THAILAND PORT THROUGHPUT PREDICTION VIA PARTICLE SWARM OPTIMIZATION BASED NEURAL NETWORK

Siwaporn Kunnapapdeelert  
Burapha University, International College, Logistics management, Chonburi, Thailand

Sirinthorn Thepmongkorn  
Burapha University, International College, International business management, Chonburi, Thailand

**Key words:** port throughput, particle swarm optimization, neural network, logistics and prediction

**doi:** 10.5937/jaes18-25687

**Cite article:**
Kunnapapdeelert, S., & Thepmongkorn, S. [2020]. Thailand port throughput prediction via particle swarm optimization based neural network. *Journal of Applied Engineering Science*, 18(3) 338 - 345.

**Online access of full paper is available at:** www.engineeringscience.rs/browse-issues
THAILAND PORT THROUGHPUT PREDICTION VIA PARTICLE SWARM OPTIMIZATION BASED NEURAL NETWORK

Siwaporn Kunnapapdeelert1*, Sirinthorn Thepmongkorn2
1Burapha University, International College, Logistics management, Chonburi, Thailand
2Burapha University, International College, International business management, Chonburi, Thailand

Shipping volume in Thailand have significantly increased in last four years. It is important to pay attention to the trend of Thailand port throughput and use as the guideline to prepare for the needs of supporting facilities, infrastructures, financial and human resources. An effective forecasting technique called particle swarm based neural network (PSONN) is developed to estimate Thailand port throughput in this work. The prediction results from PSONN and classical backpropagation training algorithm, backpropagation neural networks (BPNN) were compared. The results shown that PSONN provides more accurate results than BPNN when apply to predict port throughput of Thailand. The mean squared error obtained from PSONN are about 10 times lower than that of BPNN. This confirms that neural network based on PSO training algorithm has better performance and better ability to escape local optimum than that of BPNN.

Key words: port throughput, particle swarm optimization, neural network, logistics and prediction

INTRODUCTION

Thailand is in a center of mainland Southeast Asia and also located in between Gulf of Thailand and Andaman Sea which offer logistics opportunities to Thailand. The port throughput is one of the most important factors to indicate the economics of the country. The Port Authority of Thailand (PAT) reported that the shipping volume have been dramatically increased over past four years coupled with the strategies for developing new transportation in Thailand during 2015-2022 for water transportation is to improve seaport for both Gulf of Thailand and Andaman Sea sides. It is very interesting to study the Thailand port throughput and use as the guideline to plan for improving the seaport supply chain. Therefore, trend of port throughput is needed to be taken into account to use as the guideline to enhance port performance in the future.

Artificial intelligent and statistical techniques are generally used for solving a wide range of the real-world problems. Several statistical techniques such as moving average (MA), least squared methods (LSQ), regression, or multiple regressions, and exponential smoothing are widely used for the forecasting purpose however, the use of such techniques are not accurate in a satisfactory manner. Artificial neural network is a tool that can model complex non-linear relationships among relevant factors. It models the base on learning process from the historical data trends. It can produce high accuracy of the solutions through training, testing and validation processes. Consequently, artificial neural network (ANN) is recognized as the effective forecasting approach in real world situations. Learning performance of neural network (NN) can be improved by the integration of optimization algorithms such as simulated annealing (SA), tabu search (TS), genetic algorithm (GA), Genetic Programming (GP), Differential evolution (DE), and particle swarm optimization (PSO). Numerous studies show that the integration of optimization search technique with the neural network provide the better results than conventional neural network.

The contributions of this study are summarized as follows:

- Particle swarm optimization (PSO) is employed to train the weights of NN for higher convergence rate and higher efficiency.
- PSO-based neural network (PSONN) is developed for predicting port throughput in Thailand.
- The prediction results can be used as the benchmark and guideline for operational seaport management.

THEORY AND EXPERIMENTAL

Port capacity is theoretically measured as the maximum throughput in tons, twenty-foot equivalent unit (TEU), or others unit. The port capacity depends on eight key el-
elements i.e. 1. channels and waterways, 2. terminals, 3. berths and berth length, 4. loading and unloading equipment, 5. storage space for cargo, containers, and chassis, 6. Modal connections, 7. port operating factors, and 8. external factors such as weather, schedule reliability and institutional disruption [4].

Port throughput and capacity can be used as a parameter for estimating the performance of the port. This would be beneficial for the port management to develop competitive strategies. Another factor that must be considered for enhancing the port performance is the demand in the future. If the future demand can be accurately forecast, it would be beneficial for the port manager to plan and prepare everything for serving the customers’ need. Forecasting of the demand is not easy since there is a numerous affected factor such as economy growth, existing situation, competitors, exchange rate, etc. Consequently, forecasting approach is needed to be developed for improving the accuracy of the existing times series forecasting methods. In this case, gross provincial product per capita (GPP), sea import statistics, sea export statistics, and berth capacity are considered as the influencing factors of the container throughput.

Artificial neural network (ANN)

Recently, Artificial Neural network have been applied in a wide range of researchers due to its ability to learn a non-linear function from the example which is inspired by human brain. ANN can be used for extracting the patterns and detecting the trends the data that are complicated. Feedforward network (FFNN) or multilayer perceptron (MLP) is one of the most popular type of neural network due to its simplicity, ease of calculation and good capabilities. It has been applied in a wide range of areas i.e. marketing, electronics, economics, etc. to approximate the function. The structure of feedforward composes of three main layers which are input layer, hidden layer and output layer. Input data is delivered to the input layer, the data is then passed through the network in each layer until it reaches to output layer without feedback between layers as described in Figure 1.

Backpropagation neural networks (BPNN)

Backpropagation neural networks (BPNN) is the supervised learning technique that learn to improve itself based on error correction. BPNN is suitable for solving the problems that have a lot of input and output data but cannot find the relationship between input and output, have high complexity, the solution to problem keep changing overtime and output can be fuzzy. Consequently, BPNN has been successfully applied to various areas such as chemical [5], transportation [6], ergonomics [7], banking [8], marketing [9], economics [10-11], medical [12-13], and energy [14-15] and others. Later, BPNN was an algorithm that used for training feedforward network inspired [16]. BPNN enhances performance of FFNN by adjusting weights and bias of neurons based on the error which is the difference between target output and actual output. All weights are randomly generated and used in the first iteration and they are then adjusted and updated after iteration during training process. Steps for training the neural network can be described as follows. The first step is feedforward operation which composes of two step processes. Input values are fed into the input nodes. They are then pushed into the network through the nodes in the hidden layer. After that, they all values in the hidden nodes are multiplied with the weights of the connecting nodes. The total net input is then calculated by Equation (1). The total net input, \( y_{net} \), is then transferred by the activation function (sigmoid function is normally used as an activation function) as presented in Equation (2). After that, the error of the network is computed by Equation (3). The obtained error is used for updating the weight in the network to minimize the error function based on gradient descent method. If the obtained results from the updated weights are better than those of the previous set of weights, the new set of weights will be replaced the previous set and iteration goes on. The concept of BPNN is showed in Figure 2.
Figure 2: Backpropagation network architecture

where

\[ y_{\text{net}} = \sum_{i=1}^{n} x_i \cdot w_i + w_0 \]

\[ y_{\text{out}} = f(y_{\text{net}}) = \frac{1}{1 + e^{-y_{\text{net}}}} \]

\[ \text{TotalError} = \frac{1}{2} \sum_{i=1}^{k} (y_{\text{target}} - y_{\text{out}})^2 \]

where

\( x_i \) refers to input neuron,

\( w_i \) is weight between input and hidden layer,

\( w_0 \) represents bias,

\( y_{\text{out}} \) describes total of weighted inputs,

\( y_{\text{out}} \) shows the response of the system,

\( f(y_{\text{net}}) \) is nonlinear activation function,

\( y_{\text{target}} \) refers to target output.

Although BPNN is a very effective tool for searching the solution that is not clear in numerous areas, it takes quite long computational time for training the weights in the network and easily trapped into local solution. Subsequently, various researchers have integrated metaheuristics approaches such as genetic algorithm (GA) [17], ant colony optimization (ACO) [18], particle swarm optimization (PSO) [19], cuckoo search (CS) [20], differential evolution (DE) [21] with neural network for training the weight for managing such drawbacks.

**Particle swarm optimization (PSO)**

Particle Swarm Optimization (PSO) is firstly introduced in 1995 [21]. It inspired by the movement of a bird flock and fish school. The algorithm composes of velocity, weight, and particles (potential solutions). Each particle in PSO randomly flies in D-dimensional search space with its own velocity. Position and velocity of each particle are adjusted according to its own flying experiences and the best's experience among the group at each time step. Direction and velocity of each particle are changed towards its own best experience (pbest) and the group best experiences (gbest) as explained in Equation (4).

\[ v_{id} = w v_{id} + c_1 \cdot \text{rand}_1() \cdot (p_{id} - x_{id}) + c_2 \cdot \text{rand}_2() \cdot (g_{id} - x_{id}) \]  

(4)

\[ x_{id} = x_{id} + v_{id} \]  

(5)

when \( x_i=x_{i1},x_{i2},...,x_{iD} \) represents the \( i \)th particle, the previous best position (position of the best fitness value) of the \( i \)th particle are \( P_i=P_{i1},P_{i2},...,P_{iD} \) while symbol \( g \) represent the best particle in the population. \( V_i=v_{i1},v_{i2},...,v_{iD} \) is the velocity of each particle, \( w \) represents inertia weight that is the coefficient of previous velocity used for controlling the influence of particle's previous velocity on the particle's current velocity which affect to exploitation and exploration of the particle. If inertia weight is set to be large value, the algorithm would explore the new search space which leads to the delay of convergence and vice versa. Therefore, the adjusted inertia weight is applied in this experiment for a better trading of exploitation and exploration. \( c_1 \) and \( c_2 \) are two positive constants that called as learning factors. \( \text{rand}_1() \) and \( \text{rand}_2() \) are random numbers in the range [0,1].

PSO has been widely applied for solving optimization problems due to its simplicity, high computational speed, robustness, effectiveness. PSO has been developed for a wide range of applications such as logistics [22-23] medical [24], scheduling [25-26], etc. These imply that PSO is suitable for enhancing the performance of neural network.

**Particle swarm optimization for training backpropagation neural network (PSONN)**

Researchers have been enhancing the performance of BPNN by using optimization search techniques to train the weight. It is because BPNN can be easily trapped into the local optimum or unsatisfied solution during the
development process. Numerous studies have been proposed to identify ability of PSO as an effective training algorithm for NN. The studies showed that PSO has a very high capability for training BPNN [27-32]. Although some attempts have been made to use other optimization search techniques for training the weights, it has been found that the results obtained by PSO-BPNN provides higher accuracy when compared to other algorithms [33]. The procedure for PSO-BPNN is classified into seven steps as follows.

Step 1 Initialization of neural network’s parameters: Number of nodes in input layer \((m)\), hidden layer \((n)\) and output layer \((o)\) are designed.

Step 2 Setting of PSO parameters: Set all relevant parameters of PSO algorithm i.e. population size \((N)\), maximum number of population \((MaxIter)\), inertia weight \((\omega)\), position \((x)\), velocity \((v)\), learning factors \(c_1\) and \(c_2\).

Step 3 Determine fitness of particles: The fitness of particles is calculated from the individual best \((pbest)\) and group best \((gbest)\) experiences. Solution quality from pbest and gbest are compared and the best position is recorded. The mean square error \((MSE)\) is used for comparing the performance of each particle as presented in Equation (6).

\[
Fitness = \frac{1}{N} \sum_{j=1}^{N} \sum_{i=1}^{O} (target_{ji} - actual_{ji})^2
\]

for the \(i^{th}\) sample, and \(j^{th}\) output node.

Step 4 Comparison of particles’ fitness: The individual best and group best positions are determined as follows:

- \(if x_{id} > p_{id}\) then \(p_{id} = x_{id}\) otherwise \(p_{id} = p_{id}'\)
- \(if x_{id} > g_{id}\) then \(g_{id} = x_{id}\) otherwise \(g_{id} = g_{id}'\)

Step 5 Updating of position and velocity of each particle. The particles are updated based on the equations (3) and (4), respectively.

Step 6 Error calculation: The error of the PSO algorithm can be calculated from Equation (7)

\[
Error = \frac{1}{iter} \sum_{i=1}^{iter} fitness(g_{id})
\]

where \(iter\) is number of iteration and fitness \((g_{id})\) represents the best fitness value of the \(i^{th}\) iteration.

Step 7 Termination process: The process is terminated when stopping criterion is met, otherwise the process returns to Step 3. Stopping criterion can be either the maximum number of iterations reach, or the error is low enough.

RESULTS

PSO-based neural network (PSONN) is used for estimating throughputs of Thailand ports. The data have been collected from Port Authority of Thailand (PAT) from 2009 to 2018. In this research, all port throughputs in Thailand (divided by customs house) are used as the sample for predicting the port throughput by BPNN and PSONN. It was implemented in MATLAB R2016b. The BP training algorithm was implemented by the MATLAB Neural Network Toolbox whereas PSONN was implemented by using MATLAB code.

Dataset were divided into two groups for training and testing. 50% of the data set were used selected for training the network whereas 50% of data set were used for validating performance of the trained weights. There is no exact parameter setting values of both PSO and neural network that can work well for all types of the problems. Subsequently, the parameter setting of PSO and NN in this case are selected from the experimental results base on trial and error. Parameter selection in back BPNN depends on a few factors. There are two factors learning rate and momentum value, use for controlling the weights adjustment along the descent direction. Even though both factors are used to adjust the weights, they are differed from each other in that the learning rate is used to adjust step size of the weight whereas momentum factor is used to accelerate convergence of the network. In this work, \(\eta\) (learning rate) and \(\lambda\) (momentum factor) were selected as 0.5 and 1, respectively. In case of computation of PSO, it depends on few parameters: population size, inertia weight, maximum velocity, maximum and minimum positions and maximum number of iterations. Computation of PSONN, population size and maximum iteration of 50 and 1000 were selected, respectively. The inertia weight gradually decreased from 0.9 to 0.4 so as to balance the global and local exploration. Since particles’ velocities on each dimension are clamped to a maximum velocity, \(v_{max}\), to control the exploration ability of particles. If \(v_{max}\) is too high, the PSO facilitates global search, and particles might fly pass good solutions. However, if \(v_{max}\) is too small, the PSO facilitates local search, and particles may not explore beyond locally good regions. Thus, if \(v_{a}\) is greater than \(v_{max}\), then \(v_{a}\) is equated to \(v_{max}\). Similarly, if \(v_{a}\) is less than \(-v_{max}\), then \(v_{a}\) is equated to \(-v_{max}\). Therefore, maximum velocity, it was set at 12% of the dynamic range of the variable in each dimension. In case of, maximum and minimum position of the variable in each dimension, there were set to 0.5 and -0.5, respectively.

All variables for training the network of both algorithms were kept the same. The tan-sigmoid function was applied on the first layer for increasing the computational power of the network, whereas linear function or purelin function was applied on the last layer in order map their inputs onto a real number range that matches the expected output for both training algorithms. In addition, number of hidden node and maximum iteration were also set to 5 and 50,000 (number of epochs in BPNN = maximum iteration size of swarm in PSONN), respectively.

The tested set is used to test how well the trained neural networks generalize. Table 1 shows percent of accuracy of the predicted values over 50,000 iterations after trained and tested the neural network by using BP and PSO.
**Table 1: Results of comparison between two training methods**

| Customs   | BPNN | PSONN |
|-----------|------|-------|
|           | Train target | Test target | Train target | Test target |
| Bangkok   | 26940.03477  | 37097.39882  | 917047.2681  | 1262807.139 |
| Khlong Yai | 432394.3029  | -90637.61414 | 1365185.096  | 1723408.96  |
| Map Ta Phut | 32050096.32 | 43676808.49  | 51118547.14  | 64304241    |
| Laem Chabang | 98130349.57 | 135129028.1  | 88455738.54  | 121806740   |
| Chachengsao | 401387.6831 | 670832.9544  | 1024659.7    | 1799861.641 |
| Bangpakong  | 26940.03477  | -26838559.91 | 917047.2681  | 10463265.61 |
| Samutsakorn | 2198152.32  | 6748305.239  | 3562473.268  | 4015675.018 |
| Samutsongkhram | 659747.5331 | 1179015.906  | 1574258.545  | 2203868.098 |
| Ban Leam    | 26940.03477  | 37097.39882  | 917047.2681  | 1262807.139 |
| Koh Lak     | 4269190.943  | 7901599.668  | 3193638.884  | 4535284.644 |
| Chumphon    | 104117.9797  | 281442.7216  | 1045316.874  | 1471971.812 |
| Bandorn     | 4916600.038  | 19406848.46  | 5306003.715  | 8597481.116 |
| Koh Samui   | 27202.46319  | 37097.39882  | 917166.5384  | 1262807.139 |
| Nakornsritharamarat | 728635.8865 | 728074.1573  | 1433890.986  | 1866345.074 |
| Sichon      | 263503.731   | 3740173.859  | 2586894.816  | 3208694.261 |
| Songkhla    | 7658447.673  | -67630865.41 | 7762217.904  | 10132940.69 |
| Pattani     | 26940.03477  | 36498.79874  | 917047.2681  | 1262905.797 |
| Tak Bai     | 23178.89335  | 482490.4436  | 962364.7331  | 1397184.708 |
| KraBuri     | 26940.03477  | 37097.39882  | 917047.2681  | 1262807.139 |
| Ranong      | 217703.0012  | 374116.1762  | 1169116.688  | 1523815.785 |
| Phang Nga   | 26940.03477  | 37097.39882  | 917047.2681  | 1262807.139 |
| Phuket      | 330931.8321  | 627672.6756  | 1372353.829  | 1615026.399 |
| Krabi       | 3445561.808  | 380052.9979  | 3690526.007  | 5385912.798 |
| Kantang     | 2325638.386  | 5336438.191  | 2249548.263  | 3247687.551 |
| Satun       | 56695.91447  | 70909.9057   | 965394.8665  | 1325068.526 |
| MSE         | 1.82E+08     | 3.34E+14     | 1.8929E+13   | 2.9964E+13  |

**DISCUSSION**

Training and testing results from both training algorithms for port throughput estimation (25 customs house in Thailand) are plotted versus the corresponding output as shown in Figures 3 to 6. Mean squared error (MSE) is used to measure the performance of BPNN and PSONN. The experimental results show that the predicted values obtained from PSO-NN are not far from the desired values for both trained and tested set when used to predict port throughput. However, the results predicted from BP training algorithm gives lower accuracy than those of PSO. Table 1 shows that average values of MSE obtained from PSO-based neural network after applied for port throughput prediction are about 10 times lower than that of BPNN. Considering the average percent difference of the results from BPNN and PSONN is 167 percent. Classical BP is a network training function that updates weight and bias values according to gradient descent concept so the solution from this training may get stuck in the local minima easily. While PSO algorithm operates on information about the relative performance of the individuals on the population. Some particles will explore new space while the populations still remember the global best solution which seems to solve the problem of gradient descent. Additionally, structure of PSO consists of inertia weight and stochastic factors, which allow the particles to search the new space and possible to avoid the local optima. Consequently, the results from BPNN give high-
CONCLUSIONS

Conventional computation cannot solve the problem that cannot be directly explained by mathematical model to represent the correlation between input and output. It is because such problem has noisy or incomplete data which always happen in the real world. Industrial area also has to face with this kind of problems especially, demand forecasting. When compared to other traditional computing algorithm, neural network can determine implicit relationship between inputs and outputs by learning from the given data. It can be also applied to the problem that has dynamic or nonlinear relationship, does not limited in strict assumption such as normality, linearity, variable independence, etc. Hence, neural network was applied to estimate port throughputs since port throughputs play an important role in Thai logistics development. As BPNN is subject to local optimum and slow convergence rate therefore PSO was applied for training neural network to overcome such drawbacks. The PSO is very attractive because it presents advantages of simplicity since it requires only primitive mathematical operations, short computer code, few parameters to adjust, and fast convergence. Performance of artificial neural network bases on BP and PSO-based were compared as illustrated in Table 1. The results show that PSONN is superior to BPNN in terms of quality of solution when applied to predict port throughputs. The mean squared error obtained from PSO-based neural network after applied for
port throughput prediction are about 10 times slower than that of BPNN. This confirms that neural network based on PSO training algorithm has better performance and better ability to escape local optimum than that of classical BP training algorithm.

More influencing factors such as key element of port capacity i.e. channels, terminals, loading and unloading equipment, etc. would be taken into consideration to make the model more comprehensive and realistic and improve performance of this computational technique. Subsequently, the more precise results could be as a guideline for the better seaport supply chain management. In addition, systematic parameters selection of the PSO will certainly enhance its ease of use and should be investigated in the future. Enhance its ease of use and should be investigated in the future.

REFERENCES

1. Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., & Cosar, A. (2019). A survey on new generation metaheuristic algorithms. Computers & Industrial Engineering, 137, 106040.
2. Hassan, R., Cohanim, B., De Weck, O., & Venter, G. (2005, April). A comparison of particle swarm optimization and the genetic algorithm. In 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference (p. 1897).
3. Hamed, H. N. A., & Haza, N. (2006). Particle swarm optimization for neural network learning enhancement (Doctoral dissertation, Universiti Teknologi Malaysia).
4. UNCTAD. PORT MANAGEMENT SERIES 2016. Port Performance Linking Performance Indicators to Strategic Objectives, 4, pp. 11-12.
5. Sorrosal, G., Irigoyen, E., Borges, C. E., Martin, C., Macarulla, A. M., & Alonso-Vicario, A. (2017). Artificial neural network modelling of the bioethanol-to-olefins process on a HZSM-5 catalyst treated with alkali. Applied Soft Computing, 58, 648-656.
6. Kwon, H. B. (2017). Exploring the predictive potential of artificial neural networks in conjunction with DEA in railroad performance modeling. International Journal of Production Economics, 183, 159-170.
7. Witoomchart, P., & Chongstitvatana, P. (2017). Application of structured support vector machine back-propagation to a convolutional neural network for human pose estimation. Neural Networks, 92, 39-46.
8. Kwon, H. B., & Lee, J. (2015). Two-stage production modeling of large US banks: A DEA-neural network approach. Expert Systems with Applications, 42(19), 6758-6766.
9. Gordini, N., & Veglio, V. (2017). Customers churn prediction and marketing retention strategies. An application of support vector machines based on the AUC parameter-selection technique in B2B e-commerce industry. Industrial Marketing Management, 62, 100-107.
10. Chen, F. H., Chi, D. J., & Wang, Y. C. (2015). Detecting biotechnology industry's earnings management using Bayesian network, principal component analysis, back propagation neural network, and decision tree. Economic Modelling, 46, 1-10.
11. Kordanuli, B., Barjaktarovic, L., Jeremic, L., & Alizamir, M. (2017). Appraisal of artificial neural network for forecasting of economic parameters. Physica A: Statistical Mechanics and its Applications, 465, 515-519.
12. Samanta, B., & Nataraj, C. (2008). Automated diagnosis of cardiac state in healthcare systems using computational intelligence. International Journal of Services Operations and Informatics, 3(2), 162-177.
13. Shaikhina, T., & Khovanova, N. A. (2017). Handling limited datasets with neural networks in medical applications: A small-data approach. Artificial intelligence in medicine, 75, 51-63.
14. Phy, P. P., & Jeenanunta, C. (2019). Electricity load forecasting using a deep neural network. Engineering and Applied Science Research, 46(1), 10-17.
15. Ma, L., Hu, S., Qiu, M., Li, Q., & Ji, Z. (2017). Energy consumption optimization of high sulfur natural gas purification plant based on back propagation neural network and genetic algorithms. Energy Procedia, 105, 5166-5171.
16. Rumelhart, D. E. (1986). Parallel distributed processing: Explorations in the microstructure of cognition. Learning internal representations by error propagation, 1, 318-362.
17. Schaffer, J. D., Whitley, D., & Eshelman, L. J. (1992, June). Combinations of genetic algorithms and neural networks: A survey of the state of the art. In [Proceedings] COGANN-92: International Workshop on Combinations of Genetic Algorithms and Neural Networks (pp. 1-37). IEEE.
18. Socha, K., & Blum, C. (2007). An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training. Neural Computing and Applications, 16(3), 235-247.
19. Mazurowski, M. A., Habas, P. A., Zurada, J. M., Lo, J. Y., Baker, J. A., & Tourassi, G. D. (2008). Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. Neural networks, 21(2-3), 427-436.
20. Valian, E., Mohanna, S., & Tavakoli, S. (2011). Improved cuckoo search algorithm for feedforward neural network training. International Journal of Artificial Intelligence & Applications, 2(3), 36-43.

21. Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN’95-International Conference on Neural Networks (Vol. 4, pp. 1942-1948). IEEE.

22. Cheng, Z., & Juncheng, T. (2015). Adaptive combination forecasting model for China’s logistics freight volume based on an improved PSO-BP neural network. Kybernetes.

23. Liu, H., Zhang, Q., & Wang, W. (2011). Research on location-routing problem of reverse logistics with grey recycling demands based on PSO. Grey Systems: Theory and Application.

24. Salmeron, J. L., Rahimi, S. A., Navali, A. M., & Sadeghpour, A. (2017). Medical diagnosis of Rheumatoid Arthritis using data driven PSO–FCM with scarce datasets. Neurocomputing, 232, 104-112.

25. Kim, H. H., Choi, J. Y., & Park, S. C. (2017). Tire mixing process scheduling using particle swarm optimization. Computers & Industrial Engineering, 110, 333-343.

26. Sangsawang, C., & Sethanan, K. (2016). Hybrid particle swarm optimization with a Cauchy distribution for solving a reentrant flexible flow shop problem with a blocking constraint. Engineering and Applied Science Research, 43(2), 55-61.

27. Gudise, V. G., & Venayagamoorthy, G. K. (2003, April). Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS’03 (Cat. No. 03EX706) (pp. 110-117). IEEE.

28. Che, Z. H. (2010). PSO-based back-propagation artificial neural network for product and mold cost estimation of plastic injection molding. Computers & Industrial Engineering, 58(4), 625-637.

29. Wang, H. S., Wang, Y. N., & Wang, Y. C. (2013). Cost estimation of plastic injection molding parts through integration of PSO and BP neural network. Expert Systems with Applications, 40(2), 418-428.

30. Pwasong, A., & Sathasivam, S. (2016). A new hybrid quadratic regression and cascade forward backpropagation neural network. Neurocomputing, 182, 197-209.

31. Gordan, B., Armaghani, D. J., Hajihassani, M., & Monjezi, M. (2016). Prediction of seismic slope stability through combination of particle swarm optimization and neural network. Engineering with Computers, 32(1), 85-97.

32. Ibrahim, A. M., & El-Amary, N. H. (2018). Particle Swarm Optimization trained recurrent neural network for voltage instability prediction. Journal of Electrical Systems and Information Technology, 5(2), 216-228.

33. Engelbrecht, A. P. (2007). Computational intelligence: an introduction. John Wiley & Sons.