

## Review Article

# Concepts, Methods, and Performances of Particle Swarm Optimization, Backpropagation, and Neural Networks

**Leke Zajmi** , **Falah Y. H. Ahmed** , and **Adam Amril Jaharadak**

*Department of Information Sciences and Computing, Faculty of Information Sciences and Engineering (FISE), Management and Science University, 40100 Shah Alam, Malaysia*

Correspondence should be addressed to Falah Y. H. Ahmed; [falaha899@gmail.com](mailto:falaha899@gmail.com)

Received 16 April 2018; Revised 7 August 2018; Accepted 15 August 2018; Published 3 September 2018

Academic Editor: Christian W. Dawson

Copyright © 2018 Leke Zajmi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the advancement of Machine Learning, since its beginning and over the last years, a special attention has been given to the Artificial Neural Network. As an inspiration from natural selection of animal groups and human's neural system, the Artificial Neural Network also known as Neural Networks has become the new computational power which is used for solving real world problems. Neural Networks alone as a concept involve various methods for achieving their success; thus, this review paper describes an overview of such methods called Particle Swarm Optimization, Backpropagation, and Neural Network itself, respectively. A brief explanation of the concepts, history, performances, advantages, and disadvantages is given, followed by the latest researches done on these methods. A description of solutions and applications on various industrial sectors such as Medicine or Information Technology has been provided. The last part briefly discusses the directions, current, and future challenges of Neural Networks towards achieving the highest success rate in solving real world problems.

## 1. Introduction

Artificial Neural Network (ANN) or simply known as Neural Networks (NNs) is the area which has received and continues to receive attention from world's greatest researchers. In scientific terms it is known as a structure of interconnected units of large number of neurons. As the researcher Zhang et. al [1] mentioned in his research, each of those interconnected neurons have the ability of receiving, processing, and sending an output signals. In more common understanding, researcher Sonali et. al [2] stated that Neural Networks are a digital copy of human biological nervous system and follow the same path of learning neurons. It processes information in a similar way to how the human brain does.

A Neural Network consists of three sets, with the first set being the pattern of connections which is between neurons (an adder that sums the input data), the second set being the method of determining the weights on the connections, and lastly an activation function which limits the output amplitude of the neuron. Neural Networks (NNs) are daily used in various applications. In 2017, Rasit A. [3] found that Neural Networks are highly useful when it comes to pattern recognition, optimization, simulation, and also prediction.

A trained Artificial Neural Network (ANN) could be considered an “expert” in the task of information that is been given to analyze, and this comes as an advantage Neural Networks have in taking different approaches when it comes to problem solving.

In the following Section 2, a review is provided on the methods and algorithms of the Artificial Neural Network, such as Backpropagation and Swarm Intelligence which includes Practical Swarm Optimization. A review on Feed-forward and Backward phases of Backpropagation algorithm together with their sets of equations explained will give a brief understanding on how this method and algorithm works, together with its weaknesses and strengths. Furthermore, the discussion continues on the Multilayer Perceptron (MLP) and the supervised and unsupervised learning techniques are also briefly explained.

## 2. Artificial Neural Networks (ANN)

One of the most researched techniques of Neural Networks is Backpropagation. Backpropagation is a technique in which its network of nodes is arranged in layers. Researcher Jaswante

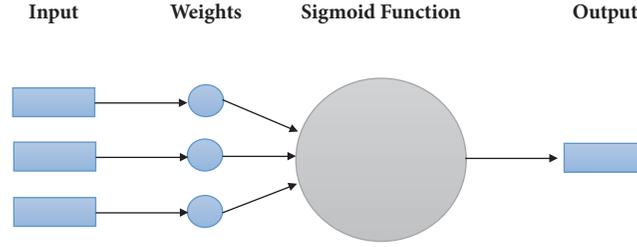


FIGURE 1: Artificial Neural Network model.

et. al [6] describes it as the first layer of the network being the input layer, and the last layer being the output layer, while all the remaining intermediate layers being called hidden layers. Backpropagation is a technique which considers a number of elements in order to get an impact on its convergence. Input, processing (hidden), and output nodes are part of those elements, together with the momentum rate and minimum error [7].

Learning in Backpropagation follows a set of steps. These steps are simplified as follows:

- (a) The input layer gets presented to input vector
- (b) The output layer gets presented to the set of a desired output
- (c) A comparison between the desired errors and the actual output is done after every forward pass
- (d) The results' comparison determines weight changes according to learning rules

Despite the hype of the widely researched Backpropagation, this algorithm is also well known for its disadvantages and its accuracy leaves room for much better desired results. A group of researchers, lead by Cho et. al. [4], states that Backpropagation takes longer in time when it comes to training. This disadvantage comes mostly due to the timing during backward moves that neuron perform until the ideal solution is found. Thus, a few researchers started using Swarm Intelligence algorithms (SI) which enhances the learning in Neural Networks using different approach.

Researchers Christian Blum and Daniel Merkle [8] describe Swarm Intelligence as a technique which has taken inspiration from the group or collective behavior of animals and insects, like the collective behavior of insects, flocks of birds, fishes, etc. This inspiration comes due to the technique the neurons (or known as swarms) use by following the group of collective neurons towards the better solution.

Swarm Intelligence uses a common and also one of the most accurate techniques known, Particle Swarm Optimization. The main objective of this method in Neural Networks is getting the best particle position from a group of particles which are either moving or trying to move towards the best solution.

**2.1. Artificial Neural Network (ANN) and Backpropagation Algorithm (BP).** Artificial Neural Network consists of a network which is made of neurons, nodes, or cells arranged and interconnected to that network. Neurons in Artificial Neural Networks have the capability of learning from examples, and they are able to respond intelligently to new triggers [9, 10].

A typical Neural Networks (NNs) topology is shown in Figure 1. Each node consists of an activation function called sigmoid function. The signal sent from each input node travels through the weighted connection, whereby according to internal activation function the output is produced.

Figure 2 shows the Multilayer Perceptron (MLP), which is the interconnection flow among nodes in Artificial Neural Network (ANN).

The equations of the processes between input ( $i$ ) and hidden ( $j$ ) layers are as follows:

$$O_j = f(\text{net}_j) = \frac{1}{1 + e^{-\text{net}_j}} \quad (1)$$

$$\text{net}_j = \sum_j w_{ij} O_i + \theta_j \quad (2)$$

with

$O_j$  being the output of node  $j$

$O_i$  being the output of node  $i$

$w_{ij}$  being the connected weight between nodes  $i$  and  $j$

$\theta_j$  being the bias of node  $j$

The further transitions between hidden layer ( $j$ ) and output layer ( $k$ ) are as follows:

$$O_k = f(\text{net}_k) = \frac{1}{1 + e^{-\text{net}_k}} \quad (3)$$

$$\text{net}_k = \sum_k w_{jk} O_j + \theta_k \quad (4)$$

with

$O_k$  being the output of node  $k$

$O_j$  being the output of node  $j$

$w_{jk}$  being the connected weight between nodes  $j$  and  $k$

$\theta_k$  being the bias of node  $k$

The error of the above process is calculated using (5). This error calculation measures or compares differences between the desired output we desired and the output which was produced. The error gets propagated backward among layers of network, from output to hidden and to input with weights

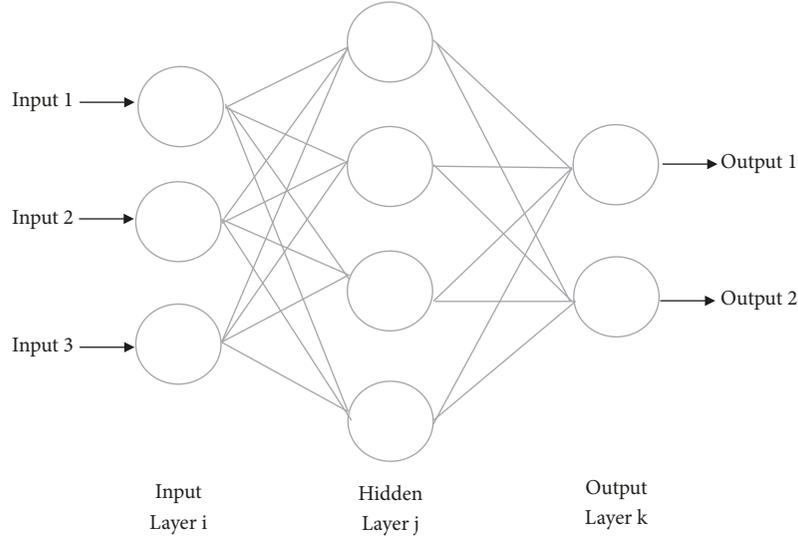


FIGURE 2: Combination of ANN with Multilayer Perceptron (MLP) model [4].

being modified while the weights are modified for error reduction during this propagation.

$$error = \frac{1}{2} (Output_{desired} - Output_{actual})^2 \quad (5)$$

Based on the calculated error above, Backpropagation algorithm gets to be applied on reversing from output (k) to hidden node (j), as shown in

$$w_{kj}(t+1) = w_{kj}(t) + \Delta w_{kj}(t+1) \quad (6)$$

$$\Delta w_{kj}(t+1) = n\delta_k O_j + \alpha \Delta w_{kj}(t) \quad (7)$$

with

$$\delta_k = O_k (1 - O_k) (t_k - O_k) \quad (8)$$

with

$w_{kj}(t)$  being the weight from nodes  $k$  to  $j$  for a time  $t$

$\Delta w_{kj}$  being weight adjustment

$n$  being the learning rate

$\alpha$  being the momentum rate

$\delta_k$  being the error at node  $k$

$O_j$  being the actual network output at node  $j$

$O_k$  being the actual network output at node  $k$

$t_k$  being the target output value at node  $k$

The backward calculations from hidden to input layers (j and i, respectively) are shown in the following examples:

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}(t+1) \quad (9)$$

$$\Delta w_{ji}(t+1) = n\delta_j O_i + \alpha \Delta w_{ji}(t) \quad (10)$$

with

$$\delta_j = O_j (1 - O_j) \sum_k \delta_k w_{kj}$$

$$O_k = f(net_k) = \frac{1}{1 + e^{-net_k}} \quad (11)$$

$$net_k = \sum_k w_{kj} O_j + \theta_k$$

where

$w_{ji}(t)$  is the weight from node  $j$  to node  $i$  at time  $t$

$\Delta w_{ji}$  is the weight adjustment

$n$  is the learning rate

$\alpha$  is the momentum rate

$\delta_j$  is the error at node  $j$

$\delta_k$  is the error at node  $k$

$O_i$  is the network output at node  $i$

$O_j$  is the network output at node  $j$

$O_k$  is the network output at node  $k$

$w_{kj}$  is the weight connected between nodes  $j$  and  $k$

$\theta_k$  is the bias of node  $k$

The repetition of this process is unlimited; however, it stops until convergence is achieved.

**2.2. Swarm Intelligence.** In 2009, researchers Konstantinos and Michael [11] called Swarm Intelligence (SI) an Artificial Intelligence (AI) branch which studies the collective behavior of complex, self-organized, and decentralized systems with social structure. In a simplified understanding, this technique got its inspiration from nature, similar to the way that ant colonies and bird flocks operate, translated into computationally intelligent systems. These systems are

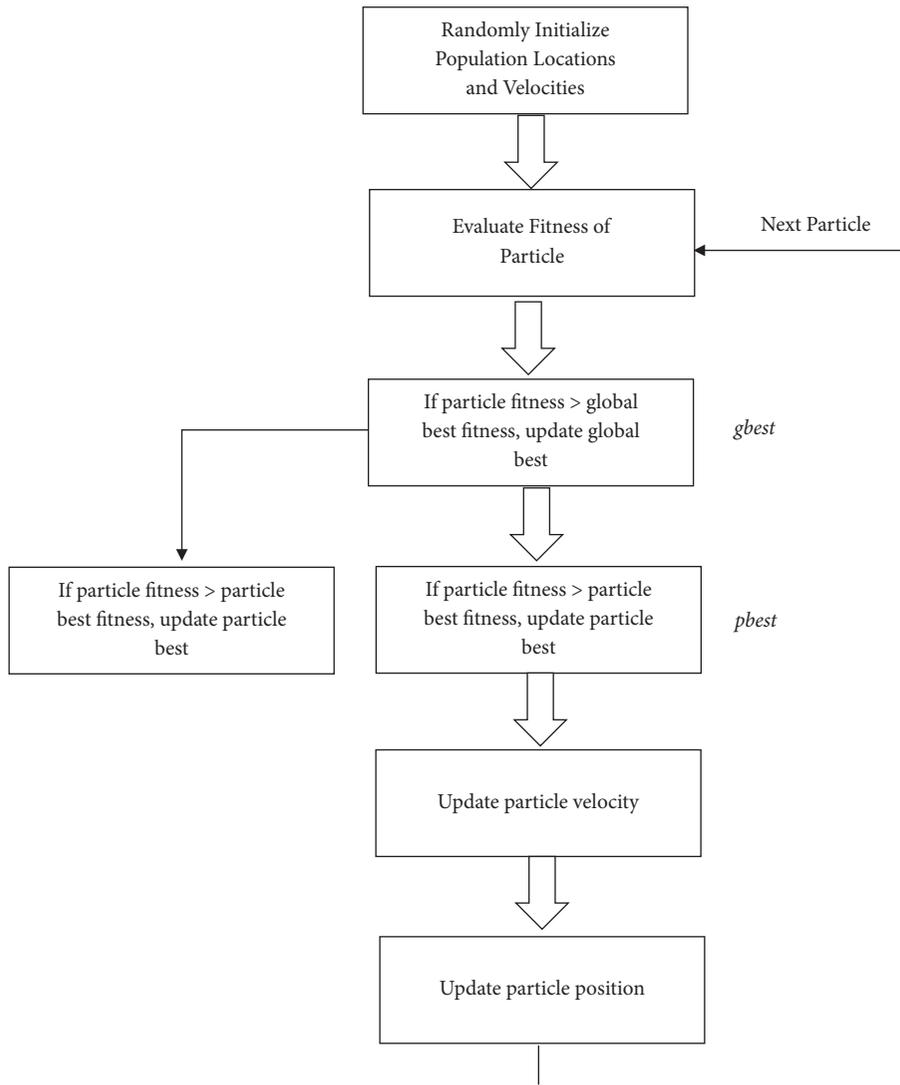


FIGURE 3: Basic PSO procedure [5].

translated in a way that is formed from interacting agents with their environment, whereby such interaction leads to a global solution behavior, similar to bird flocks where each bird is part of the contribution on reaching the destination, or global solution. In Swarm Intelligence, these interacting agents mean that all the neurons or particles of the group work as a team on finding the best place to be. Swarm Intelligence (SI) spreads through specialized optimization techniques, with two of the major techniques known such as Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO), which we will be reviewing in the next section. The Ant Colony Optimization (ACO) comes in handy while solving computational problems through finding good ways/paths using graphs. Its inspiration comes from ants findings ways from colony to their food. On the other side, the second major technique is known, and one of the three reviewed methods of this article is Particle Swarm Optimization (PSO). Due to the more accurate performance, Particle Swarm

Optimization is known to have replaced Genetic Algorithm.

Particle Swarm Optimization is a technique where particles move in group for finding better results. Researcher Gerhard et. al [5] mentioned that when these particles move in group, a vector is used to update the position of those particles, called velocity vector. Figure 3 shows the basic flow procedure of Particle Swarm Optimization.

Particle Swarm Optimization achieves its success rate using different ways of modifications. In 2011, a group of researchers [12] concluded that modification in Particle Swarm Optimization algorithm consists of three categories, the extension of field searching space, adjustment of the parameters, and hybridization with another technique.

**2.3. Particle Swarm Optimization (PSO).** As mentioned previously, this method is one of the core and most interesting parts of Neural Networks.

Particle Swarm Optimization has started from the analysis of real life samples and social models [10, 13]. As Particle Swarm Optimization belongs to the family of Swarm Intelligence, swarms or neurons work together on finding the best solution. Thus, its concept is adapted from natural causes, such as the bird flocking and fish schooling, and this makes Particle Swarm Optimization a population algorithm.

The physical position of the particle is not important, therefore, the swarm (neuron) gets initialized by being assigned to any random position and velocity, as well as the potential solutions which are flown through the hyperspace. Ferguson [14] mentions that similar to our brain neurons which learn from our past experiences, so do the particles in Particle Swarm Optimization.

All particles working towards the global best solution keep record of each position they have taken and achieved to the moment [15]. From these values, the best personal value of particle is called pbest (personal best), and the best value obtained from the overall particle group is called gbest (global best). Iteration of each particle causes acceleration towards their own personal best position (pbest), as well as the overall global best position (gbest). In 2000, the researcher Van den Bergh et. al. [16] stated that these two record (pbest and gbest) velocities were weighted randomly and then produce a new velocity for the particle which will affect the future next positions of the particle.

Particle Swarm Optimization (PSO) includes a set of two equations called the equation of movements (12) and the equation of velocity update (13). The movement of particles by using their specific vector velocity is shown in (12), where the velocity update equation which provides the velocity vector adjustment given the two competing forces (gbest and pbest) is shown in (13).

$$x_n = x_n + v_n * \Delta t \quad (12)$$

$$v_n = v_n + c_1 * rand() * (g_{best,n} - x_n) + c_2 * rand() * (p_{best,n} - x_n) \quad (13)$$

Equation (12) is used for all the elements of x position and v velocity vector. The  $\Delta t$  parameter defines the discrete interval time in which swarm will be moving, and it is usually set to 1.0. This movement results in the new position of the swarm. In (13), the result brings a subtraction of the dimensional element from the best vector—which is then multiplied with a random number of 0 to 1, and also with an acceleration constant of  $C_1$  and  $C_2$ . Hence, the sum gets added to velocity. This process is performed for all the population. If we choose random numbers, those would also provide an amount of randomness helping the swarm towards its path throughout the solution space.  $C_1$  and  $C_2$  acceleration constantly provide control to the equation that defines which one should be given more rights towards the path, which is either global or personal best [4].

In Table 1 we will demonstrate the movement of Particle A towards global best solution in a two-dimensional space (2D). The example gets  $C_1 = 1.0$  and  $C_2 = 0.5$ .  $C_1$  has a higher value than  $C_2$ , which means that it gives higher attention and emphasis to finding the best global solution.

TABLE 1: The value and position of Particle A.

Particle A	Gbest	Pbest
Value = 5	5	15
Value = 10	13	13

We assume that velocity values of Particle A have been calculated in the previous iteration  $P_v = (0.1)$ . First of all, the velocity vector has to be updated for current iteration using (13).

The first position of Article A with the value of 5 is

$$v_x = v_x + c_1 * rand() * (g_{best,x} - Position_x) + c_2 * rand() * (p_{best,x} - Position_x)$$

$$v_x = 0 + 1.0 * 0.35 * (5 - 5) + 0.5 * 0.1 * (15 - 5) \quad (14)$$

$$v_x = 0 + 0.35 + 0.05 * (5)$$

$$v_x = 0.35 + 0.25$$

$$v_x = 0.6$$

Second position of Article A with the value of 10 is

$$v_y = v_y + c_1 * rand() * (g_{best,y} - Position_y) + c_2 * rand() * (p_{best,y} - Position_y)$$

$$v_y = 1 + 1.0 * 0.2 * (13 - 10) + 0.5 * 0.45 * (13 - 10) \quad (15)$$

$$v_y = 1 + 0.2 * (3) + 0.225 * (3)$$

$$v_y = 1 + 0.6 + 0.675$$

$$v_y = 2.275$$

As we can see, the velocity value of Particle A is now  $P_v = (0.6, 2.275)$ ; therefore new velocities will be applied upon particle positions using (2j) as follows:

$$Position_x = Position_x + v_x * \Delta t$$

$$Position_x = 5 + (0.6) * 1.0$$

$$Position_x = 5.6$$

$$Position_y = Position_y + v_y * \Delta t \quad (16)$$

$$Position_y = 5 + (2.275) * 1.0$$

$$Position_y = 7.275$$

From the above calculations, we got to conclude the new updated value for Particle A which you can see in Table 2.

In [17] it was stated that the position of each particle represents a set of weights for each iteration, for neural network implementation; thus, the dimensionality of those particles would be the number of weights that are associated with the network.

TABLE 2: The updated value of Particle A.

Particle A	Gbest	Pbest
New Value = 5.6	5	15
New Value = 7.275	13	13

To minimize the learning error of the results, and to produce better quality ones, we need Mean Squared Error.

As such, Mean Squared Error (MSE) is the minimization of the learning error by each particle moving in the weight space. Position change comes with weight update in order to reduce particular current epoch. Epoch is the state where the particles update positions from calculating their new velocity which is used for moving forward towards new positions. These new positions are a state of new weights that are used to obtain the new error. In Particle Swarm Optimization (PSO) these weights are adopted even without new improvement. The global best position of particle is chosen after the process repetition and selection for all the particles with the lowest error. Once the satisfactory error is achieved, this process ends and once the training ends, those weights are used for training patterns by calculating the classification error, and the same set of weights is also used for network testing using testing patterns.

**2.4. Classification Problems.** Classification ranks as one of the most important implementations of Neural Networks. As it is, solving classification problems requires a lot of work and ranging all the available static patterns for classes. These patterns include various parameters, such as if related to Information Security it could be intrusion detection, for banking it could be bankruptcy prediction, and as for medical field it could be the diagnosis. Patterns are represented by vectors, which influence pattern assignment decision in classes. An example of the usage of vectors is on the medical field, where the vector component is served from the checkup data, and the Neural Networks determine to where the pattern is going to be assigned, based on the available information for it.

The input data for the Neural Networks must be in a normalized format and is done in several ways. By normalization, it is meant that all data numbers should be of the range between zero to one. Although classification has shown significant progress in various areas of Neural Networks, in 2017 [18], it was mentioned that there still are a number of unsolved issues completely. First classification issue being the amount of data which Neural Networks deal with, and secondly the prediction that often causes errors due to learning error problems.

The usage of Neural Networks classification has shown success and has been applied to various world classification tasks, from business industry, to Information Technology and to science. Specific examples of usage of Neural Networks are many, as mentioned in [19]; the usefulness of Neural Networks is seen in improving the broader family of the overall Machine Learning.

### 3. Conclusion

Learning in Neural Networks has made it possible for scientists and researchers to create various applications for multiple industries and create ease in everyday life. The methods reviewed in this paper, namely, Artificial Neural Networks, Backpropagation, and Particle Swarm Optimization have a significant role in Neural Networks in understanding real world problems and tasks, such as image processing, speech or character recognition, and intrusion detection [20, 21]

Contributions of these methods are significant; however, each category still lacks the definite success as these fields are still progressing and improving on enclosing the gap which exists between its theory and practice. The novelty of the classification concept in Artificial Neural Networks, in particular Particle Swarm Optimization and Backpropagation, means that this field of research is openly, actively, and highly researched every year. This paper has provided a brief review on the concepts, methods, and performances of Particle Swarm Optimization, Backpropagation, and Neural Networks.

In conclusion, the study on Artificial Neural Networks and its methods is promising and ongoing, especially the improvement studies that are needed in learning error side and in the classification accuracy. The improvement of these two categories is a big part for Neural Networks solutions for the new emerging field technologies like Deep Learning in Medicine, Cloud Computing, and even in Information Security [22–25].

### Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

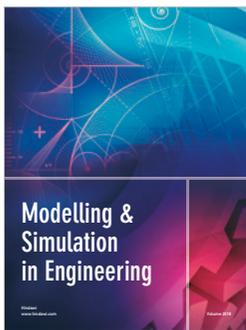
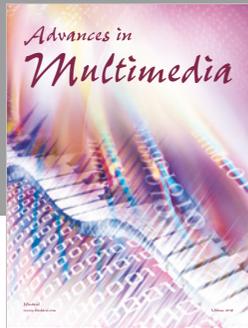
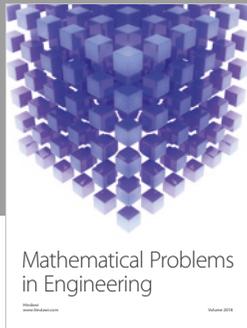
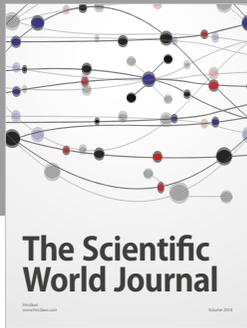
### Acknowledgments

This review paper and research have been supported by Faculty of Information Sciences and Engineering at Management and Science University, Malaysia.

### References

- [1] Z. Chiyuan, B. Samy, H. Moritz, R. Benjamin, and V. Oriol, "Understanding deep learning requires rethinking generalization," *International Conference on Learning Representations*, 2017.
- [2] B. M. Sonali and W. Priyanka, "Research Paper on Basic of Artificial Neural Network," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, no. 1, 2014.
- [3] A. Rasit, *Renewable and Sustainable Energy Reviews* 534-562, Celal Bayar University, Turkey, 2015.
- [4] T. Cho, R. Connors, and P. Araman, "Fast backpropagation learning using steep activation functions and automatic weight reinitialization," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1587–1592, Charlottesville, VA, USA.

- [5] V. Gerhard and S. Jaroslaw, "Particle Swarm Optimization," *AIAA Journal*, vol. 41, no. 8, 2003.
- [6] A. Jaswante, K. Asif, and G. Bhupesh, *IJCSNS*, vol. 14, No. 11, 2014.
- [7] A. Hamed and N. Haza, *Particle Swarm Optimization for Neural Network Learning Enhancement*, Universiti Teknologi Malaysia, 2006, Particle Swarm Optimization for Neural Network Learning Enhancement. Universiti Teknologi Malaysia.
- [8] C. Blum and D. Merkle, *Swarm Intelligence: Introduction and Applications, Natural Computing Series*, Springer, 2008.
- [9] A. P. Markopoulos, S. Georgiopoulos, and D. E. Manolakos, "On the use of back propagation and radial basis function neural networks in surface roughness prediction," *Journal of Industrial Engineering International*, vol. 12, no. 3, pp. 389–400, 2016.
- [10] F. Y. H. Ahmed, S. M. Shamsuddin, and S. Z. M. Hashim, "Improved SpikeProp for using particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 257085, 13 pages, 2013.
- [11] P. Konstantinos and V. Michael, "Particle Swarm Optimization and Intelligence," *Advances and Applications*, 2009.
- [12] D. P. Rini, S. M. Shamsuddin, and S. S. Yuhaniz, "Particle swarm optimization: technique, system and challenges," *International Journal of Computer Applications*, vol. 14, no. 1, pp. 19–26, 2011.
- [13] Y. Shi, "Particle Swarm Optimization," *IEEE Neural Network Society*, p. 13, 2004.
- [14] D. Ferguson, *Particle Swarm*, University of Victoria, Canada, 2004.
- [15] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, 2001.
- [16] F. A. Van den Bergh and A. Engelbrecht, "Cooperative Learning in Neural Networks using Particle Swarm Optimizers," *South African Computer Journal*, vol. 26, pp. 84–90, 2000.
- [17] B. Al-Kazemi and C. K. Mohan, "Training feedforward neural networks using multi-phase particle swarm optimization," in *Proceedings of the 9th International Conference on Neural Information Processing, ICONIP 2002*, New York, 2002.
- [18] Zhang. Guoqiang, "IEEE Transactions on Systems, Man and Cybernetics," *IEEE, Part C, Applications and Reviews*, vol. 30, no. 4, 2017.
- [19] M. K. Khan, W. G. Al-Khatib, and M. Moinuddin, "Automatic classification of speech and music using neural networks," in *Proceedings of the the 2nd ACM international workshop*, p. 94, Washington, DC, USA, November 2004.
- [20] I. R. Ali, H. Koliwand, and M. H. Alkawaz, "Lip syncing method for realistic expressive 3D face model," *Multimedia Tools and Applications*, vol. 77, no. 5, pp. 5323–5366, 2018.
- [21] Y. H. A. Falah, "A Survey and Analysis of Image Encryption Methods," *International Journal of Applied Engineering Research*, 2017, Research India Publications.
- [22] H. Igor, J. Bohuslava, J. Martin, and N. Martin, "Application of neural networks in computer security," *Procedia Engineering*, vol. 69, pp. 1209–1215, 2013.
- [23] R. A. Abbas, "Improving time series forecast errors by Using Recurrent Neural Networks," *Australian Journal of Forensic Sciences*, 2018.
- [24] M. H. Alkawaz, *Automated Kinship verification and identification through human facial images*, vol. 76 of *Multimedia Tools and Applications*, Kluwer Academic Publishers, 2017.
- [25] M. H. Alkawaz, *Detection of Copy-move Image forgery based on discrete Cosine transform*, Neural Computing and Applications, Springer verlag, 2018.



Hindawi

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

