Turkish Journal of Engineering

*Turkish Journal of Engineering (TUJE)*
*Vol. 2, Issue 3, pp. 119-124, September 2018*
*ISSN 2587-1366, Turkey*
*DOI: 10.31127/tuje.408976*
*Research Article*

# PERFORMANCE COMPARISON OF ANFIS, ANN, SVR, CART AND MLR TECHNIQUES FOR GEOMETRY OPTIMIZATION OF CARBON NANOTUBES USING CASTEP

Mehmet Acı [1], Çiğdem İnan Acı *[2] and Mutlu Avcı [3]

[1] Mersin University, Faculty of Engineering, Department of Computer Engineering, Mersin, Turkey
ORCID ID 0000-0002-7245-8673
maci@mersin.edu.tr

[2] Mersin University, Faculty of Engineering, Department of Computer Engineering, Mersin, Turkey
ORCID ID 0000-0002-0028-9890
caci@mersin.edu.tr

[3] Çukurova University, Faculty of Engineering, Department of Biomedical Engineering, Adana, Turkey
ORCID ID 0000-0002-4412-4764
mavci@cu.edu.tr

* Corresponding Author

Received: 22/03/2018 Accepted: 10/04/2018

**ABSTRACT**
Density Functional Theory (DFT) calculations used in the Carbon Nanotubes (CNT) design take a very long time even in the simulation environment as it is well known in literature. In this study, calculation time of DFT for geometry optimization of CNT is reduced from days to minutes using seven artificial intelligence-based and one statistical-based methods and the results are compared. The best results are achieved from ANFIS and ANN based models and these models can be used instead of CNT simulation software with high accuracy.

***Keywords:*** *Geometry Optimization, Cnt, Dft, Artificial Intelligence*

## 1. INTRODUCTION

Density functional theory (DFT) (Kohn and Sham, 1965) is the most successful method that calculates atomic coordinates faster than other mathematical approaches and it also reaches more accurate results. DFT uses ground state energy formula which is developed by Kohn and Sham (Eq. (1)).

$$E = \sum_1^v \epsilon_j - \frac{1}{2}\int \frac{n(r)n(r')}{|r-r'|}\mathrm{d}r\mathrm{d}r' - \int v_{xc}(r)n(r)\mathrm{d}r + E_{xc}[n(r)] \quad (1)$$

where the $\epsilon_j$ and $n$ are the self-consistent quantities, $V_{xc}$ is the exchange correlation potential energy, $E_{xc}$ is the exchange correlation energy, and $n(r)$ is the electron density. However, the elapsed time for calculation of high number of atoms may even take several days due to calculation capability limits of workstation computers. On the other hand, users need to use more powerful workstations and parallel computer grids which are too expensive to buy easily for reducing the calculation time. In literature, many researchers remark this calculation time problem in their studies. General view of the researchers who studied on geometry optimization using DFT can be summarized as “DFT calculations are time consuming”.

Many researchers study on CNT to obtain perfect CNTs and widen their application areas: Some of the studies are focused on geometry optimizations of the CNTs (Kanamitsu and Saito, 2002; Kürti *et al.*, 2003; Moradian *et al.*, 2008, 2009; Yagi *et al.*, 2004). Also many of the researchers that study on CNT calculations, incorporate artificial intelligence methods into their works (Abo-Elhadeed, 2012; Akbari *et al.*, 2014; Cheng *et al.*, 2015; Ensafi *et al.*, 2010; Hassanzadeh *et al.*, 2015; Hayati *et al.*, 2010; Rahimi-Nasrabadi *et al.*, 2015; Salehi *et al.*, 2016; Shanbedi *et al.*, 2013). Nowadays, this incorporation trend has been increasing (Acı and Avcı, 2016).

The motivation of this research is to reduce the calculation time for atomic coordinates from days to minutes. It is known that the current mathematical methods cannot reduce the calculation time up to this level. In this work, the problem is investigated in another perspective. Instead of calculation; the atomic coordinates are predicted as accurately as possible in a short time. These predicted atomic coordinates can be used as initial coordinates for the simulation software as depicted in Fig. 1. Thus, the exact atomic coordinates can be calculated within minutes or hours instead of days utilizing the proposed approach. In some researches predicted atomic coordinates may be enough in accuracy. In that case, the predicted coordinates may provide the fastest solution.

The main objective aimed in this work is to develop prediction models using regression-based supervised artificial intelligence techniques such as Adaptive-Network Based Fuzzy Inference System (Jang, 1993) (ANFIS), four types of Artificial Neural Network (Gupta, 2013) (ANN) (i.e. Feed Forward Neural Network (FFNN), Function Fitting Neural Network (FITNET), Cascade-Forward Neural Network (CFNN) and Generalized Regression Neural Network (GRNN)), Classification and Regression Tree (Lawrence and Wright, 2001) (CART) and Support Vector Regression (Smola and Vapnik, 1997) (SVR) to estimate the atomic coordinates of CNTs. One statistical method (i.e. Multiple Linear Regression (Eberly, 2007) (MLR)) is used to compare results with artificial intelligence based methods.

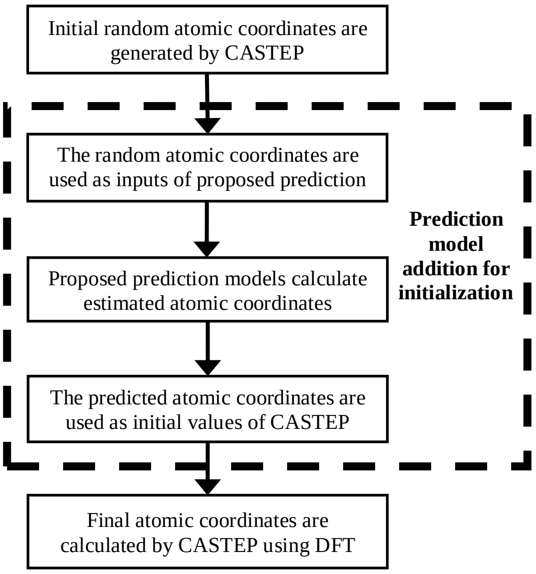
Initial random atomic coordinates are generated by CASTEP
The random atomic coordinates are used as inputs of proposed prediction
Proposed prediction models calculate estimated atomic coordinates
The predicted atomic coordinates are used as initial values of CASTEP
Final atomic coordinates are calculated by CASTEP using DFT
Prediction model addition for initialization

Fig. 1. Steps of the two-staged CASTEP simulation

## 2. MATERIALS AND METHODS

### 2.1. Dataset

Two distinct datasets named as input dataset and output dataset are prepared to be used in prediction models. Initial atomic coordinates u, v, w and the pair of integers (n, m) used for specifying the chiral vector are the 5 parameters of input dataset. On the other hand, calculated atomic coordinates u’, v’, w’ are the 3 parameters of output dataset. The dataset consisting of 10,721 data samples firstly is divided randomly into training, validating and testing data randomly, in which there are 70–15–15% training, validating and testing sets respectively. A summary of the descriptive statistics for the dataset is given in Table 1. These statistics include minimum, maximum, mean and standard deviation values of the dataset.

Table 1. Descriptive statistics for the dataset

| | **Inputs** | | | | | **Outputs** | | |
|---|---|---|---|---|---|---|---|---|
| | **u** | **v** | **w** | **m** | **n** | **u’** | **v’** | **w’** |
| **Minimum** | 0.0451 | 0.0451 | 0 | 2 | 1 | 0.0385 | 0.0389 | 0 |
| **Maximum** | 0.9548 | 0.9548 | 0.9999 | 12 | 6 | 0.9614 | 0.9610 | 0.9996 |
| **Mean** | 0.5000 | 0.5000 | 0.4994 | 8.2252 | 0.3378 | 0.5000 | 0.4999 | 0.4993 |
| **Standard Deviation** | 0.2900 | 0.2900 | 0.2900 | 2.1400 | 1.6800 | 0.2909 | 0.2910 | 0.2884 |

The datasets used in this study are generated with CASTEP (2016) using CNT geometry optimization. Different chiral vectors are used for each CNT simulation. The atom type is selected as carbon, bond length is used as 1.42 Å (default value), and then the nanotube is built by CASTEP.

CASTEP uses a parameter named as elec_energy_tol (electrical energy tolerance) which represents that the change in the total energy from one iteration to the next remains below some tolerance value per atom for a few self-consistent field steps to finalize the computation. This parameter also determines the calculation level of inputs and outputs. The default value of the parameter is 1x$10^{-5}$ eV per atom and is usually suitable (CASTEP, 2016).

Initial coordinates of all carbon atoms are generated randomly. The number of simulated atoms ranges from 32 to 588. The calculation time according to these calculations are approximately varies from 10 minutes to 5 days. These calculation times will take weeks and months for higher atom numbers. All calculations are run on a workstation that has a 2.0 Ghz power on 2 Xeon processors with 4 cores and 8 GB of RAM, using all cores under Linux operating system. The calculation time according to these calculations are approximately varies from 10 minutes to 5 days as given in Table 2 in seconds. These calculation times will take weeks and months for higher atom numbers.

Table 2. The calculation time for CNT simulations in seconds

| ID of CNT | Number of Atoms | Calculation Time (Seconds) |
|---|---|---|
| 1 | 28 | 1050.14 |
| 2 | 52 | 1256.40 |
| 3 | 76 | 2254.18 |
| 4 | 84 | 2816.61 |
| 5 | 56 | 834.04 |
| 6 | 148 | 18496.07 |
| 7 | 124 | 33280.80 |
| 8 | 156 | 9604.28 |
| 9 | 196 | 59998.78 |
| 10 | 244 | 19389.07 |
| 11 | 172 | 64102.59 |
| 12 | 104 | 7207.22 |
| 13 | 84 | 6842.67 |
| 14 | 152 | 7715.19 |
| 15 | 364 | 78421.39 |
| 16 | 228 | 13513.07 |
| 17 | 268 | 27799.53 |
| 18 | 316 | 35982.11 |
| 19 | 372 | 94767.02 |
| 20 | 436 | 266027.27 |
| 21 | 508 | 233190.94 |
| 22 | 292 | 27792.55 |
| 23 | 168 | 13292.91 |
| 24 | 388 | 106542.26 |
| 25 | 112 | 6838.91 |
| 26 | 516 | 201214.72 |
| 27 | 296 | 34136.32 |
| 28 | 364 | 103759.38 |
| 29 | 412 | 95275.60 |
| 30 | 156 | 9558.96 |
| 31 | 532 | 427956.63 |
| 32 | 228 | 27392.57 |
| 33 | 444 | 159893.39 |
| 34 | 248 | 74997.74 |
| 35 | 312 | 40999.97 |
| 36 | 140 | 10014.92 |
| 37 | 392 | 83173.05 |
| 38 | 588 | 397369.36 |
| 39 | 344 | 89653.65 |
| 40 | 252 | 26127.05 |
| 41 | 208 | 23123.48 |
| 42 | 168 | 11826.42 |

## 2.2. Prediction Models

ANFIS, FFNN, FITNET, CFNN, GRNN, CART, SVR and MLR models were trained and tested with the dataset of atomic coordinates for CNTs.

A Sugeno-type (Takagi and Sugeno, 1985) Fuzzy Inference System (FIS) is used by MATLAB's implementation of ANFIS. An adaptive neural network technique is used to train the Sugeno-type FIS parameters. Inputs are mapped through input membership functions and associated parameters by ANFIS. Then through output membership functions and associated parameters to outputs, can be used to interpret the input/output map. The parameters associated with the membership functions change through the learning process. A gradient vector facilitates the computation of these parameters. A measure of how well the FIS is modeling the input/output data for a given set of parameters is provided by this gradient vector. When the gradient vector is obtained, any of several optimization routines can be applied in order to adjust the parameters to reduce some error measure. The sum of the squared difference between actual and desired outputs usually defines this error measure. Either back propagation or a combination of least squares estimation and backpropagation for membership function parameter estimation is used by ANFIS (MATLAB, 2016).

A matrix of training data forms the input training data and last column is the target output data. Test input data is used also in the same way. While the fuzzy toolbox provides many membership function types, Generalized Bell-Shaped Membership Function (gbellmf), Gaussian Curve Membership Function (gaussmf) and Pi-Shaped Membership Function (pimf) performed the best for the prediction of atomic coordinates. After the five inputs enter the ANFIS then they processed for the u' coordinate prediction. Similar FIS models are also designed for prediction of v' and w' coordinates.

FFNN, FITNET and CFNN models have 3 layers (input, hidden and output) and the input and output layers have 5 and 3 neurons respectively. The hidden layers of FITNET and CFNN have 10 neurons and that of FFNN has 20 neurons. A log-sigmoid activation function (LOGSIG) is used in FFNN model and the hyperbolic tangent sigmoid activation function (TANSIG) is used in FITNET and CFNN models in the hidden layer. A pure-linear activation function is used in all models in the output layer and Levenberg–Marquardt algorithm (Moré, 1978) is utilized for training the networks. Weights and biases were randomly initialized. The other important parameters of the FFNN and FITNET models are the number of epochs (1000), the learning rate (0.02), and momentum (0.5). The network parameters have been optimized by try-and-error (i.e. after testing the neural network with several different configurations and observing that these numbers yield the lowest error rates for prediction) in order to reach the accurate results. The parameter σ of GRNN model (also called 'spread' in MATLAB) determines the generalization capability of the GRNN. The best spread parameter is adopted as 2 through this work.

CART analysis was performed using "classregtree" function which specifies some optional parameters in MATLAB. The "prune" option computes the full tree and prunes the subtrees. The "minparent" option splits impure nodes which have k or more observations to be split for given number of k. The "qetoler" option defines tolerance on quadratic error per node for CART. Splitting nodes stops when quadratic error per node drops below qetoler*qed, where qed is the quadratic error for the entire data computed before the decision tree is grown. The "mergeleaves" option merges leaves that originate from the same parent node and give the sum of risk values greater or equal to the risk associated with the parent node. The values of the parameters were chosen as prune=on, minparent=50 (default is 10), qetoler=1E-7 (default is 1E-6), mergeleaves=on. All of the parameter values were decided by trial-and-error.

SVR model was designed by running "fitrsvm" function in MATLAB. Several combinations have been tried and Radial Basis Function was chosen as the kernel for performance comparison.

MLR prediction model is designed with the purpose of comparing Machine Learning methods with a statistical regression method. The model is coded in Statistics and Machine Learning Toolbox of MATLAB using "fitlm" function. Model specification is specified as "linear" and model specification equations are set for each coordinate for the predictions of coordinates such as 'u' ~ u+v+w+m+n' is set for u'; 'v' ~ u+v+w+m+n' is set for v'; and 'w' ~ u+v+w+m+n' is set for w'.

## 3. RESULTS AND DISCUSSION

Mean Squared Error (MSE), Mean Absolute Error (MAE), Standard Error of Estimation (SEE) and Multiple Correlation Coefficient (R) are calculated to evaluate the performance of prediction models. Summaries of mathematic equations of these performance measures are given in Eq. (2), Eq. (3), Eq. (4) and Eq. (5) respectively (Witten and Frank, 2005).

$$MSE = \frac{1}{n}\left[\sum_{i=1}^{n}(O_i - P_i)^2\right] \quad (2)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|O_i - P_i| \quad (3)$$

$$SEE = \sqrt{1 - \frac{\sum_{i=1}^{n}(O_i - P_i)^2}{n}} \quad (4)$$

$$R = \sqrt{1 - \frac{\sum_{i=1}^{n}(O_i - P_i)^2}{\sum_{i=1}^{n}(O_i - O_m)^2}} \quad (5)$$

where $n$ is the number of data points used for testing, $P_i$ is the predicted value, $O_i$ is the observed value and $O_m$ is the average of the observed values. MATLAB (R2015b 64 bit) (MATLAB, 2016) was utilized for designing proposed models and obtaining performance measures.

Table 3, Table 4 and Table 5 summarize the performance results of u', v' and w' coordinates prediction using ANFIS, FFNN, FITNET, CFNN, GRNN, SVR, CART and MLR models respectively.

Table 3. Performance results of u' coordinate prediction (the best results are outlined in bold).

| | **Performance metrics** | | | |
|---|---|---|---|---|
| **Models** | **MSE** | **MAE** | **SEE** | **R** |
| ANFIS | 1.077E-05 | 2.224E-03 | **3.282E-03** | **9.999E-01** |
| FFNN | 9.879E-06 | 2.084E-03 | 1.000E+00 | **9.999E-01** |
| FITNET | 1.130E-05 | 2.143E-03 | 1.000E+00 | **9.999E-01** |
| CFNN | **9.403E-06** | **2.025E-03** | 1.000E+00 | **9.999E-01** |
| GRNN | 7.911E-03 | 2.532E-01 | 1.000E+00 | 9.371E-01 |
| SVR | 5.102E-04 | 1.957E-02 | 9.998E-01 | 9.970E-01 |
| CART | 2.471E-05 | 3.868E-03 | 1.000E+00 | **9.999E-01** |
| MLR | 1.525E-01 | 2.482E+00 | 3.000E+00 | 5.999E-01 |

Table 4. Performance results of v' coordinate prediction (the best results are outlined in bold).

| | **Performance Metrics** | | | |
|---|---|---|---|---|
| **Models** | **MSE** | **MAE** | **SEE** | **R** |
| ANFIS | 1.101E-05 | 2.294E-03 | **3.318E-03** | **9.999E-01** |
| FFNN | **7.307E-06** | **1.870E-03** | 1.000E+00 | **9.999E-01** |
| FITNET | 8.708E-06 | 1.997E-03 | 1.000E+00 | **9.999E-01** |
| CFNN | 8.537E-06 | 1.965E-03 | 1.000E+00 | **9.999E-01** |
| GRNN | 7.995E-03 | 2.554E-01 | 1.000E+00 | 9.298E-01 |
| SVR | 4.985E-04 | 1.940E-02 | 9.998E-01 | 9.971E-01 |
| CART | 2.484E-05 | 3.835E-03 | 1.000E+00 | 9.998E-01 |
| MLR | 1.298E-01 | 2.300E+00 | 3.000E+00 | 6.000E-01 |

Table 5. Performance results of w' coordinate prediction (the best results are outlined in bold).

| | **Performance Metrics** | | | |
|---|---|---|---|---|
| **Models** | **MSE** | **MAE** | **SEE** | **R** |
| ANFIS | **4.962E-08** | 1.534E-04 | **2.228E-04** | **1.000E+00** |
| FFNN | 5.501E-08 | 1.531E-04 | 1.000E+00 | **1.000E+00** |
| FITNET | 5.554E-08 | 1.523E-04 | 1.000E+00 | **1.000E+00** |
| CFNN | 5.418E-08 | **1.498E-04** | 1.000E+00 | **1.000E+00** |
| GRNN | 8.088E-03 | 2.466E-01 | 9.587E-01 | 9.313E-01 |
| SVR | 4.321E-04 | 1.832E-02 | 9.997E-01 | 9.983E-01 |
| CART | 2.140E-05 | 3.021E-03 | 1.000E+00 | 9.999E-01 |
| MLR | 4.865E-02 | 1.476E-01 | 3.000E+00 | 6.000E-01 |

All models performed good levels of successes when we look at all results with a general view. However, the results of ANFIS, FFNN, FITNET and CFNN models have a superiority over other models. This implies that ANN based models can closely estimate atomic coordinates of CNTs. CART and SVR models produced average performance results comparing to other models. The results can be analyzed in detail as follows:

- CFNN model has the best performance values for u' and w' coordinates prediction by means of MSE and MAE. However, MSE and R results of ANFIS, FFNN and FITNET models are very close to CFNN model.
- The best SEE results are achieved by ANFIS model for all coordinates.
- R value prediction results varies in a very small range (from 9.298E-01 to 1.000E+00) for all models except MLR based model.
- MLR based model yielded the worst performance results for all coordinates.
- The estimation results with the highest accuracy are yielded for w' coordinate (MSE results of the first four models are almost zero).

The results obtained from this study can be used in two ways: i) The predicted atomic coordinates can be used in physical calculations without using a simulation software, ii) The estimated results can be used as an initial value of simulation software for reducing duration of the atomic coordinate calculation seriously.

## 4. CONCLUSION

As a result of this work, the effectiveness of artificial intelligence based solutions, which estimate atomic coordinates that can be integrated into software, has been observed and the results that have significantly shortened the simulation processes in the field of nanotechnology have been presented to be integrated into scientific or commercial software.

## ACKNOWLEDGMENTS

We would like to thank Çukurova University Scientific Research Projects Center for supporting this work (Project code: FDK-2015-3170).

## REFERENCES

Abo-Elhadeed, A. F. (2012). "Modeling carbon nanotube transistors using neural networks approach." *Proc., International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD),* pp. 125-128.

Acı, M. and Avcı, M. (2016). "Artificial neural network approach for atomic coordinate prediction of carbon nanotubes." *Applied Physics A*, vol. 7 No. 122, pp. 1-14.

Akbari, E., Buntat, Z., Enzevaee, A., Ebrahimi, M., Yazdavar, A. H. and Yusof, R. (2014). "Analytical modeling and simulation of I–V characteristics in carbon nanotube based gas sensors using ANN and SVR methods." *Chemometrics and Intelligent Laboratory Systems*, Vol. 137, No. 1, pp. 173-180.

CASTEP Biovia Materials. http://accelrys.com/products/collaborative-science/biovia-materials-studio/quantom-catalysis-software.html [Accessed 10 May 2017].

Cheng, W., Cai, C., Luo, Y., Li, Y. and Zhao, C. (2015). "Mechanical properties prediction for carbon nanotubes/epoxy composites by using support vector regression." *Modern Physics Letters B*, Vol. 29, No. 05, pp. 1550016.

Eberly, L. E. (2007). *Multiple linear regression. In: Topics in biostatistics*, Humana Press, Totowa, NJ, USA.

Ensafi, A. A., Taei, M., Khayamian, T. and Hasanpour, F. (2010). "Simultaneous voltammetric determination of enrofloxacin and ciprofloxacin in urine and plasma using multiwall carbon nanotubes modified glassy carbon electrode by least-squares support vector machines." *Analytical Sciences*, Vol. 26, No. 7, pp. 803-808.

Gupta, N. (2013). "Artificial neural network." *Network and Complex Systems*, Vol. 3, No. 1, pp. 24-28.

Hassanzadeh, Z., Kompany-Zareh, M., Ghavami, R., Gholami, S. and Malek-Khatabi, A. (2015). "Combining radial basis function neural network with genetic algorithm to QSPR modeling of adsorption on multi-walled carbon nanotubes surface." *Journal of Molecular Structure*, Vol. 1098, No. 1, pp. 191-198.

Hayati, M., Rezaei, A. and Seifi, M. (2010). "CNT-MOSFET modeling based on artificial neural network: Application to simulation of nanoscale circuits." *Solid-State Electronics*, Vol. 54, No. 1, pp. 52-57.

Jang, J.-S. (1993). "ANFIS: adaptive-network-based fuzzy inference system." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 3, pp. 665-685.

Kanamitsu, K. and Saito, S. (2002). "Geometries, electronic properties, and energetics of isolated single walled carbon nanotubes." *Journal of the Physical Society of Japan*, Vol. 71, No. 2, pp. 483-486.

Kohn, W. and Sham, L. J. (1965). "Self-consistent equations including exchange and correlation effects." *Physical review*, Vol. 140, No. 4A, pp. A1133–A1138.

Kürti, J., Zólyomi, V., Kertesz, M. and Sun, G. (2003). "The geometry and the radial breathing mode of carbon nanotubes: beyond the ideal behaviour." *New Journal of Physics*, Vol. 5, No. 1, pp. 125.

Lawrence, R. L. and Wright, A. (2001). "Rule-based classification systems using classification and regression tree (CART) analysis." *Photogrammetric Engineering and Remote sensing*, Vol. 67, No. 10, pp. 1137-1142.

MATLAB. https://www.mathworks.com/ [Accessed 12 May 2017].

Moradian, R., Behzad, S. and Chegel, R. (2008). "Ab initio density functional theory investigation of structural and electronic properties of silicon carbide nanotube bundles." *Physica B: Condensed Matter*, Vol. 403, No. 19, pp. 3623-3626.

Moradian, R., Behzad, S. and Chegel, R. (2009). "Ab initio density functional theory investigation of Li-intercalated silicon carbide nanotube bundles." *Physics Letters A*, Vol. 373, No. 26, pp. 2260-2266.

Moré J.J. (1978). *The levenberg-marquardt algorithm: implementation and theory. In: Numerical analysis*, Springer, Berlin, Germany.

Rahimi-Nasrabadi, M., Akhoondi, R., Pourmortazavi, S. M. and Ahmadi, F. (2015). "Predicting adsorption of aromatic compounds by carbon nanotubes based on quantitative structure property relationship principles." *Journal of Molecular Structure*, Vol. 1099, No. 1, pp. 510-515.

Salehi, E., Abdi, J. and Aliei, M. H. (2016). "Assessment of Cu (II) adsorption from water on modified membrane adsorbents using LS-SVM intelligent approach." *Journal of Saudi Chemical Society*, Vol. 20, No. 2, pp. 213-219.

Shanbedi, M., Jafari, D., Amiri, A., Heris, S. Z. and Baniadam, M. (2013). "Prediction of temperature performance of a two-phase closed thermosyphon using artificial neural network." *Heat and Mass Transfer*, Vol. 49, No. 1, pp. 65-73.

Smola, A. and Vapnik, V. (1997). "Support vector regression machines." *Advances in Neural Information Processing Systems*, Vol. 9, No. 1, pp. 155-161.

Takagi, T. and Sugeno, M. (1985). "Fuzzy identification of systems and its applications to modeling and control." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-15, No. 1, pp. 116-132.

Witten, I. H. and Frank, E. (2005). *Data mining: practical machine learning tools and techniques*, Morgan Kaufmann, San Francisco, USA.

Yagi, Y., Briere, T. M., Sluiter, M. H., Kumar, V., Farajian, A. A. and Kawazoe, Y. (2004). "Stable geometries and magnetic properties of single-walled carbon nanotubes doped with 3 d transition metals: A first-principles study." *Physical Review B,* Vol. 69, No. 7, pp. 075414.