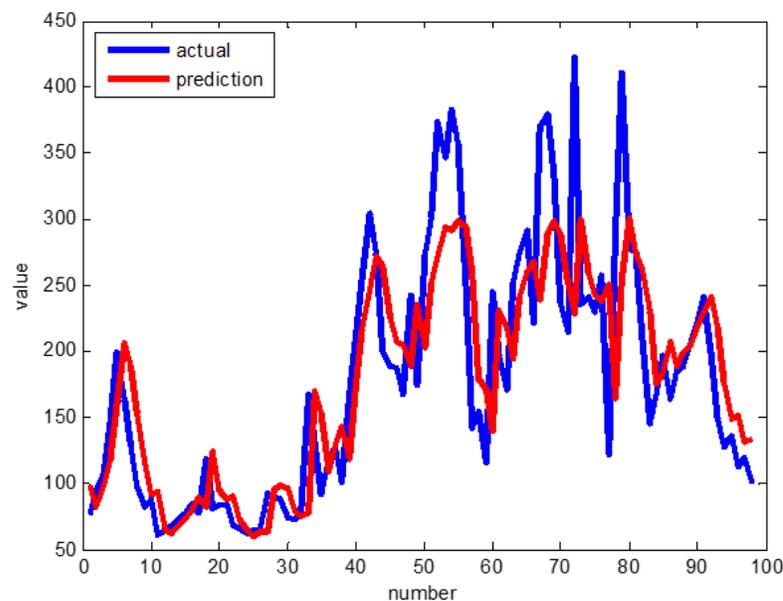


Figure 4: Plot of Actual and In-sample Prediction of SPM Data

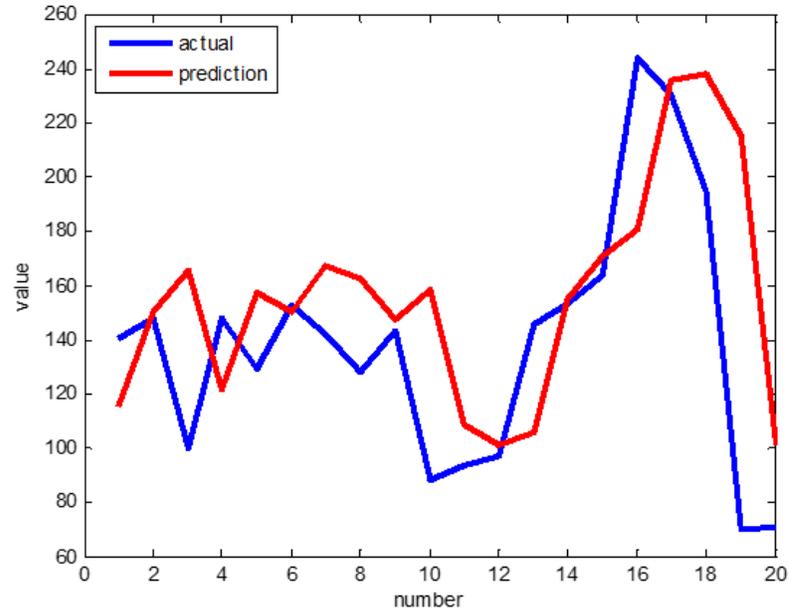


Figure 5: Plot of Actual and the Out-sample Prediction of SPM Data

The comparison of the in-sample prediction and the actual as in Figure 5 shows that the proposed method gave a good prediction result. It can be seen from the pattern of the prediction that followed the pattern of the actual. We can conclude that the prediction was closest to the actual and as a consequence, the error line was closest to zero. This result is similar with the results of a study by Xu and Pei [7]. A few big errors were found at the points which had extreme values compared with the one step before. A similar condition was also found at the out-sample prediction, especially for short time prediction. In NN-PSO for prediction model as in this research, the selection of network parameters affected the performance of the system. Adjustment of parameter can be done as well as to achieve better enhancement. This is in line with a study by [23].

5 Conclusion

The procedure of choosing optimum architecture of neural network modeling with particle swarm optimization in time series prediction has been developed. Stability analysis of the prediction results is the basis of choosing the best one. Various network architectures have been built. The weights in each architecture are obtained by particle swarm optimization, and the process is repeated many times. Statistics of mean and variance of the MSE of training and testing are observed. The investigation is done by comparing the accuracy resulted by various architectures. The average shows only small error value occurs and the variance shows consistent estimation results. The results show that NN-PSO is applicable for predicting the SPM data. The comparison between PSO and other heuristic optimization techniques or the hybrid with the gradient based methods in optimizing NN can be done as the future works.

References

- [1] Montana. Neural network weight selection using genetic algorithms. *Advances in Neural Information, Processing Systems*. 1994. 4: 1110–1117.
- [2] Warsito, B. *Wavelet Neural Network Modeling with Genetic Algorithm*. Ph.D. Thesis. Gadjah Mada University. 2016.
- [3] Ling, Q.H., Song, Y.Q., Han, F., Zhou, C.H. and Lu, H. An improved learning algorithm for random neural networks based on particle swarm optimization and input-to-output sensitivity. *Cognitive Systems Research*. 2019. 53: 51–56.
- [4] Lu, W.Z. and Xue, Y. Prediction of particulate matter at street level using artificial neural networks coupling with chaotic particle swarm optimization algorithm. *Building and Environment*. 2014. 78: 111–117.
- [5] Lu, W.Z., Fan, H.Y., Leung, A.Y.T. and Wong, J.C.K. Analysis of pollutant levels in central Hong Kong applying neural network method with particle swarm optimization. *Environmental monitoring and assessment*. 2002. 79(3): 217–230.
- [6] Ren, C., An, N., Wang, J., Li, L., Hu, B. and Shang, D. Optimal parameters selection for BP neural network based on particle swarm optimization: A case study of wind speed forecasting. *Knowledge-Based Systems*. 2014. 56: 226–239.
- [7] Xu, J. and Pei, L. Air quality index prediction using error back propagation algorithm and improved particle swarm optimization. *International Conference on Mechatronics and Intelligent Robotics*. Springer, Cham. 2017. 9–14.
- [8] Cabaneros, S.M.S., Calautit, J.K.S. and Hughes, B.R., Hybrid Artificial Neural Network Models for Effective Prediction and Mitigation of Urban Roadside NO₂ Pollution. *Energy Procedia*. 2017. 142: 3524–3530. 9th International Conference on Applied Energy, ICAE2017, 21-24 August 2017, Cardiff, UK
- [9] Challoner, A., Pilla, F. and Gill, L. Prediction of indoor air exposure from outdoor air quality using an artificial neural network model for inner city commercial buildings. *International journal of environmental research and public health*. 2015. 12(12): 15233–15253.
- [10] Chaloulakou, A., Grivas, G. and Spyrellis, N. Neural network and multiple regression models for PM₁₀ prediction in Athens: a comparative assessment. *Journal of the Air & Waste Management Association*. 2003. 53(10): 1183–1190.
- [11] Gao, M., Yin, L. and Ning, J. Artificial neural network model for ozone concentration estimation and Monte Carlo analysis. *Atmospheric Environment*. 2018. 184: 129–139.
- [12] Azid, A., Juahir, H., Latif, M.T., Mohd Zain, S. and Osman, M.R. Feed-forward artificial neural network model for air pollutant index prediction in the southern region of Peninsular Malaysia. *Journal of Environmental Protection*. 2013. 4(12): 1–10.
- [13] Alkasassbeh, M., Sheta, A.F., Faris, H. and Turabieh, H. Prediction of PM₁₀ and TSP Air Pollution Parameters Using Artificial Neural Network Autoregressive, External Input Models: A Case Study in Salt, Jordan. *Middle-East Journal of Scientific Research*. 2013. 14(7): 999–1009.
- [14] Kurt, A., Gulbagci, B., Karaca, F. and Alagha, O. An online air pollution forecasting system using neural networks. *Environment International*. 2008. 34(5): 592–598.

- [15] Lal Benjamin, N., Sharma, S., Pendharker, U. and Shrivastava, J.K. Air quality prediction using artificial neural network. *IJCS*. 2014. 2(4): 07–09.
- [16] Kang, G.K., Gao, J.Z., Chiao, S., Lu, S., and Xie, G. Air quality prediction: big data and machine learning approaches. *International Journal of Environmental Science and Development*. 2018. 9(1): 8–16.
- [17] Zhang, J. and Ding, W. Prediction of air pollutants concentration based on an extreme learning machine: the case of hong kong. *International Journal of Environmental Research and Public Health*. 2017. 14(114): 1–19.
- [18] Zhu, D., Cai, C., Yang, T. and Zhou, X. A Machine learning approach for air quality prediction: model regularization and optimization. *Big Data and Cognitif Computing*. 2018. 2(5): 1–15.
- [19] Prasad, D. and Sanyal, S. Air Quality and its Effect on Health: A Geographical Perspective of Lucknow City, *Space and Culture*. 2016. 4(1): 51–64.
- [20] Ghose, M. K., Paul, R. and Banerjee, R. K. Assessment of the status of urban air pollution and its impact on human health in the city of kolkata. *Environmental Monitoring and Assessment*. 2005. 108: 151–167.
- [21] Esposito, E., De Vito, S., Salvato, M., Bright, V., Jones, R.L. and Popoola, O. Dynamic neural network architectures for on field stochastic calibration of indicative low-cost air quality sensing systems. *Sensors and Actuators B: Chemical*. 2016. 231: 701–713.
- [22] Zajmi, L., Ahmed, F. Y. H., and Jaharadak, A. A. Concepts, Methods, and Performances of Particle Swarm Optimization, Backpropagation, and Neural Networks. *Applied Computational Intelligence and Soft Computing*. 2018. Article ID 9547212: 1–7.
- [23] Rosli, N. S., Ibrahim, R. and Ismail, I. Intelligent prediction system for gas metering system using particle. *Procedia Computer Science*. 2017. 105: 165–169.
- [24] Kennedy, J. and Eberhart, R. Particle Swarm Optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, Piscataway, NJ, USA: Citeseer. 1995. 1942–1948.
- [25] Kennedy, J. Particle swarm optimization. *Encyclopedia of machine learning*. 2011. 760–766. Springer US.
- [26] Chaturvedi, T.C., Pandit, M. and Srivastava, L. Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch. *International Journal of Electrical Power & Energy Systems*. 2009. 31(6): 249–257.
- [27] Qi, C., Fourie, A. and Chen, Q. Neural network and particle swarm optimization for predicting the unconfined compressive strength of cemented paste backfill. *Construction and Building Materials*. 2018. 159: 473–478.