

## Early fault prediction of a wind turbine using a novel ANN training algorithm based on ant colony optimization

Yunus Erođlu 

İskenderun Technical University, Department of Industrial Engineering, yunus.eroglu@iste.edu.tr

Serap Ulusam Seçkiner 

Gaziantep University, Department of Industrial Engineering, seckiner@gantep.edu.tr

Submitted: 30.08.2019

Accepted: 05.12.2019

Published: 31.12.2019



**Abstract:** The technological developments in wind energy field have reduced the investment and the operation costs. For this reason, wind farms have become more popular around the world. Increasing the share of wind energy in the market has led to the need for secure, inexpensive, and effective monitoring and control approaches. In the present work, various monitoring and control tools, which are cheap and easy to implement in wind farms using existing system data are proposed. The primary purpose of this study is to offer a new methodology, i.e. an artificial neural network (ANN) design with a novel training algorithm called Antrain ANN, in order to explore the early fault detection in a wind turbine. Our case problem is the fault detection for a wind turbine. For this issue, we used real data consisting of 873 samples with 12 inputs and one output. The models used in the work try to forecast fault occurrence before 10 minutes it happens. The proposed Antrain ANN algorithm is compared with Quick Propagation, Conjugate Gradient Descent, Quasi-Newton, Limited Memory Quasi-Newton, Online Backpropagation, and Batch Back Propagation algorithms, respectively. The results have shown that the proposed novel approach has better results in the correct classification rates than other algorithms except the Quasi-Newton and Limited Memory Quasi-Newton ones.

**Keywords:** *Ant colony algorithm, Artificial neural networks, Fault detection, Wind energy, Wind turbine*

Cite this paper as: Erođlu, Y., Ulusam Seçkiner, S., Early fault prediction of a wind turbine using a novel ANN training algorithm based on ant colony optimization. *Journal of Energy Systems* 2019, 3(4), 139-147, DOI: 10.30521/jes.613315

## 1. INTRODUCTION

The wind energy field is of a crucial and an indispensable element of present world. The factors such as the limited resources, excessive and growing consumption, and high costs make it mandatory to have a wind energy management perspective as well as the efficient usage of the wind farms. In this context, efficient and well-planned wind energy management is widely accepted as an emerging factor for the future life of humanity, which requires not only the technical innovation and the alternative energy resources but also the awareness of all consumers about the depletion of resources and efficient use of wind energy.

Fault diagnosis and condition monitoring issues are nowadays used with intelligent algorithms to predict unwanted breakdowns. There are model-based and non-model based approaches used Supervisory Control and Data Acquisition (SCADA) data of wind energy systems. Expert systems are one of the essential novel condition monitoring CM methodologies to identify and predict faults with a rule-based system [1]. The acquisition and new knowledge integration are two main capabilities of expert systems. Artificial Intelligence (AI) techniques are also more popular than before in novel CM methods over recent years [2, 3]. Zaher and McArthur improved a multi-agent-based wind turbine condition monitoring system [4]. In 2016, Qui et al., used the simulation of oil temperature data to predict the faults [5]. In addition, a deep auto-encoder model has been presented by Wang et al., to monitor wind turbine blade breakage [6]. In another work, Liu has contributed to an artificial immune system to predict gearbox faults [7].

Historically, the empirical Mode Decomposition technique has also been presented as a sufficiently intelligent condition monitoring methodology [8]. Yang and Tian studied on the monitoring power quality by the electrical signals measured from the wind turbine generator [9]. The time series modeling, Monte Carlo methods, and Markov Chains are also presented as condition monitoring approaches in this manner [10].

Analyzing the normal behavior of SCADA data is also presented as a CM tool, so that the anomalies can be identified. Schlechtingen et al., used a machine learning approach model to learn the normal behavior of a healthy component [11]. By the way, many other studies use data mining and statistical methods to analyze the jerk data obtained from the SCADA system of a wind turbine and to detect faults of a wind turbine in the time domain [12-15].

To sum up, the review of the literature presents the following conclusions:

Higher investment rates occur due to the increasing technical capacity of a wind turbine.

The number of installed wind turbines is rapidly increasing.

The cost of proactive maintenance is high, therefore, there is a common need for fault prediction and health monitoring.

CM has a growing trend for wind energy researchers.

CM companies are growing in the wind energy area.

Novel techniques need to be more focused on developing smart CM and control strategies.

SCADA data has great importance in monitoring issues of wind energy systems.

By the way, Artificial Neural Network (ANN) can be a beneficial way to accomplish an efficient and a well-planned wind energy management, especially in the field of fault prediction. ANN models try to learn the relationship between input and output data within a training approach. There are many algorithms to train these models. On the other hand, current training algorithms, such as

Backpropagation, tend to be stuck in local minimums, preventing the learning process. Hence, the heuristic approaches are used to train neural networks. The primary purpose of the training process is to minimize the model error. The weights, which are continuous coefficients of the network, are used to evaluate the target of the inputs. The outputs of the networks have errors. Therefore, any successful methodology providing global minimum (i.e. minimum of errors) can be adapted to ANN models as a training approach.

This paper proposes a new ANN training algorithm, namely Antrain ANN, based on ant colony optimization for continuous minimization problems using a novel pheromone update strategy (ACO-NPU) [16] for fault prediction problems in the wind energy area.

The most commonly used learning algorithms for ANN models are the Levenberg–Marquardt, and Backpropagation algorithms which involve derivative methodologies. There are many other methods, such as particle swarm optimization, ant colony optimization, taboo search, and genetic algorithm, to find the best values of the weights of the neurons. In the early years of the ANN model studies, there is an only single-layer model, which was firstly introduced by McCulloch and Pitts [17]. Those single-layer models consist of only input and output layers and a linear output function. This was the main disadvantage of the model, disallowing solving nonlinear models. There are many other ANN models, such as generalized mean neuron [18], multiple spiking neural network model [19], and geometric mean neuron model [20].

To our knowledge, this study is the first for fault prediction of a wind turbine using artificial neural networks trained by the ant colony algorithm. In this context, some reasons like the gap in the literature about this topic, the growing wind energy sector, and increasing its energy consumption, efforts to find alternative energy technologies and sources reducing energy consumption form the motivation for this study. By reviewing and iterating what one knows in relation to new ways and solution methods for wind energy applications and how wind energy professionals mostly benefit from this data and information, one considers that this study can be incentive and assistant for concerned professionals, researchers, practitioners, having the agenda and idea to study energy applications in wind farms.

The paper consists of as follows: In section 2, a novel training algorithm, namely Antrain ANN (for ANN models using ACO-NPU) is presented. In section 3, as a case study, early fault prediction of a wind turbine solved by Antrain ANN is presented. Finally, the results and conclusions are discussed.

## 2. MATERIAL and METHOD

The training process of an ANN is a crucial step in the forecasting models to represent the sample data. In the current literature, there are many modern approaches using heuristics in the learning process of ANN models. Chau [21] contributed a particle swarm optimization approach to the learning process of the ANN model, which was used to predict the stage of the Shing Mun River. A supervised learning algorithm for multiple spiking ANN models is presented by Ghosh-Dastidar and Adeli [19] to detect the seizures in epilepsy. A fuzzy clustering methodology was used to define the training sets of ANN models to predict the intrusion detection systems by Wang et al. [22]. Baş et al., [23] proposed a robust training algorithm for a multiplicative ANN. Their primary purpose was to contribute at a new ANN learning model, enabling to learn with data outliers. Mohamad et al. [24] used a genetic algorithm approach in the training of an ANN model to minimize the prediction errors of the ripping process. Lee et al. [25] proposed a harmony search algorithm to determine the near-global optimal initial weights in the training of the ANN model, which is used to predict the stability number of breakwater armor stones. A dual-stage multi-resource data training with the multi-objective genetic algorithm was presented by Ganguly et al., [26] and applied in alloy design problems. A differential evolution algorithm is contributed to the

train of multiplicative neuron model for forecasting by Baş [27]. This study presented a heuristic approach to train ANN models with a novel pheromone updating strategy of ant colony optimization called Antrain ANN.

In the current literature, ACO based approaches are used in training procedures of ANN models. Li and Chung [28] presented a novel backpropagation ANN model with a combination of ACO. Their approach has based on the vital factor of the optimal weight determination by the concentration of pheromone laid of the artificial ants moving on the connection path. Socha and Blum [29] presented their continuous ant colony optimization technique in order to train of feed-forward neural networks for pattern classification. The main characteristic of the contributed algorithm is that the ACO enables to find the minimum errors of the backpropagation process or standard Levenberg–Marquardt approach. Therefore, it can be said that it is a hybrid model of Backpropagation and/or Levenberg–Marquardt training algorithm in this respect. Their proposed approach has been tested on a classification problem with a feed-forward approach of the ANN model. Besides, the fact that another recent work on ACO based ANN training approach is contributed by Saghatforoush et al. [30], they only consider ACO to optimize given patterns of ANN model result.

This study proposes a new approach to train ANN models, exactly differing from other aforementioned current studies. Fig. 1 gives the flow diagram of the proposed approach. The first step is the initialization of the parameters and getting input and target data to form the user.  $m$  represents the number of ants,  $T$  is the maximum iteration number,  $k$  is the archive size, sample size is defined according to sample of the input and target data,  $N_{wi}$  is the number of hidden layer weights according to the input parameters.  $N_{wo}$  is the number of output layer weights according to number of target parameters,  $B_{wi}$  is the number of input bias weights,  $B_{wo}$  is the number of output bias weights, weight ranges are the interval of the initial weights,  $minError$  is the minimum target error defined by the user,  $MaxRepeat$  is the number of maximum repetition of the algorithm if the  $minError$  is not found. Then, all inputs and target data are normalized between [0,1] according to Eq. (1):

$$\text{NormalizedData} = \frac{(\text{CurrentData} - \text{MinimumOfData})}{(\text{MaximumOfData} - \text{MinimumOfData})}, \quad (1)$$

The input biases and target biases are identity matrix with the number of hidden layers and output layer, respectively. The next step is generating an archive size of random weights of input layers, output layers, input biases, and output biases in the range of defined intervals for each one. Now, the algorithm is ready to evaluate the initial fitness values of each element in the archive.

Fig. 2 represents the basic architecture of the ANN models. There is an input layer, taking the input parameters (i.e.  $I_1, I_2, \dots, I_n$ ). The proposed approach uses the normalized input values to prevent the outliers of different kinds of input parameters. The second layer is the hidden neuron layer (i.e.  $H_1, H_2, \dots, H_m$ ). Every hidden neuron has its weights ( $W_i$ ), which needs to be optimized. By the way, every hidden neuron has a bias value ( $B_i$ ) with a weight of  $B_{wi}$ . The output of the hidden neuron is computed with the following Eq. (2),

$$\text{Output}_{\text{hidden}i} = \sum_{l=1}^n \sum_{j=1}^m I_l W_{lj} + \sum_{j=1}^m B_{wi} B_{wj}, \quad (2)$$

where  $n$  is the number of inputs,  $m$  is the number of the hidden neuron, and  $W_i$  is the weights of hidden neurons,  $B_{wi}$  is the weight of hidden neuron biases. The sigmoid function is used as an activation function in this training approach. Hence, the hidden neuron outputs are activated with the following Eq. 3:

$$\text{Net}_i = 1/(1 + (e^{-\text{Output}_{\text{hidden}i}})), \quad (3)$$

Then, thus activated hidden neuron output is used to evaluate network output with the following Eq. 4:

$$Output_{oi} = \sum_{j=1}^m \sum_{p=1}^k Net_{ij} Wo_p + \sum_{p=1}^k Bias_{op} Bwo_p, \quad (4)$$

where  $k$  is the number of outputs,  $Wo$  is the output layer weights coming from the input layer,  $Bwo$  is the weights of the output layer biases. Then, the result of the output layer is also activated with the sigmoid function as follows in Eq. (5):

$$Output = 1/(1 + (e^{-Output_i})), \quad (5)$$

Finally, the model for each sample  $i$  error is computed with Eq. (6):

$$Error_i = |Target_i - Output_i|, \quad (6)$$

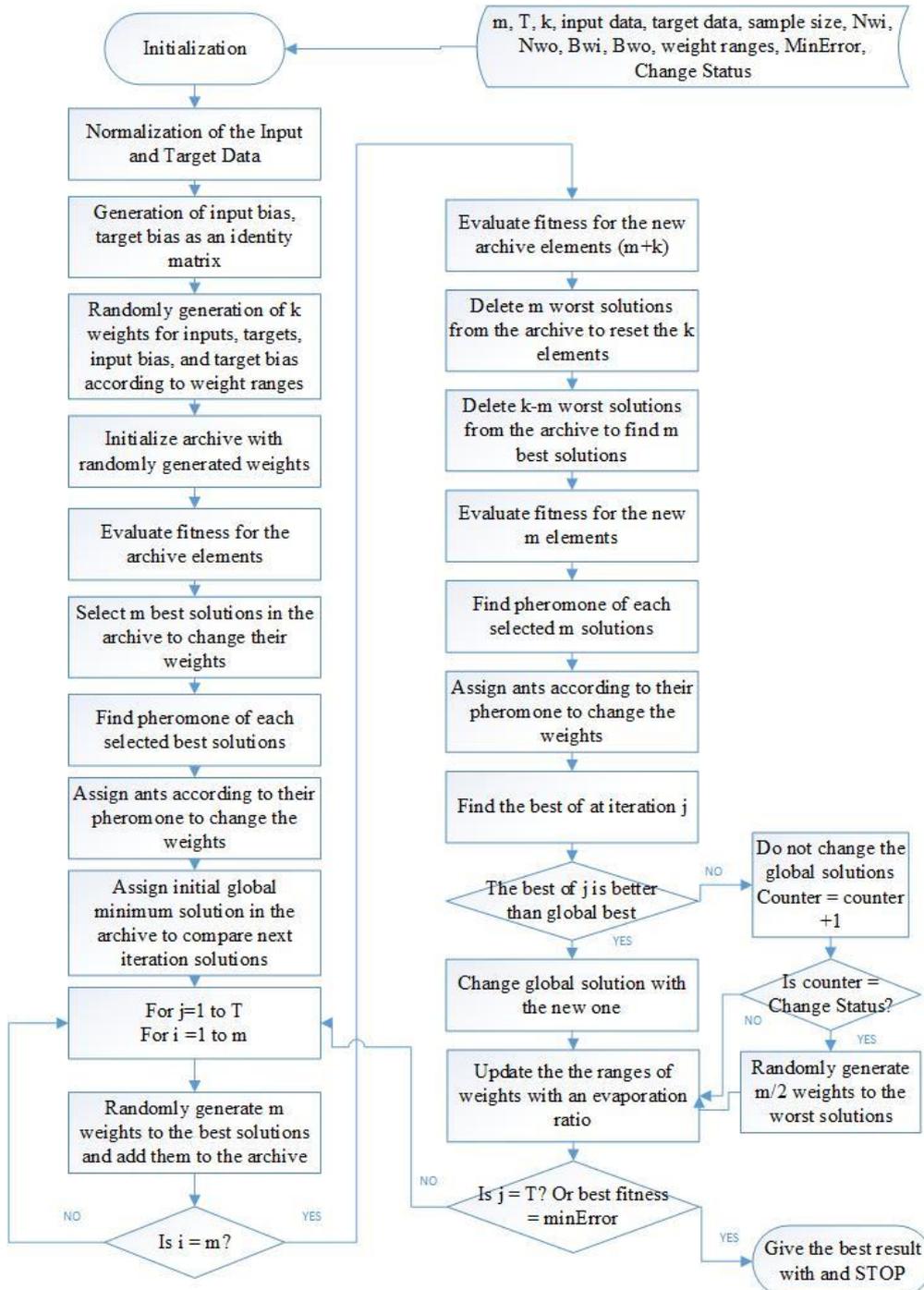


Figure 1. Flow diagram of Antrain ANN training process

The proposed approach computes all sample evaluations as a batch process for each element in the archive. Then the sum of squared errors (SSE) is computed with the following Eq. (7):

$$SSE_k = \sqrt{\sum_{i=1}^{sample} Error_i^2} \quad (7)$$

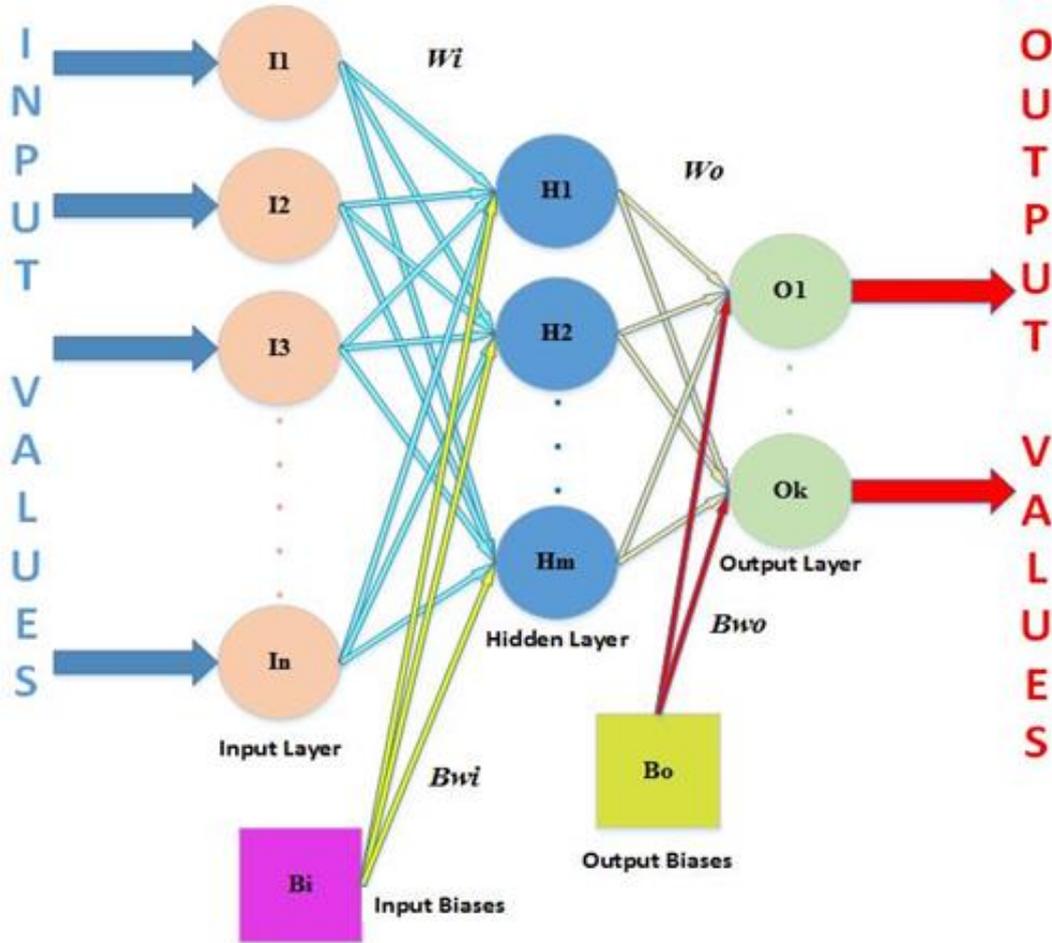


Figure 2. A general framework of an ANN structure

After these initialization steps, the ant colony procedure is prepared by selecting  $m$  best solutions according to their SSE values in the archive. The normal probabilities of those selected solutions are evaluated with Eq. (8),

$$Gaussian_m = e^{\frac{(SSE_m - \min(SSE))^2}{2\delta}}, \quad (8)$$

where  $m$  is the indices of the selected solution, and  $\delta$  is the standard deviation for a normal probability distribution. Then those probabilities are normalized to find pheromone values of each solution with Eq. 9:

$$Pheromone_m = \frac{Gaussian_i}{\sum_{i=1}^m Gaussian_i}, \quad (9)$$

The minimum of those initial solutions is assigned to the global minimum. Then the algorithm starts to find new solutions. Initially, random  $m$  weights are produced by using the best  $m$  solutions and add them to the archive. Then, evaluate fitness to those  $k + m$  sized solutions with Eqs. (6) and (7). Delete the worst  $m$  solution to gain a  $k$  sized archive. Then, define the best  $m$  solution in the archive to compute their pheromone values with Eqs. (8) and (9). Find the best solution for the current iteration and compare

it to the global best one. If the new best is less than the global one, change the best weights and solution with the new one, else continue with updating the ranges of the weights with an evaporation rate. Also, the proposed approach uses a counter to limit the worst results. If the counter reaches the Change Status value, half of the number of ants of the worst solutions in the archive changed with new weights, which are randomly generated from the worst results by using global best weights. While the maximum iteration number or the minimum required error reached, the algorithm stops and gives the best outputs and weights for the network.

### 3. PROBLEM DESCRIPTION

Early fault prediction of a wind turbine is modeled with real data of a wind farm in Turkey. The primary purpose is predicting the faults 10 minutes before it happens using current SCADA data. The data set consists of 873 samples with 12 inputs and 1 output. The input parameters are 10-minute averaged generator speed, hydraulic oil temperature, rotor speed, wind speed, generated power, yaw angle, current phase 1, current phase 2, current phase 3, voltage phase 1, voltage phase 2, and voltage phase 3 of the current time. Table 1 summarizes the descriptive statistics of inputs. The output is the fault occurrence of the next 10-minute period. The data set has 673 faults and 200 normal operation data. The data set is partitioned as training (813 cases), testing (40 cases), and validation parts (20 cases).

The proposed Antrain ANN training algorithm is tested on wind turbine fault detection problems. The ANN architecture is 12 input, one output single layer ANN with 12 hidden layers. The parameters of the algorithm are set for nearly all cases problem as follows:  $m=60$ ,  $k=100$ ,  $T=100$ , Change Status=10,  $W_i$  range is [-10, 10],  $W_o$  range is [-10, 10],  $B_{wi}$  range is [-10, 10],  $B_{wo}$  range is [-10, 10], minimum error rate is 0.005, the evaporation rate for the weight ranges is 0.9, and  $\delta=0.0005$ .

*Table 1. The descriptive statistics of Wind Turbines Faults Test Data*

| Descriptive    | Generator Speed (RPM)    | Hydraulic Oil Temperature (°C) | Rotor Speed (RPM)        | Wind Speed (m/s)       | Generated Power (kW)   | Yaw Angle (°C)         |
|----------------|--------------------------|--------------------------------|--------------------------|------------------------|------------------------|------------------------|
| Range          | 1687.00                  | 35.00                          | 16.00                    | 22.70                  | 2899.40                | 357.30                 |
| Minimum        | 0.00                     | 17.00                          | 0.00                     | 0.50                   | -20.40                 | -179.20                |
| Maximum        | 1687.00                  | 52.00                          | 16.00                    | 23.20                  | 2879.00                | 178.10                 |
| Mean           | 531.49                   | 45.20                          | 5.04                     | 7.62                   | 335.99                 | 0.48                   |
| Std. Deviation | 638.21                   | 4.84                           | 6.08                     | 4.03                   | 620.42                 | 37.17                  |
| Variance       | 407317.80                | 23.40                          | 36.95                    | 16.21                  | 384927.16              | 1381.26                |
| Skewness       | 0.60                     | -1.86                          | 0.60                     | 1.04                   | 2.12                   | 0.21                   |
| Kurtosis       | -1.39                    | 4.58                           | -1.39                    | 0.95                   | 3.81                   | 11.15                  |
| Descriptive    | Current Phase 1 (Ampere) | Current Phase 2 (Ampere)       | Current Phase 3 (Ampere) | Voltage Phase 1 (Volt) | Voltage Phase 2 (Volt) | Voltage Phase 3 (Volt) |
| Range          | 1629.50                  | 1648.10                        | 1638.70                  | 584.40                 | 584.00                 | 580.20                 |
| Minimum        | 1.00                     | 1.00                           | 1.00                     | 28.80                  | 29.00                  | 28.70                  |
| Maximum        | 1630.50                  | 1649.10                        | 1639.70                  | 613.20                 | 613.00                 | 608.90                 |
| Mean           | 198.50                   | 201.75                         | 203.76                   | 569.96                 | 570.39                 | 567.05                 |
| Std. Deviation | 344.34                   | 350.36                         | 352.12                   | 57.91                  | 57.81                  | 57.62                  |
| Variance       | 118567.76                | 122755.53                      | 123988.13                | 3353.30                | 3342.55                | 3319.88                |
| Skewness       | 2.14                     | 2.12                           | 2.11                     | -7.43                  | -7.45                  | -7.42                  |
| Kurtosis       | 3.92                     | 3.84                           | 3.79                     | 58.06                  | 58.39                  | 57.94                  |

#### 4. RESULTS and DISCUSSIONS

The proposed algorithm runs ten times for the case problem. The results of these run trials given with the best, average, and the worst ones to compare current training algorithms such as Quick Propagation, Conjugate Gradient Descent, Quasi-Newton, Limited Memory Quasi-Newton, Levenberg-Marquardt, Online Backpropagation, Batch Back Propagation which are available in Alyuda NeuroIntelligence Network Software. The data of each case are partitioned to train, validation, and test sets. The validation sets are used as test sets for the Antrain ANN algorithm because it does not require validation sets. The performance parameter is selected as Correct Classification Rate (CCR) for classification outputs. The CCR is the percentage of the correct classification of the proposed approach.

Table 2 gives the comparison of Antrain ANN with current training algorithms. While the best algorithm is Limited Memory Quasi-Newton, Antrain ANN has provided a competitive CCR rate with 86%.

*Table 2. The comparison of Antrain ANN with current training algorithms for the test problem*

| Algorithms                 | CCR %   | Algorithms                  | CCR %   |
|----------------------------|---------|-----------------------------|---------|
| Antrain ANN – Best         | 86.4833 | Quasi-Newton                | 86.7125 |
| Antrain ANN – Average      | 82.1305 | Limited Memory Quasi-Newton | 89.0034 |
| Antrain ANN – Worst        | 77.3195 | Online Backpropagation      | 32.6460 |
| Quick Propagation          | 85.2234 | Batch Back Propagation      | 77.0905 |
| Conjugate Gradient Descent | 86.2543 |                             |         |

#### 5. CONCLUSIONS

The current study contributes to a novel training algorithm for ANN models with an ACO approach, which is called as Antrain ANN for wind turbine fault prediction problems. The performance of the Antrain ANN is tested on different wind turbine fault detection problems. The results have proven that the proposed methodology can compete with other traditional training algorithms, used in the ANN models. Because the nature of the Antrain ANN algorithm is heuristic, there are many parameters which can affect the performance of the methodology. By the way, parameter optimization is another crucial issue when the heuristic approaches are used, thereby, the parameter settings of the algorithm are not considered in this study. Consequently, the developed Antrain ANN algorithm can train many kinds of ANN models with different types of problems. Future works can focus on parameter optimization of the proposed approach. Also, for further time-period, such as 1-hour, 1-day, or 1-week, the predictions of faults can be realized as another crucial problem for wind turbines.

#### REFERENCES

- [1] Tautz-Weinert, J., Watson, S. J. Using SCADA data for wind turbine condition monitoring – a review, *IET Renewable Power Generation* 2016, 11(4), 382-394.
- [2] Yang, W., Tavner, P. J., Crabtree, C. J., Feng, Y., Qiu, Y. Wind turbine condition monitoring: technical and commercial challenges, *Wind Energy* 2014, 17(5), 673–693.
- [3] Kandukuri, S. T., Klausen, A., Karimi, H. R., Robbersmyr, K. G. A review of diagnostics and prognostics of low-speed machinery towards wind turbine farm-level health management, *Renewable and Sustainable Energy Reviews* 2016, 53, 697–708.
- [4] Zaher, A. S., McArthur, S. D. J. A Multi-Agent Fault Detection System for Wind Turbine Defect Recognition

- and Diagnosis. In *Power Tech, 2007 IEEE Lausanne* (pp. 22–27). Presented at the Power Tech, 2007 IEEE Lausanne. doi:10.1109/PCT.2007.4538286
- [5] Qiu, Y., Feng, Y., Sun, J., Zhang, W., Infield, D. Applying thermophysics for wind turbine drivetrain fault diagnosis using SCADA data. *IET Renewable Power Generation* 2016, 10(5), 661–668.
- [6] Wang, L., Zhang, Z., Xu, J., Liu, R. Wind Turbine Blade Breakage Monitoring with Deep Autoencoders, *IEEE Transactions on Smart Grid* 2016, 1(1), 99.
- [7] Liu, T. Fault diagnosis of gearbox by selective ensemble learning based on artificial immune algorithm. In *2016 3rd International Conference on Systems and Informatics (ICSAI)* 2016, 460–464. Presented at the 2016 3rd International Conference on Systems and Informatics (ICSAI). doi:10.1109/ICSAI.2016.7810999
- [8] Yang, W., Tavner, P. J., Crabtree, C. An Intelligent Approach to the Condition Monitoring of Large Scale Wind Turbines. In *European Wind Energy Conference* 2009. Marseille, France.
- [9] Yang, W., Tian, S. W. Research on a power quality monitoring technique for individual wind turbines. *Renewable Energy* 2015, 75, 187–198. doi:10.1016/j.renene.2014.09.037
- [10] Lu, B., Li, Y., Wu, X., Yang, Z. A review of recent advances in wind turbine condition monitoring and fault diagnosis. In *IEEE Power Electronics and Machines in Wind Applications* 2009. PEMWA 2009 (pp. 1–7). Presented at the IEEE Power Electronics and Machines in Wind Applications, 2009. doi:10.1109/PEMWA.2009.5208325
- [11] Schlechtingen, M., Santos, I. F., Achiche, S. Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 1: System description. *Applied Soft Computing* 2013. 13(1) 259–270. doi:10.1016/j.asoc.2012.08.033
- [12] Zhang, Z., Verma, A., Kusiak, A. Fault Analysis and Condition Monitoring of the Wind Turbine Gearbox. *IEEE Transactions on Energy Conversion* 2012. 27(2) 526–535. doi:10.1109/TEC.2012.2189887
- [13] Zhang, Z., Kusiak, A. Monitoring Wind Turbine Vibration Based on SCADA Data. *Journal of Solar Energy Engineering* 2012, 134(2) 021004–021004. doi:10.1115/1.4005753
- [14] Kusiak, A., Verma, A. A Data-Driven Approach for Monitoring Blade Pitch Faults in Wind Turbines. *IEEE Transactions on Sustainable Energy* 2011. 2(1) 87–96. doi:10.1109/TSSTE.2010.2066585
- [15] Kusiak, A., Zheng, H., Song, Z. Models for monitoring wind farm power. *Renewable Energy* 2009. 34:3: 583–590. doi:10.1016/j.renene.2008.05.032
- [16] Seçkiner, S.U., Eroğlu, Y., Emrullah, M., Dereli T. Ant colony optimization for continuous functions by using novel pheromone updating. *Applied mathematical and Computation* 2013. 219(9), 4163-4175.
- [17] McCulloch W. S. Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics* 1943. 5(4), 115-133.
- [18] Yadav, R.N., Kumar, N., Kalra, P.K., John, J. “Learning with generalized-mean neuron model”, *Neurocomputing*. 2006. 69(16-18), 2026-203.
- [19] Ghosh-Dastidar S. Adeli, H. A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection. *Neural Networks* 2009. 22(10) 1419-1431.
- [20] Shiblee, M. Chandra, B. ve Kalra, P. K. Learning of geometric mean neuron model using resilient propagation algorithm. *Expert Syst. Appl.* 2010. 37(12), 7449-7455.
- [21] Chau, K. W. Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River. *J. Hydrol.* 2006. 329(3-4), 363-367.
- [22] Wang, G. Hao, J. Ma J., ve Huang, L. A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering, *Expert Syst. Appl.* 2010. 37(9), 6225-6232.
- [23] Bas E., Uslu V. R., Egrioglu E. Robust learning algorithm for multiplicative neuron model artificial neural networks. *Expert Syst. Appl.* 2016. 56, 80-88.
- [24] Mohamad E. T., Faradonbeh R. S., Armaghani D. J., Monjezi M., Majid M. Z. A. An optimized ANN model based on genetic algorithm for predicting ripping production. *Neural Comput. Appl.* 2016. 1-14.
- [25] Lee A., Geem Z. W., Suh K. D. Determination of Optimal Initial Weights of an Artificial Neural Network by Using the Harmony Search Algorithm: Application to Breakwater Armor Stones. *Appl. Sci.* 2016. 6(6), 164.
- [26] Ganguly S., Patra A., Chattopadhyay P. P., Datta S. New training strategies for neural networks with application to quaternary Al–Mg–Sc–Cr alloy design problems. *Appl. Soft Comput.* 2016. 46, 260-266.
- [27] Bas E. The Training of Multiplicative Neuron Model Based Artificial Neural Networks With Differential Evolution Algorithm For Forecasting. *J. Artif. Intell. Soft Comput. Res.* 2016. 6(1), 5-11.
- [28] Li J.B. and Chung Y.K. A Novel Back-propagation Neural Network Training Algorithm Designed by an Ant Colony Optimization. *2005 IEEE/PES Transmission Distribution Conference Exposition: Asia and Pacific* 2005. 1-5.
- [29] Socha K. and Blum C. An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training. *Neural Comput. Appl.* 2007. 16(3), 235-247.
- [30] Saghatforoush A., Monjezi M., Faradonbeh R. S., Armaghani D. J. Combination of neural network and ant colony optimization algorithms for prediction and optimization of flyrock and back-break induced by blasting. *Eng. Comput.* 2016. 32(2), 255-266.