

Comparing Two Methods of Neural Networks to Evaluate Dead Oil Viscosity

Meysam Dabiri-Atashbeyk¹, Mehdi Koolivand-Salooki², Morteza Esfandiyari^{3,*}, and Mohsen Koulivand⁴

¹ M.S., National Iranian South Oil Field Company, Ahvaz, Iran

² Senior Process Researcher, Gas Research Division, Research Institute of Petroleum Industry (RIPI), Tehran, Iran

³ Assistant Professor, Department of Chemical Engineering, University of Bojnord, Iran

⁴ M.S. Student, Department of Engineering, Borujerd Branch, Islamic Azad University, Borujerd, Iran

Received: December 18, 2016; *revised:* February 11, 2017; *accepted:* October 21, 2017

Abstract

Reservoir characterization and asset management require comprehensive information about formation fluids. In fact, it is not possible to find accurate solutions to many petroleum engineering problems without having accurate pressure-volume-temperature (PVT) data. Traditionally, fluid information has been obtained by capturing samples and then by measuring the PVT properties in a laboratory. In recent years, neural network has been applied to a large number of petroleum engineering problems. In this paper, a multi-layer perception neural network and radial basis function network (both optimized by a genetic algorithm) were used to evaluate the dead oil viscosity of crude oil, and it was found out that the estimated dead oil viscosity by the multi-layer perception neural network was more accurate than the one obtained by radial basis function network.

Keywords: Dead Oil Viscosity, Radial Basis Function (RBF), Multi-layer Perceptron (MLP), Genetic Algorithm, Neural Network

1. Introduction

The reliable measurement and prediction of phase behavior and properties of petroleum reservoir fluids are essential in designing optimum recovery processes and enhancing hydrocarbon production. In most hydrocarbon reservoirs, fluid composition varies vertically and laterally in the formation. The PVT properties can be obtained from a laboratory experiment using representative samples of the crude oils. However, the values of reservoir liquid and gas properties must be computed when detailed laboratory PVT data are not available. In recent years, artificial neural network (ANN) technology has been applied to a large number of petroleum engineering problems. These problems include diverse applications such as dew-point pressure prediction for retrograde gases (Abedini et al., 2011; González et al., 2003) and well permeability estimation of reservoirs (Saemi et al., 2007). Applications of ANN technology to PVT modeling include the prediction of oil bubble point pressure and formation-volume factor (Al-Marhoun and Osman, 2002; Gharbi et al., 1999) and the estimation

* Corresponding Author:
m.esfandiyari@ub.ac.ir

of oil viscosity at pressures below the bubble point (Ayoub et al., 2007). Varotsis et al. (Varotsis et al., 1999) developed an ANN-based method to predict properties as a function of pressure for oils and gas condensates. These ANN-PVT applications all require laboratory-measured property values such as saturation pressure, stock-tank-oil API gravity, and gas gravity as their inputs.

This work demonstrates the novelty of a hybrid genetic algorithm-artificial neural network (GA-ANN) and genetic algorithm-radial basis function (GA-RBF), which combines both stochastic and deterministic routines for improved optimization results. The new hybrid genetic algorithm developed is applied to PVT data sets. For the case studies, the hybrid genetic algorithm found a better optimum candidate than those reported by the ANN and the RBF networks alone.

Optimization is crucial in analyzing experimental results and models. Stochastic optimization performs well where deterministic optimization fails. Problems with high dimensionality or very large variable spaces cannot be solved by deterministic methods in a reasonable runtime. Stochastic algorithms can search the variable space and can dynamically bias the candidate generation toward the optimum solution. Stochastic algorithms are also crucial in situations where deterministic methods cannot be applied; examples include a machine part failure, optimum critical path problems, decay fission products etc. Genetic algorithms (GA's) utilize stochastic operations such as crossover and mutation on a population to make a change of generation. Crossover combines the substructures of parents to produce new individuals. Crossover is the core of the genetic algorithm and sets it apart from other stochastic methods such as simulated annealing. Mutation is another operation that helps the algorithm to prevent local convergence and search the global variable space. A simple genetic algorithm with operators such as crossover, mutation, and elitism yields good results in practical optimization problems compared to a deterministic algorithm. Few works have been carried out on understanding the effect of the genetic algorithm operators such as crossover, mutation, and elitism on the function convergence (Barhen et al., 1997; Basso, 1982; McDiarmid, 1991). All the drawbacks of the optimization algorithms are mainly due to poorly known fitness functions that generate bad chromosome blocks, whereas only good chromosome blocks can cross over. On the other hand, the hybridization of the genetic algorithm with a deterministic algorithm helps to overcome these problems and to achieve the globally optimum solution applicable to any industry requiring optimization, especially for problems with high dimensionality and large domain spaces which are not readily solved by traditional deterministic or stochastic algorithms.

This study has proved the hypothesis that the hybridization of the genetic algorithm with a deterministic algorithm improves the optimum solution obtained by statistical methods alone. This hybrid genetic algorithm can successfully be applied to various processes and can achieve better optimized results compared to the traditional methods (Barhen et al., 1997; Basso, 1982; Caprani et al., 1993).

2. MLP and RBF networks

The feed-forward neural networks are the most popular architectures due to their structural flexibility, good representational capabilities, and the availability of a large number of training algorithms (Khandekar et al., 2008). This network consists of neurons arranged in layers in which every neuron is connected to all the neurons of the next layer (a fully connected network) (Sargolzaei et al., 2015). MLP and RBF networks are two kinds of feed-forward neural network with different transfer functions. The MLP network consists of an input layer, a hidden layer, and an output layer. The input nodes receive the data values and transfer them to the first hidden layer nodes. Each one collects the input from all the input nodes after multiplying each input value by a weight, attaches a bias to this sum, and passes on the results through a non-linear transformation like the sigmoid transfer function.

This forms the input either to the second hidden layer or to the output layer that operates similar to the hidden layer (Sargolzaei and Moghaddam, 2013). The resulting transformed output from each output node is the network output. The network needs to be trained using a training algorithm such as back propagation, cascade correlation, and conjugate gradient. The goal of every training algorithm is to reduce this global error by adjusting the weights and biases (Abedini et al., 2012; Esfandyari et al., 2016).

The basic element of a multi-layer perceptron (MLP) neural network is the artificial neuron which performs a simple mathematical operation on its inputs.

It is important to note that ANN has several limitations as follows (Samui, 2011):

- ANN could not provide enough information about the relative importance of different parameters;
- The knowledge obtained through the training of the net is stored in an implicit way; thus, it is very difficult to reasonably interpret the overall structure of the network;
- Slow convergence speed, less generalizing performance, arriving at local minimum, and overfitting problems are some inherent drawbacks of ANN;

There are some modifications such as optimizing by genetic algorithm, using hybrid ANN-fuzzy methods etc. to face these problems.

Moreover, MLP has the following drawbacks:

- it cannot be parallelized;
- it needs an additional parameter, i.e. μ ;
- it may cause additional zigzagging;
- its step length still depends on partial derivatives.

As its name implies, radially symmetric basis function is used as activation functions of hidden nodes. The transformation from the input nodes to the hidden nodes is non-linear, and the training of this portion of the network is generally accomplished by an unsupervised fashion. The training of the network parameters (weight) between the hidden and output layers occurs in a supervised fashion based on the target outputs.

3. Results and discussion

3.1. Genetic optimization of multilayer perceptron method

The back-propagation training module was used to evaluate all chromosomes; in other words, after the parameter values for each chromosome were translated into the predefined ANN, the network was trained on the training data set, and the cross-validation data set was used to prevent the overfitting problem and to test if the stopping criteria were satisfied. In this study, the training process of the back propagation network (BPN) stopped after a maximum of 6000 epochs or when there was no improvement in the mean squared error (MSE) (because of memorizing) for 100 epochs on cross-validation data set. The fitness of every chromosome was evaluated by measuring the MSE values, which are the estimated results on a cross-validation data set. A better network has a lower training error but requires a higher fitness value. The cross-validation data were not used to train the ANN model, but were used to test the ANN model at the training stage.

This process of phenotypic fitness measurement, selection, cross-over recombination, and mutation was iterated through 27 generations; then, the network with the lowest error in the 24 generations was

designated as the optimal evolved network. Table 1 illustrates the summarized report of the best fitness and the average fitness values of all the data. Also, the corresponding plots resulted from this table are shown in Figures 1 and 2. For each of these plots, the minimum MSE, the generation of this minimum, and the final MSE are displayed across all the generations.

Table 1
Performance of GA-MLP model for test data set.

Performance	Viscosity
Mean squared error (MSE)	0.011134496
Normalized mean squared error (NMSE)	0.009357474
Mean absolute error (MAE)	0.081647241
Min absolute error	0.017183716
Max absolute error	0.18898419
R	0.997475672

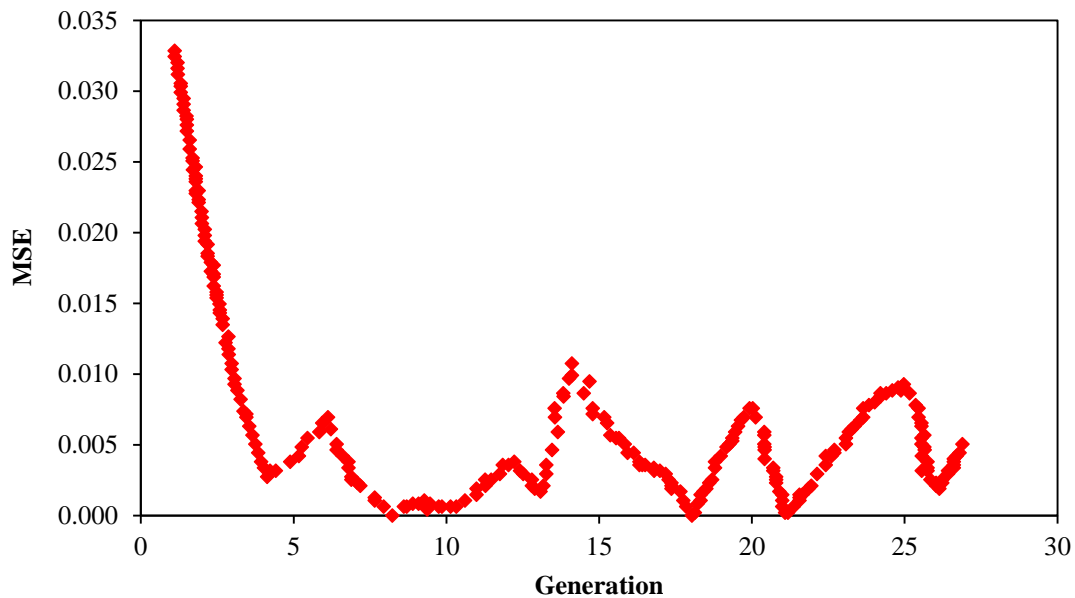


Figure 1

Average fitness (MSE) versus generation in GA-MLP.

In Figure 1, the average fitness achieved during each generation of the optimization is illustrated. The average fitness is the average of the minimum MSE (cross validation MSE) taken across all of the networks within the corresponding generation. The fitness function is an important factor for the convergence and the stability of the genetic algorithm. The collision avoidance and the shortest distance should be considered in path planning. Therefore, the smallest fitness value was used to evaluate the convergence behavior of the genetic algorithm. Figure 2 demonstrates the best fitness value versus the number of generation. In this case, to evaluate the hybrid model generalization on the chosen data set or the test data points, the performance of each model is reported in Table 1.

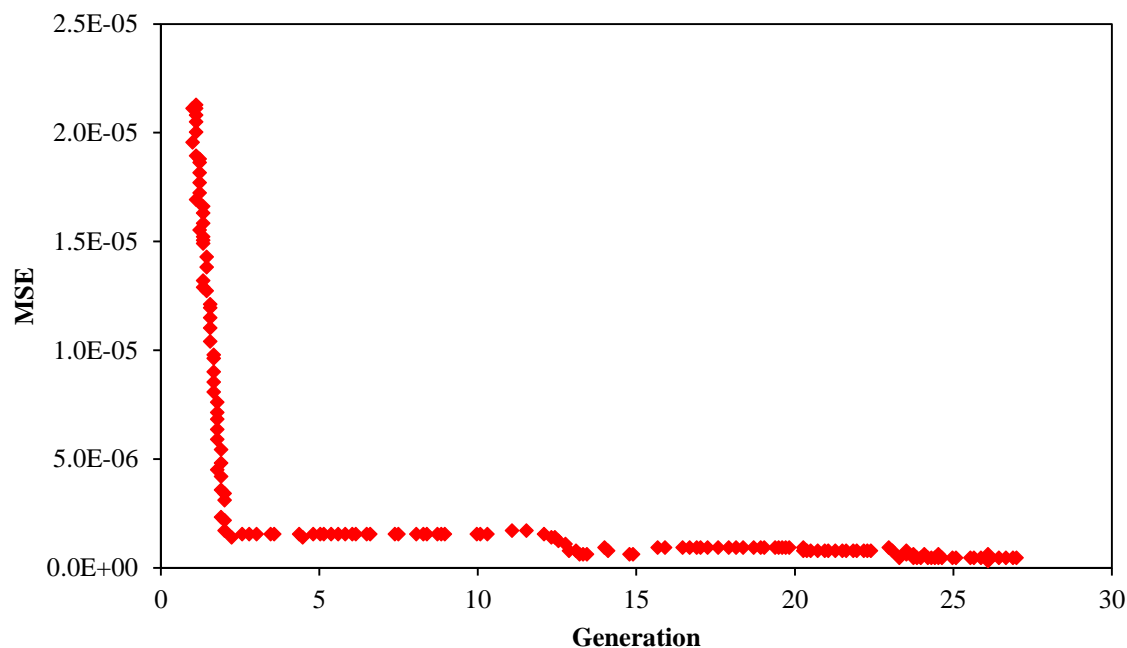


Figure 2

Best fitness (MSE) versus generation in GA-MLP.

Figure 3 shows the plot of the network output and the desired network output for test data set. In this figure, the desired output (or dead oil viscosity) is plotted in a solid line, and the corresponding network output is drawn in a dash line.

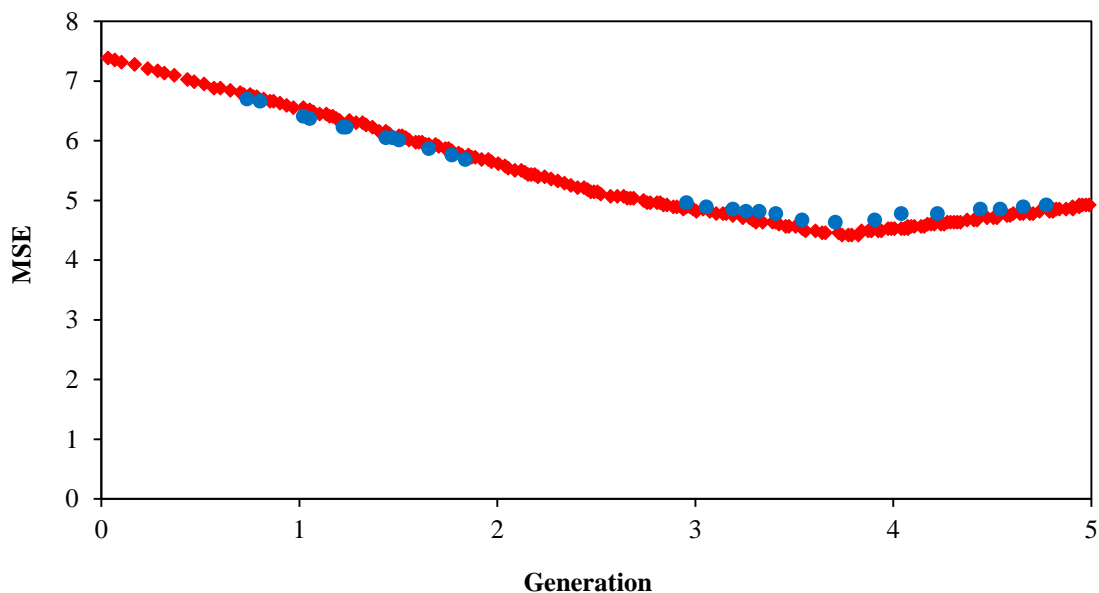


Figure 3

Desired output and actual GA-MLP network output.

Figure 3 displays the actual dead oil viscosity which is measured in the experiments and is never seen by the network during the genetic training in comparison with the network's estimation/prediction for each sample. The results tabulated in the Table 2 reveal that the correlation coefficient between the desired parameters and GA-MLP output is 0.997475672. Accordingly, the genetically trained network

is able to predict/estimate dead oil viscosity amounts in good agreement with the actual measurements in the experiments. The capabilities of the optimized neural networks in pattern recognition were also established.

Table 2
Training and cross-validation error obtained from the trained GA-MLP network.

Optimization summary	Best fitness	Average fitness
Generation number	24	21
Minimum MSE	5.94747×10^{-07}	5.5981×10^{-06}
Final MSE	5.94747×10^{-07}	0.005515255

3.2. Genetic optimization of radial basis function method

The GA-RBF algorithm provides a complete framework for building models based on available data since, apart from providing a mathematical expression, it selects the appropriate factors that are going to be used as the inputs to the model. Therefore, it considers the problem as a small or medium scale variable selection problem (Kudo et al., 2000). The two objectives (the selection of proper regressors and the minimization of the prediction error) are combined into a single objective function. The methodology uses a specially designed GA as a search strategy, which employs a hybrid coding of genes. More precisely, each potential input variable is coded by a binary gene denoting whether the variable is present in the model (the gene has the value of 1) or not (the gene has the value of 0). An additional integer gene in each chromosome corresponds to the number of fuzzy sets that are defined in the domain of each variable, which is a parameter used by the fuzzy means algorithm. Thus, the length of each chromosome is equal to the number of candidates for becoming input variables plus one.

The RBF neural network model corresponding to a particular chromosome is constructed by using only the input variables in the binary genes of the chromosome. Then, the fuzzy means algorithm is applied, where the number of fuzzy sets in each input direction is set equal to the content of the integer gene of the chromosome. A new population is generated by selecting individuals from the old population based on the previously calculated fitness of chromosomes. The crossover and mutation genetic operators are applied to the new population. The crossover operator is employed to exchange genes between two chromosomes. For the specific methodology, a one-point crossover scheme is utilized, where they exchange strings of genes after some pairs of chromosomes are randomly selected based on the probability of crossover. During the crossover operation, the last integer gene is treated in the same manner as the binary genes, and it is exchanged as well.

However, the different nature of the last gene necessitates a special treatment during the mutation operation. Thus, uniform flip bit mutation is applied to the binary genes with a probability equal to p_{um} , but nonuniform mutation with a probability of p_{num} is used if the gene selected for mutation is the integer gene representing the number of fuzzy sets (Michalewicz et al., 1996). Nonuniform mutation is preferred over uniform mutation for altering the number of fuzzy sets since it initially searches the space uniformly, but, as the algorithm proceeds, it performs a more local search. The algorithm will be performed continuously unless either it has reached the maximum number of iterations or the fitness value has not been improved for the last L iterations. The final outcome of the algorithm is the chromosome that produces the best fitness value during the entire procedure. This chromosome defines the optimal subset of the input variables and the produced RBF model based on the above description. The following is the tuning parameters of the GA-RBF algorithm that must be defined by the user:

- the size of the population, which represents the total number of chromosomes;

- the total number of generations;
- the number of consecutive generations for which the objective value is not improved;
- the probability of crossover;
- the probabilities of uniform mutation and nonuniform mutation.

The data of 20 experiments were divided into three sets of different lengths: the first set (60% of the data set for training) used for model building and parameter estimation; the second set (20% of the data set for cross-validation); and the remaining 20% of the data set used to test the model. The target was to build a model that can predict the result of this experiment in other conditions which cannot be achieved in our experiment. The GA-RBF and GA-MLP methods were programmed in the MATLAB 7.7 environment and NeuroSolutions Release 5.0 software separately, and their results were in good agreement. GA-RBF was used to select the optimal subset of variables by employing the parameter values summarized in Table 3. The runtime of the program was less than 15 min using a Pentium IV 1.86 MHz processor, which shows that the method is not computationally demanding; this is due the fact that the algorithm needs to run only once.

Table 3
Parameters for the variable selection algorithm.

Parameter	Value
Population size	50
Maximum number of generations	27-40
Maximum number of consecutive generations for which no improvement is observed	100
Crossover probability	50
Mutation probability	0.9
Cluster centers	0.15

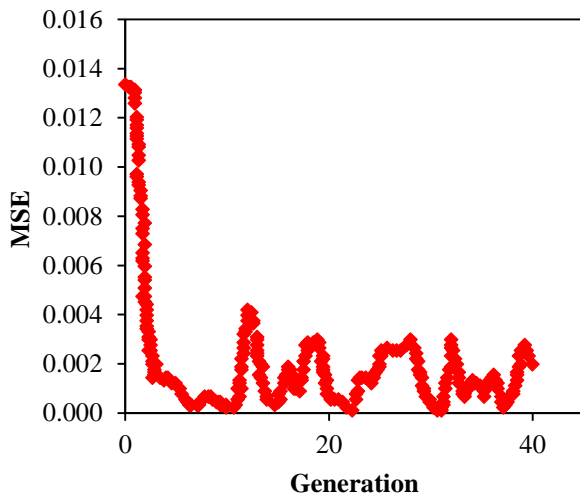
The same results obtained by GA-RBF method are listed in Tables 4 and 5 and Figures 4-6.

Table 4
Training and cross-validation error obtained from the trained GA-RBF network.

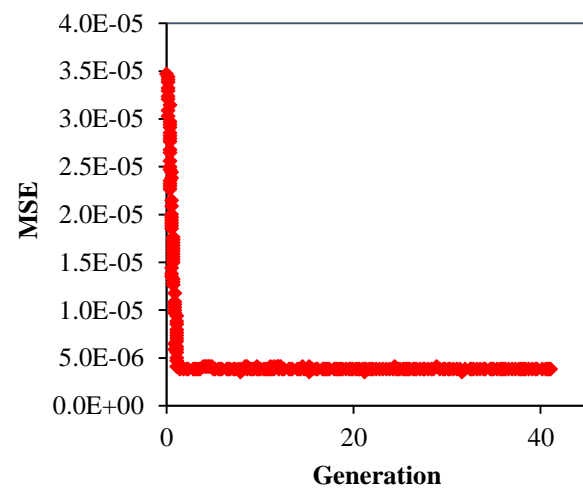
Optimization summary	Best fitness	Average fitness
Generation number	2	23
Minimum MSE	3.80505×10^{-06}	6.42298×10^{-05}
Final MSE	3.80505×10^{-06}	0.00199039

Table 5
Performance of GA-RBF model for the test data set.

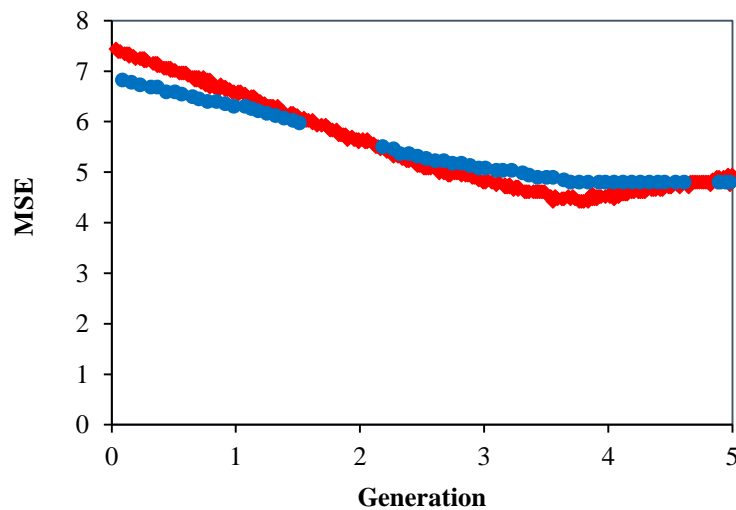
Performance	Viscosity
MSE	0.108100505
NMSE	0.090848089
MAE	0.283118159
Min absolute error	0.117146862
Max absolute error	0.564296578
<i>R</i>	0.985606095

**Figure 4**

Average fitness (MSE) versus generation in GA-RBF.

**Figure 5**

Best fitness (MSE) versus generation in GA-RBF.

**Figure 6**

Desired output and actual GA-RBF network output.

4. Conclusions

In this paper, we presented a complete framework for the development of PVT evaluation model. An attempt was made to develop a methodology for designing the neural network architecture using genetic algorithm. Genetic algorithm was used to determine the number of neurons in the hidden layers, the momentum, and the learning rates in order to minimize the time and the effort required to find the optimal architecture. The methodology is particularly useful for the prediction of PVT evaluation result. The GA-RBF and GA-MLP methods combine two advanced artificial intelligence technologies, namely the RBF and MLP neural networks architecture, and a specially designed genetic algorithm to select the appropriate explanatory variables.

Comparing the prediction performance efficiency of the GA-MLP with that of GA-RBF showed that GA-MLP model significantly outperformed the GA-RBF model. On the other hand, GA was found to be a good alternative to the trial-and-error approach to quickly and efficiently determine the optimal

ANN architecture and internal parameters. The performance of the networks with respect to the predictions made on the test data sets showed that the neural network model incorporating a GA was able to sufficiently estimate the PVT evaluation with a high correlation coefficient.

Acknowledgments

The authors would like to thank National Iranian Oil Field Company (NIOC) and National Iranian South Oil Field Company (NISOC) for cooperating in conducting this project, for giving permission to publish the related data, and for the financial support of this project.

Nomenclature

ANN	: Artificial neural network
API	: American Petroleum Institute
BPN	: Back propagation network
GA	: Genetic algorithm
MAE	: Mean absolute error
MLP	: Multi-layer perceptron
MSE	: Mean squared error
NMSE	: Normalized mean squared error
PVT	: Pressure-volume-temperature
RBF	: Radial basis function

References

- Abedini, R., Esfandyari, M., Nezhadmoghadam, A., and Adib, H., Evaluation of Crude Oil Property Using Intelligence Tool: Fuzzy Model Approach, *Chemical Engineering Research Bulletin*, Vol. 5, No. 1, p. 30-33, 2011.
- Abedini, R., Esfandyari, M., Nezhadmoghadam, A., and Rahmanian, B., The Prediction of Undersaturated Crude Oil Viscosity: an Artificial Neural Network and fuzzy model approach, *Petroleum Science and Technology*, Vol. 30, No. 19, p. 2008-2021, 2012.
- Al-Marhoun, M. and Osman, E., Using Artificial Neural Networks to Develop New PVT Correlations for Saudi Crude Oils, Abu Dhabi International Petroleum Exhibition and Conference, Society of Petroleum Engineers, 2002.
- Ayoub, M.A., Raja, A.I., and Almarhoun, M., Evaluation of Below Bubble Point Viscosity Correlations and Construction of a New Neural Network Model, Asia Pacific Oil and Gas Conference and Exhibition, Society of Petroleum Engineers, 2007.
- Barhen, J., Protopopescu, V., and Reister, D., TRUST: a Deterministic Algorithm for Global Optimization, *Science*, Vol. 276, No. 5315, p. 1094-1097, 1997.
- Basso, P., Iterative Methods for the Localization of the Global Maximum, *SIAM Journal on Numerical Analysis*, Vol. 19, No. 4, p. 781-792, 1982.
- Caprani, O., Godthaab, B., and Madsen, K., Use of a Real-valued Local Minimum in Parallel Interval Global Optimization, *Interval Computations*, Vol. 2, p. 71-82, 1993.

- Esfandyari, M., Fanaei, M. A., Gheshlaghi, R., and Mahdavi, M. A., Neural Network and Neuro-fuzzy Modeling to Investigate the Power Density and Columbic Efficiency of Microbial Fuel Cell, *Journal of the Taiwan Institute of Chemical Engineers*, Vol. 58, p. 84-91, 2016.
- Gharbi, R. B., Elsharkawy, A. M., and Karkoub, M., Universal Neural-network-based Model for Estimating the PVT Properties of Crude Oil Systems, *Energy and Fuels*, Vol. 13, No. 2, p. 454-458, 1999.
- González, A., Barrufet, M. A., and Startzman, R., Improved Neural-network Model Predicts Dew Point Pressure of Retrograde Gases, *Journal of Petroleum Science and Engineering*, Vol. 37, No. 3, p. 183-194, 2003.
- Khandekar, S., Joshi, Y. M., and Mehta, B., Thermal Performance of Closed two-phase Thermosyphon Using Nanofluids, *International Journal of Thermal Sciences*, Vol. 47, No. 6, p. 659-667, 2008.
- McDiarmid, C., *Simulated Annealing and Boltzmann Machines a Stochastic Approach to Combinatorial Optimization and Neural Computing*, Wiley Online Library, 1991.
- Michalewicz Z., Dasgupta D., Le Riche R.G., and Schoenauer M., Evolutionary Algorithms for Constrained Engineering Problems, *Computers and Industrial Engineering*, Vol. 30, No. 4, p. 851-870, 1996.
- Saemi, M., Ahmadi, M., and Varjani, A.Y., Design of Neural Networks Using Genetic Algorithm for the Permeability Estimation of the Reservoir, *Journal of Petroleum Science and Engineering*, Vol. 59, No. 1, p. 97-105, 2007.
- Samui, P., Application of Least Square Support Vector Machine (LSSVM) for Determination of Evaporation Losses in Reservoirs, *Engineering*, Vol. 3, No. 4, p. 431-434, 2011.
- Sargolzaei, J., Hedayati Moghaddam, A., Nouri, A., and Shayegan, J., Modeling the Removal of Phenol Dyes Using a Photocatalytic Reactor with SnO₂/Fe₃O₄ Nanoparticles by Intelligent System, *Journal of Dispersion Science and Technology*, Vol. 36, No. 4, p. 540-548, 2015.
- Sargolzaei, J. and Moghaddam, A. H., Predicting the Yield of Pomegranate Oil from Supercritical Extraction Using Artificial Neural Networks and an Adaptive-network-based Fuzzy Inference System, *Frontiers of Chemical Science and Engineering*, Vol. 7, No. 3, p. 357-365, 2013.
- Varotsis, N., Gaganis, V., Nighswander, J., and Guieze, P., A Novel Non-iterative Method for the Prediction of the PVT Behavior of Reservoir Fluids, *SPE Annual Technical Conference*, 1999.