




Article

Estimating the Heat Capacity of Non-Newtonian Ionanofluid Systems Using ANN, ANFIS, and SGB Tree Algorithms

Reza Daneshfar ¹, Amin Bemani ¹, Masoud Hadipoor ¹, Mohsen Sharifpur ^{2,3,*},
Hafiz Muhammad Ali ⁴, Ibrahim Mahariq ⁵ and Thabet Abdeljawad ^{6,7,8,*}

¹ Department of Petroleum Engineering, Ahwaz Faculty of Petroleum Engineering, Petroleum University of Technology (PUT), Ahwaz P.O. Box 63431, Iran; reza_daneshfar@ut.ac.ir (R.D.); A.bemani@aut.ac.ir (A.B.); masoud.hadipoor@put.ac.ir (M.H.)

² Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam

³ Department of Mechanical and Aeronautical Engineering, University of Pretoria, Pretoria 0002, South Africa

⁴ Mechanical Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia; hafiz.ali@kfupm.edu.sa

⁵ College of Engineering and Technology, American University of the Middle East, Al-Eqaila 54200, Kuwait; ibrahim.maharik@aum.edu.kw

⁶ Department of Mathematics and General Sciences, Prince Sultan University, P.O. Box 66833, Riyadh 11586, Saudi Arabia

⁷ Department of Medical Research, China Medical University, Taichung 40402, Taiwan

⁸ Department of Computer Science and Information Engineering, Asia University, Taichung 40402, Taiwan

* Correspondence: mohsen.sharifpur@up.ac.za or mohsensharifpur@duytan.edu.vn (M.S.); tabdeljawad@psu.edu.sa (T.A.)

Received: 2 July 2020; Accepted: 4 September 2020; Published: 15 September 2020



Abstract: This work investigated the capability of multilayer perceptron artificial neural network (MLP–ANN), stochastic gradient boosting (SGB) tree, radial basis function artificial neural network (RBF–ANN), and adaptive neuro-fuzzy inference system (ANFIS) models to determine the heat capacity (C_p) of ionanofluids in terms of the nanoparticle concentration (x) and the critical temperature (T_c), operational temperature (T), acentric factor (ω), and molecular weight (M_w) of pure ionic liquids (ILs). To this end, a comprehensive database of literature reviews was searched. The results of the SGB model were more satisfactory than the other models. Furthermore, an analysis was done to determine the outlying bad data points. It showed that most of the experimental data points were located in a reliable zone for the development of the model. The mean squared error and R^2 were 0.00249 and 0.987, 0.0132 and 0.9434, 0.0320 and 0.8754, and 0.0201 and 0.9204 for the SGB, MLP–ANN, ANFIS, and RBF–ANN, respectively. According to this study, the ability of SGB for estimating the C_p of ionanofluids was shown to be greater than other models. By eliminating the need for conducting costly and time-consuming experiments, the SGB strategy showed its superiority compared with experimental measurements. Furthermore, the SGB displayed great generalizability because of the stochastic element. Therefore, it can be highly applicable to unseen conditions. Furthermore, it can help chemical engineers and chemists by providing a model with low parameters that yields satisfactory results for estimating the C_p of ionanofluids. Additionally, the sensitivity analysis showed that C_p is directly related to T , M_w , and T_c , and has an inverse relation with ω and x . M_w and T_c had the highest impact and ω had the lowest impact on C_p .

Keywords: ionic liquid; nanofluid; heat capacity; soft computing models; particle swarm optimization

1. Introduction

The efficiency of conventional heat-transfer fluids is augmented by using nanofluids as a novel technique [1–5]. During the history of heat-transfer development, due to the interesting thermal properties of nanofluids, they have been utilized in the production of heat-transfer fluids [6,7].

Ionic liquids (ILs) are a novel group of salt-like materials that are liquid under ambient conditions [8] and have great potential for use in different industries. Although their transport properties have been widely investigated in recent years, investigations of the thermal properties are very limited in the literature [9–12]. Because of their excellent thermal properties, such as high thermal conductivity (TC) and high heat capacity (C_p), they are recognized as an appealing candidate for heat transfer applications [13,14].

In light of previous evidence, it is important to investigate the benefit of utilizing ILs augmented with nanoparticles. By adding only a small amount of nanoparticles into pure ILs, the thermophysical properties of ILs will be enhanced [9,15–18]. A similar idea has been observed in the chemical-enhanced oil recovery methods, which increase the amount of recovered oil by adding a very small amount of nanoparticles to the injected water [19]. This group of nanofluids with ILs as the base fluids that show higher thermophysical properties is called ionanofluids or nanoparticle-enhanced ILs (NEILs) [16].

Paul et al. [20,21] carried out an experimental study and observed an increase in C_p of up to 49% compared with the base ILs when Al_2O_3 nanoparticles were added to pyrrolidinium- and imidazolium-based NEILs. Furthermore, silver nanofluid heat transfer through a tube with twisted tape inserts was tested by Waghole et al. [22]. They concluded that the heat transfer rate was enhanced by dispersing nanoparticles through water. By adding multi-walled carbon nanotubes (MWCNTs) to different ionanofluids and measuring their thermophysical properties, Nieto de Castro et al. concluded that the TC and C_p of ionanofluids were enhanced by $\approx 8\%$ and $\approx 9\%$, respectively [16,23].

What we know about the thermophysical properties of NEILs is largely based upon empirical studies, and these data are controversial regarding their accuracy; furthermore, there is little empirical data available in the literature. Therefore, the determination of the thermophysical properties of NEILs by utilizing theoretical methods is a necessary endeavor.

Computational investigations in particular have become vitally important [24–33], where many researchers have widely implemented intelligent methods based on neuro-fuzzy neural networks to model many engineering processes [6,34–39] and predict the thermophysical properties of different nanofluids [40–42]. In 2013, Salehi et al. [43] used an interesting adaptive neuro-fuzzy inference system (ANFIS) modeling technique in a study that set out to predict the heat transfer coefficient of a nanofluid containing Al_2O_3 under a uniform heat flux condition. Other authors have questioned the usefulness of such an approach. Mehrabi et al. [44] established a genetic algorithm–polynomial neural network and a fuzzy C-means (FCM) based neuro-fuzzy inference system to determine the TC ratio of Al_2O_3 -based nanofluids based on the concentration and size of the nanoparticles, as well as the temperature. Golzar et al. [45] used artificial neural network (ANN) methods and the approximation of general function for the calculation of the thermophysical properties of quaternary ammonium-based ILs in terms of the critical temperature of the ILs and the water content. Soriano et al. determined the refractive index of binary solutions of IL systems based on ANN algorithms [46]. Lashkarblooki et al. calculated the viscosity of ILs by employing boiling temperatures based on ANN algorithms [47]. After that, Hezave et al. employed an ANN to determine the electrical conductivity of a ternary mixture of ILs [48].

Friedman suggested a robust decision tree algorithm named stochastic gradient boosting (SGB), which can be applied in estimation and classification problems. This strategy has shown wide applications as one of the more powerful schemes for different purposes [49–55].

The advantages of boosting (which combines several models) and regression trees are used simultaneously in an SGB tree. A regression tree is created using small incremental changes in the loss function from the previous tree. One of the improvements in the model is the ability to assemble a tree based on a randomly selected data subset. Furthermore, the accuracy of the estimation is maximized and overfitting is minimized via the utilization of a small number of training data

points. Moreover, this algorithm reduces the demands regarding the transformation of the input or the selection of features, which is an advantage when performing in a high dimensional space. Furthermore, this approach has several other beneficial features, such as [49–51,56]:

- SGB-based methods display better prediction performance than competing composite-tree models, including boosting or bagging by applying other approaches, such as AdaBoost.
- SGB-based methods are easy to create.
- SGB-based methods can use a large number of predictor parameters.
- SGB-based methods are developed quickly (100 times faster than neural networks for some problems).
- SGB-based methods are immune to outliers.
- SGB-based methods perform acceptably when solving regression and classification problems.
- SGB-based methods display an equivalent or better predictive ability than neural networks
- Unrelated predictor parameters can be detected automatically such that they do not affect the estimating model.
- Randomization of the elements guards against overfitting in this method.

This current study aimed to investigate these debates through an examination and prediction of the C_p of ionanofluids as a function of the nanoparticle concentration (x) and the operational temperature (T), molecular weight (M_w), acentric factor (ω), and critical temperature (T_c) of pure ILs using a group method of data handling techniques. We evaluated our forecasting model by comparing it with experimental data using three accuracy measurements: R^2 , mean relative error (MRE%), and the mean squared error (MSE).

2. Theory

2.1. Data Preparation

In this work, the predictive capability of four groups of intelligence models was evaluated regarding their ability to estimate the C_p of some NEILs. For this aim, MATLAB 2014 (version 2014, MathWorks, Natick, MA, USA) was used. The characterization of the input and output parameters of the models was the next step after the data preparation. Concerning this, the C_p of the ionanofluids was taken to be the output, while the nanoparticle concentration (x) and the T , M_w , ω , and T_c of pure ILs and were chosen as the five input parameters. The first category, namely, the training set, contained 429 data points. The remaining 142 data points (i.e., 25% of the whole dataset) were employed to test the proficiency of the algorithms. At first, all of the data were normalized in the $[-1, 1]$ interval:

$$D_N = 2 \frac{D - D_{min}}{D_{max} - D_{min}}, \quad (1)$$

where, D , D_N , D_{max} , and D_{min} are the actual, normalized, maximum, and minimum data points, respectively.

2.2. Theory of an ANN

An ANN, which is a kind of intelligence model, is proficient at adapting to changes in the environment, learning from experience, and improving its performance [57–59]. An ANN is composed of neurons. Multi-layer perceptron (MLP) and radial basis function (RBF) networks are two common types of ANNs.

A common MLP structure has the following steps: (1) an input layer, (2) one or more hidden layers, and (3) an output layer. There are some elements (or neurons) in each layer. By using optimization algorithms, the number of neurons used in the hidden layer should be determined [60].

A kind of feed-forward net is sketched using iterative localized basis functions, and the function approximation algorithms are RBF-ANNs. RBF-ANNs and MLP-ANNs differ not only in design

but also in their responses to patterns; with more details, simple design, and very precise response to patterns that are not used for training, RBF-ANNs have an advantage over MLP-ANNs [61]. Because the RBF-ANN training process is much faster than for an MLP-ANN and the structure of an RBF-ANN is simpler, it is an acclaimed alternative to an MLP-ANN [62]. The RBF-ANN structure contains three layers: (1) an input layer, (2) a hidden layer with a non-linear active RBF function, and (3) an output layer. The following equation shows the RBF-ANN output:

$$y_i(x) = \sum_{k=1}^h w_{ki} \varphi(\|x - c_k\|), \tag{2}$$

where x represents an input pattern, $y_i(x)$ denotes the i th output, w_{ki} refers to the connection weight between the k th hidden unit and the i th output, $\| \cdot \|$ represents the Euclidean norm, and c_k represents the center of the k th hidden unit. The Gaussian function was chosen as the RBF (φ), where the Gaussian function is presented below:

$$h(x) = \exp\left(-\frac{(x - c)^2}{r^2}\right). \tag{3}$$

The center (c) and radius (r) are known as Gaussian parameters. The offered MLP-ANN approach utilizes the log-sigmoid transfer function (logsig) and linear transfer function (purelin) in its hidden and output layers, respectively.

2.3. Theory of an ANFIS

An ANFIS usually contains five layers. Jang introduced the method in 1997 [63]. A common ANFIS strategy is training using optimization methods. ANFIS employs the capabilities of fuzzy logic and neural network methods. Algorithms such as particle swarm optimization (PSO) and genetic algorithm (GA) can be employed in the ANFIS to determine the optimal model [64–67].

The structure of an ANFIS is shown in Figure 1, where two inputs (x, y) and one output (f_{out}) are shown. Accordingly, the first layer can be defined for node i as [63]:

$$O_i^1 = \mu_{A_i}(x) \text{ for } i = 1, 2 \text{ or } O_i^1 = \mu_{B_{i-1}}(y) \text{ for } i = 3, 4. \tag{4}$$

By using a membership function (with a range covering the interval (0,1)), all nodes will be parameterized. We used the Gaussian function (given below) as a membership function in the ANFIS approach [63,68]:

$$\mu_A(X) = e^{-\frac{(x-c_i)^2}{2\sigma_i^2}}, \tag{5}$$

where σ and C are the parameters of the Gaussian function.

Some weighted terms and constant nodes are in the second layer [69]:

$$O_i^2 = \omega_i = O_i^1 = \mu_{A_i}(x)\mu_{B_i}(y) \text{ for } i = 1, 2. \tag{6}$$

The average values of the weights are calculated using the following formula in the third layer:

$$\sigma_i^3 = \bar{\omega}_i = \frac{\omega_i}{\omega_1 + \omega_2} \text{ for } i = 1, 2. \tag{7}$$

Each average value of the weight is multiplied by its associated function in the fourth layer:

$$\sigma_i^4 = \bar{\omega}_i f_i = \bar{\omega}_i(p_i x + q_i y + r_i) \text{ for } i = 1, 2, \tag{8}$$

where $p_i, r_i,$ and q_i are the resulting parameters.

Eventually, by summing the previous outputs in the last layer, the output is calculated as follows:

$$\sigma_i^5 = \sum_i \bar{\omega}_i f_i = \frac{\sum_i \omega_i f_i}{\sum_i \omega_i} \tag{9}$$

According to the ten clusters proposed in the ANFIS, 120 membership function (MF) parameters should be optimally determined, where the number 120 reached by multiplying the number of clusters (in this work, 10), the number of parameters of the membership function (in this work, 2), and the sum of the input and output parameters (in this work, 6). The PSO was used to optimize the ANFIS model.

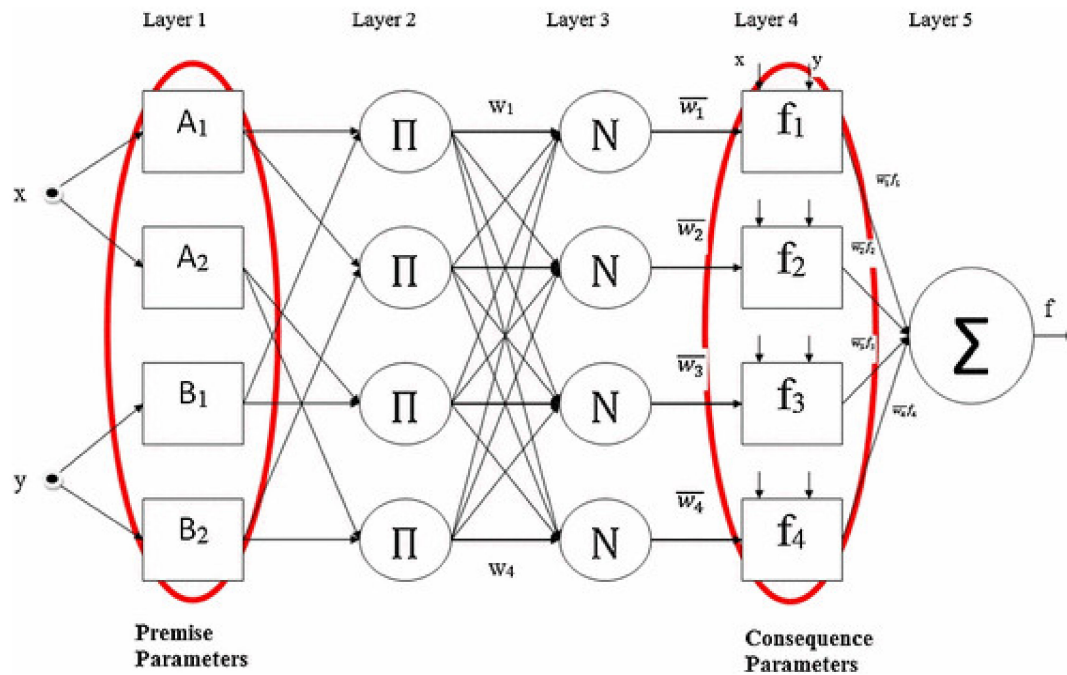


Figure 1. The adaptive neuro-fuzzy inference system (ANFIS) structure.

2.4. Theory of the PSO

PSO is a population-based algorithm that begins with random solutions that are named particles [70]. Eberhart and Kennedy first introduced the concept, which was based on the group behavior of birds and used it for the optimization of continuous non-linear functions [71]. PSO shares many similarities with other evolutionary, population-based algorithms. In an optimization problem, each particle can be considered as a solution. The first stage of the optimization process is the random distribution of particles during the search. The personal best (p_{best}) and global best (g_{best}) values are the optimal solution that has a particle and the optimal solution calculated by the swarm, along with its particle index, respectively. Therefore, the particle velocity (in the next step) can be calculated using p_{best} (cognitive component), g_{best} (social component), and the current particle velocity. The cognitive and social components are both randomly selected [72]. The p th particle is introduced by:

$$X_{pd}^{iter+1} = X_{pd}^{iter} + v_{pd}^{iter+1}, \tag{10}$$

where X_{pi} stands for the i th data point in the D -dimensional space. $G = \{g_1, g_2, \dots, g_D\}$ and $p_p = \{p_{p1}, p_{p2}, \dots, p_{pD}\}_{ss}$ represent the best position among all particles and the acceptable performance for the p th particle, respectively. The particle velocity is represented by $V_p = \{V_{p1}, V_{p2}, \dots, V_{pD}\}$.

The particle position changes due to its speed, and it is updated in each iteration. The particle moves based on its speed, which is calculated as follows:

$$v_{pd}^{iter+1} = \omega v_{pd}^{iter} + C_1 rand(0, 1)(p_{pd} - x_{pd}) + C_2 rand(0, 1)(g_d - X_{pd}). \tag{11}$$

The new position is computed as follows:

$$X_{pd}^{iter+1} = X_{pd}^{iter} + v_{pd}^{iter+1}. \tag{12}$$

The factor ω represents the inertia weight. The positive constants C_1 and C_2 are denoted as learning factors and help particles to move toward a more appropriate space to reach a better solution. PSO updates the inertia weight for each iteration using [73]:

$$\omega_{iter} = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{iter_{max}} \times iter, \tag{13}$$

where ω_{iter} denotes the inertia weight and $iter_{max}$ represents the maximum number of iterations. According to experimental reports, the values of $\omega_{max} = 0.4$ and $\omega_{min} = 0.9$ are apt [72].

2.5. Stochastic Gradient Boosting

Boosting is known as an approach for enhancing the degree of precision in an estimating tool, which is carried out by repeatedly implementing the function as a series to provide a combination of outputs and weights to control the overall error of estimation. This approach has become one of the newest and most powerful learning algorithms, which are applicable in regression and classification problems [74].

A new form of function approximation and statistical learning is Friedman’s SGB approach, which comes from implementing boosting in regression trees [49]. In this method, a series of relatively simple trees is determined, where each successive tree is created based on estimation residuals from the previous tree. The main source of complexity in these trees comes from two child nodes and a root node. In the SGB approach, the optimal data partitioning is calculated based on a step-by-step process; after that, the residuals of each partition are determined. Fitting a three-node tree on those residuals is the next stage for finding a new partition that will reduce the variance of the residual of the data from the aforementioned sequence of trees. According to the typical form of classification for the tree, the classification of observation data is completed, and each tree constructed is summed using this process. The combined result is used to reduce the sensitivity of this algorithm for the suspected datasets. Ensemble learning methods come from machine learning and data mining methods, which combine estimations from some algorithms through bagging, boosting, and related methods. These approaches can be formulated as follows [75]:

$$F(x) = a_0 + \sum_{k=1}^K a_k f_k(x), \tag{14}$$

where K and $f_k(x)$ are the ensemble size and base learner, respectively, for inputs x in the training dataset. In this equation, $F(x)$ gives the ensemble estimation, which is determined via the linear combinations of estimations. Boosting approximates can be determined using an additive expansion form of the previous equation:

$$F(x) = \sum_{k=0}^K \beta_k g(x; a_k), \tag{15}$$

where $g(x; a_k)$ is chosen as a simple function in terms of x and a . A forward-stage approach is employed to forecast the parameters a_k and the expansion coefficient of β_k for the training dataset. The first stage in this process is the initial prediction $F_0(x)$, with the other k values to follow:

$$(\beta_k, a_k) = \operatorname{argmin}_{\beta, a} \sum_{i=1}^N L(y_i, F_{k-1}(x_i) + \beta_k g(x_i; a_k)). \quad (16)$$

L is Huber's function: $L(y, F(x)) = \{[y - F(x)]^2 \text{ for } |y - F(x)| \leq \delta \text{ and } 2\delta|y - F(x)| - \delta^2 \text{ for } |y - F(x)| > \delta$. Furthermore:

$$F_k(X_i) = F_{k-1}(x_i) + \beta_k g(x_i; a_k). \quad (17)$$

Gradient boosting is used to solve Equation (16). First, a least-squares criterion is used to fit the base learning function to the pseudo-residuals:

$$a_k = \operatorname{argmin}_{a, \rho} \sum_{i=1}^N (y_{ik} - \rho g(x_i; a))^2, \quad (18)$$

$$y_{ik} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{k-1}(x)}. \quad (19)$$

The best coefficient is determined as follows:

$$\beta_k = \operatorname{argmin}_{\beta} \sum_{i=1}^N L(y_i, F_{k-1}(x_i) + \beta_k g(x_i; a_k)). \quad (20)$$

According to this result, a hard optimization problem is changed into an easier problem by using one least-square, single-parameter optimization and the general form of a loss criterion. In the SGB algorithm, there is a focus on solving the problem using an observation that is located around the decision boundaries, which is expressed by performing the boosting operation of the model [49]. During the boosting process, it is possible to correct for the observation using an individual tree that is near to another class [76].

3. Data Preparation

To apply the above methods and construct an accurate model, in the first stage, a set of experimental data points for the network training is introduced. A review of empirical studies has been performed, and the C_p values of some NEILs with wide ranges of nanoparticle concentrations and temperatures were found [20,21,77–79]. The total collected experimental dataset (571) was randomly split into two parts: training (75%) and testing (25%) subsets. The first part was required to calculate the model parameters for creating the best network and the testing section was used to validate the predictive power and performance of each model. The characteristics of the studied ILs with added nanoparticles under different temperatures are summarized in Table 1. Furthermore, the M_w , ω , and T_c of pure ILs, which are the model parameters, are given in Table S1 [80,81].

Table 1. The list of ionic liquids (ILs) studied in the literature for the prediction of C_p .

Name	Abbreviation	Temperature (°C)	Nanoparticle Type and Concentration (vol%)	No. of Data Points	Ref.
1-butyl-3-methylimidazolium bis(trifluoromethylsulfonyl)imide	[C4mim][NTf2]	25–345	Al2O3, 0–0.9	154	[20, 21]
1-butyl-2,3-dimethylimidazolium bis(trifluoromethylsulfonyl)imide	[C4mmim][NTf2]	25–345	Al2O3, 0–0.9	132	[77]
N-butyl-N-methylpyrrolidinium bis(trifluoromethylsulfonyl)imide	[C4mpyrr][NTf2]	25–345	Al2O3, 0–0.9	132	[20, 79]
N-butyl-N,N,N-trimethylammonium bis(trifluoromethylsulfonyl)imide	[N4111][NTf2]	25–345	Al2O3, 0–0.9	154	[78]

4. Model Evaluation Parameters

The efficiency of the aforesaid tools was evaluated using statistical variables, namely, the MSE, MRE, R^2 , the standard deviation (STD), and the root of MSE (RMSE) between the empirical and predicted values. These analyses are presented in Equations (22)–(26), respectively:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (X_i^{\text{actual}} - X_i^{\text{predicted}})^2, \quad (21)$$

$$\text{MRE} = \frac{100}{N} \sum_{i=1}^N \left(\frac{X_i^{\text{actual}} - X_i^{\text{predicted}}}{X_i^{\text{actual}}} \right), \quad (22)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (X_i^{\text{actual}} - X_i^{\text{predicted}})^2}{\sum_{i=1}^N (X_i^{\text{actual}} - \overline{X^{\text{actual}}})^2}, \quad (23)$$

$$\text{STD} = \left(\frac{1}{N-1} \sum_{i=1}^N (\text{error} - \overline{\text{error}}) \right)^{0.5}, \quad (24)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N ((X_i^{\text{actual}} - X_i^{\text{predicted}})^2)}. \quad (25)$$

In these relationships, X_i^{actual} , N , and X_i^{actual} show the actual variable, the number of data points, and the output of the network, respectively. Furthermore, $\overline{X^{\text{actual}}}$ is the average of the actual points.

5. Results and Discussion

The offered MLP-ANN approach utilized the log-sigmoid transfer function (logsig) and linear transfer function (purelin) in its hidden and output layers, respectively. The aim was to optimize the hidden layer neurons. This was done using trial and error. Twelve neurons were optimally available in the hidden layer of the MLP-ANN tool. Figure 2 illustrates the MLP-ANN model's performance when using the training data over several iterations. The MLP-ANN was trained using the Levenberg-Marquardt (LM) algorithm. Table S2 shows the optimal values of the MLP-ANN structure. Moreover, in the hidden layers of the RBF-ANN model, the RBF was used. Based on previous results, for the ANN, the number of hidden layer neurons is chosen to be less than one-tenth of the total number of training data points [82]. Ergo, it was assumed that the number of hidden layers of this algorithm was one-tenth of the total number of data points used for training. Figure 3 shows the LM algorithm based on the MSE of various iterations for the RBF-ANN. In Figure S1, the obtained membership functions are shown for the ANFIS model. This figure emphasizes that all of the data (output and input) were normalized in the range $[-1, 1]$. Figure 4 demonstrates the RMSE of the actual

and estimated C_p values for the training data. The highest number of iterations was 1000, and the best RMSE was 0.17416.

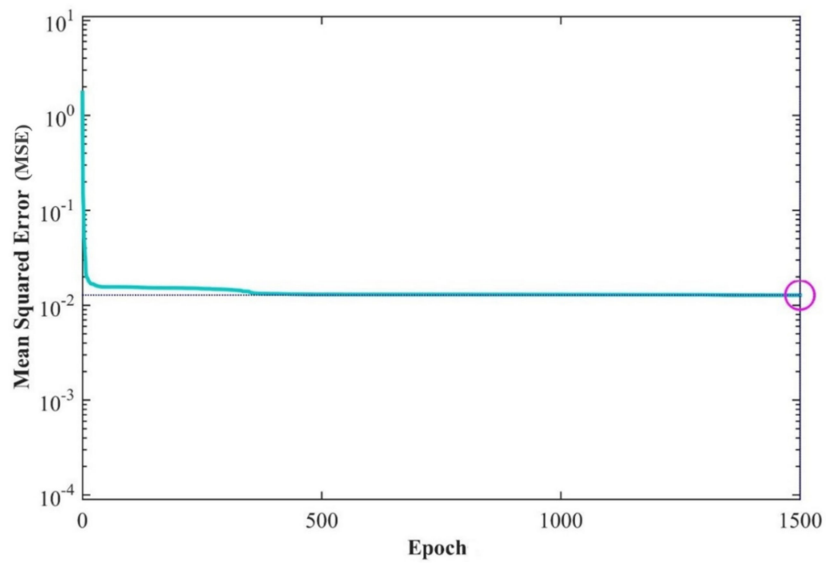


Figure 2. The performance of the Levenberg–Marquardt (LM) algorithm according to the mean squared error (MSE) of different iterations for the multilayer perceptron–artificial neural network (MLP–ANN).

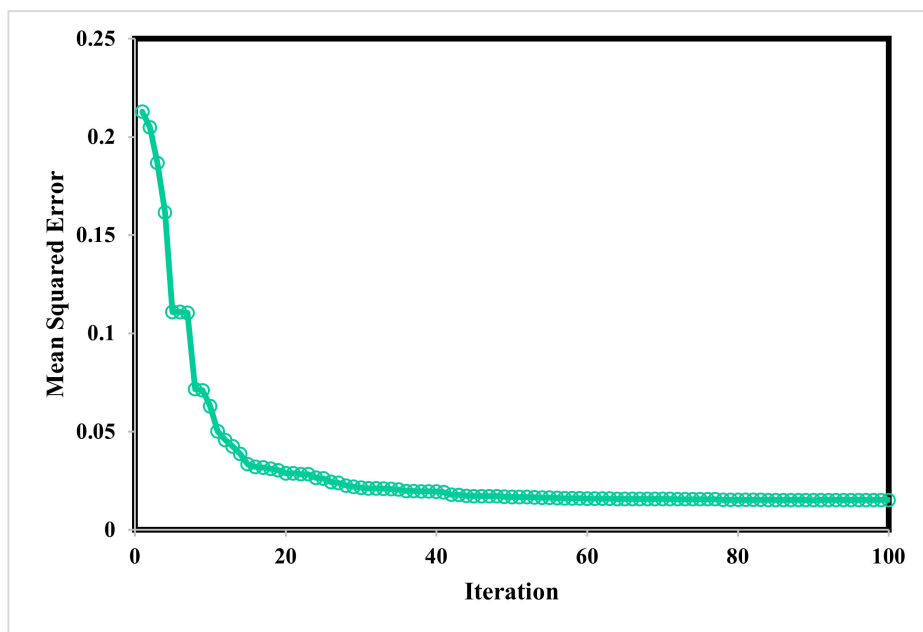


Figure 3. The performance of the LM algorithm according to the MSE of different iterations for the radial basis function (RBF)–ANN.

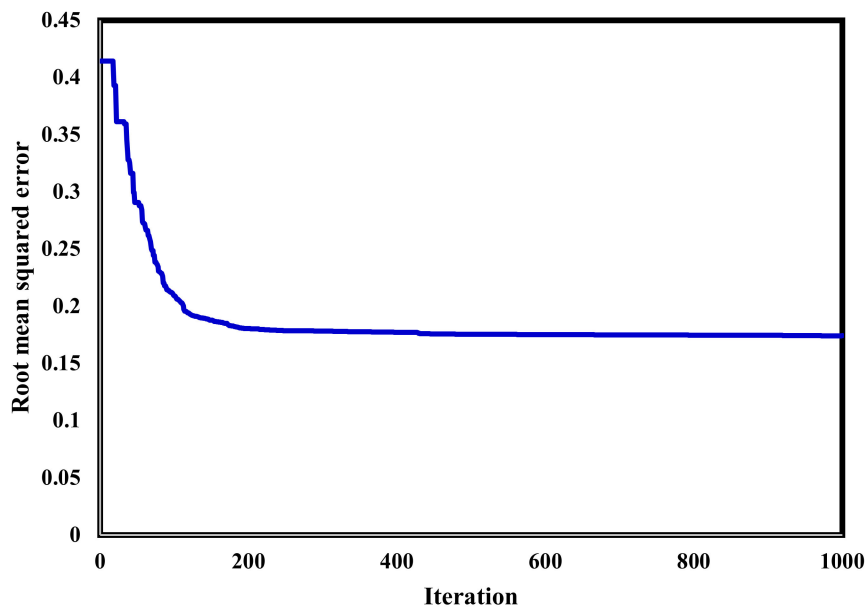


Figure 4. ANFIS performance during the training stage using the particle swarm optimization (PSO) approach (with ten clusters).

To create the structure of the SGB, the different parameters of this algorithm needed to be determined. These parameters were the number of additive terms, learning rates, minimum n in the child node, and the proportion of the sub-dataset. Table 2 gives details of the trained models. Figure S2 shows the predicted and actual (experimental) values of the C_p s, which were obtained with different models. Figure 5 illustrates the regression diagrams of four models for the computed and actual values. According to these figures, both the training and testing results showed a surprisingly good fit to a straight line for the SGB model; however, the obtained fits for the other models did not give benchmarking results compared to the SGB. The MLP-ANN algorithm seems to have produced estimations than the ANFIS and RBF-ANN algorithms. The R^2 coefficient of the SGB method was 0.994 for both the training and testing datasets. The R^2 values were 0.933 and 0.943, 0.8499 and 0.8754, and 0.9327 and 0.9204 for the training and testing datasets of the MLP-ANN, ANFIS, and RBF-ANN algorithms, respectively. According to the statistical analysis, the linear regression equations of the testing datasets of the ANFIS, MLP-ANN, RBF-ANN, and SGB models were respectively expressed as follows:

$$y = 1.0003x - 0.0071; \quad R^2 = 0.8754, \quad (26)$$

$$y = 0.988x + 0.0223; \quad R^2 = 0.943, \quad (27)$$

$$y = 0.993x + 0.0232; \quad R^2 = 0.9204, \quad (28)$$

$$y = 1.0044x - 0.0167; \quad R^2 = 0.9868. \quad (29)$$

The regression equations express the accuracy or the deviation of the calculated C_p of the ionanofluids from actual values. As these equations are more similar to the bisector line equation, they gave more accurate predictions. The relative deviations (%) between the predicted and actual C_p of the ionanofluids for these approaches are depicted in Figure S3. The MREs (%) of the testing and training data of the SGB model were 0.93789% and 0.81109%, respectively. Furthermore, these values were 1.9799% and 1.8353%, 3.5439% and 3.5403%, and 1.768% and 1.6416% for the MLP-ANN, ANFIS, and RBF-ANN, respectively. It is clear that the SGB predicted the C_p better than the other models. The RBF-ANN and MLP-ANN showed similar results, but the RBF-ANN was slightly better than the MLP-ANN. Table 3 summarizes the statistical techniques, including the R^2 , MSE, MRE, RMSE, and STD for training, testing, and total datasets for each model. The determined indexes in the training

phase show that the proposed models were trained to acceptable degrees of accuracy. After the evaluation of the training phase, it was important to identify the performance of models regarding the determination of the C_p of ionanofluids for unseen conditions. Therefore, the statistical indexes in the testing phase were investigated to ensure the generalization of the model. The generalization of models was confirmed by the determination of the error values. The low error values show that the SGB model had an interesting accuracy for unseen points.

Table 2. More details of trained models for the prediction of the C_p s of the ILs.

SGB Tree		ANFIS	
Type	Value/Comment	Type	Value/Comment
Learning rate	0.1	Membership function	Gaussian
Number of additive terms	300	No. of membership function (MF) parameters	120
Number of data points used for training	429	No. of clusters	10
Number of data points used for testing	142	Number of data points used for training	429
Minimum n in the child node	1	Number of data points used for testing	142
		Optimization method	PSO
		Population size	85
		Iteration	1000
		C_1	1
		C_2	2
MLP-ANN		RBF-ANN	
No. input neuron layer	5	No. input neuron layer	5
No. hidden neuron layer	12	No. hidden neuron layer	100
No. output neuron layer	1	No. output neuron layer	1
Hidden layer activation function	Logsig	Hidden layer activation function	RBF
Output layer activation function	Purelin	Output layer activation function	Purelin
Number of data points used for training	429	Number of data points used for training	429
Number of data points used for testing	142	Number of data points used for testing	142
Maximum iterations	1000	Maximum iterations	100

Table 3. Evaluating the performance of the proposed models using statistical analyses.

Model	Dataset	R^2	MRE (%)	MSE	RMSE	STD
ANFIS	Training	0.8499	3.5403	0.0298	0.1726	0.1520
	Testing	0.8754	3.5439	0.0320	0.1788	0.1587
	Total	0.8575	3.5412	0.0303	0.1788	0.1536
MLP-ANN	Training	0.9332	1.8353	0.0138	0.1173	0.1098
	Testing	0.9434	1.9799	0.0132	0.1147	0.1067
	Total	0.9360	1.8714	0.0136	0.1147	0.1089
RBF-ANN	Training	0.9327	1.6416	0.0134	0.1160	0.1100
	Testing	0.9204	1.7680	0.0201	0.1418	0.1360
	Total	0.9290	1.6732	0.0151	0.1418	0.1170
SGB tree	Training	0.996	0.81109	0.00089	0.02977	0.02289
	Testing	0.987	0.97389	0.00249	0.04994	0.04461
	Total	0.994	0.85179	0.00129	0.03589	0.02983

MRE: Mean relative error, MSE: mean squared error, RMSE: root mean squared error, STD: standard deviation.

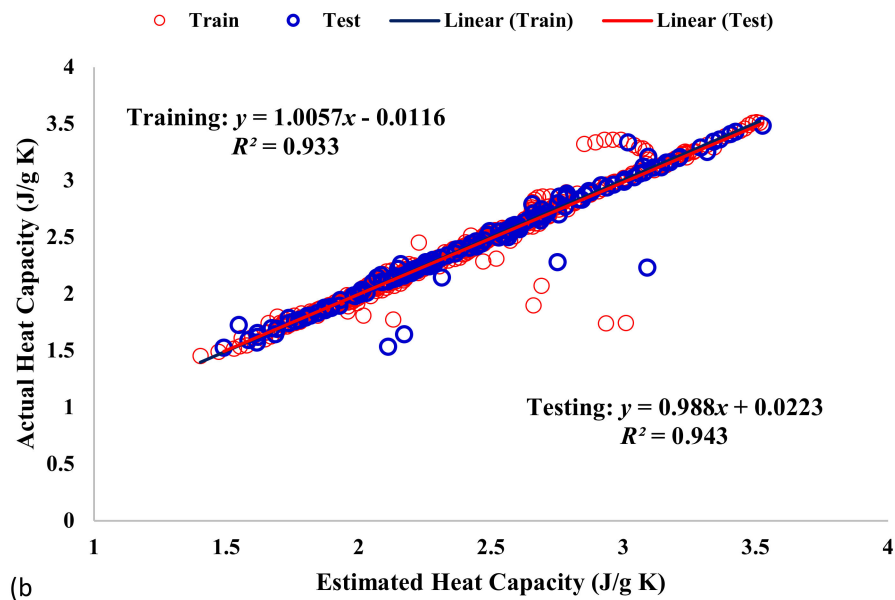
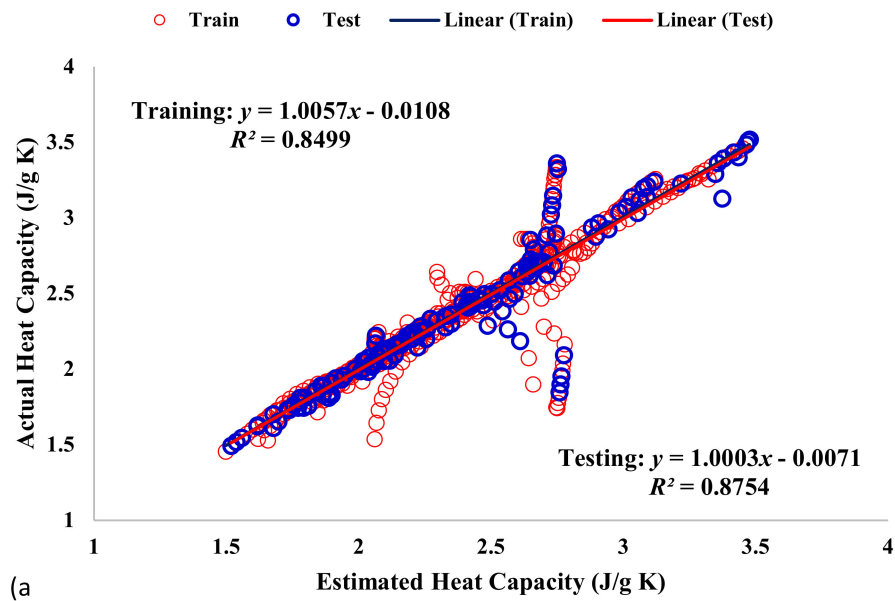


Figure 5. Cont.

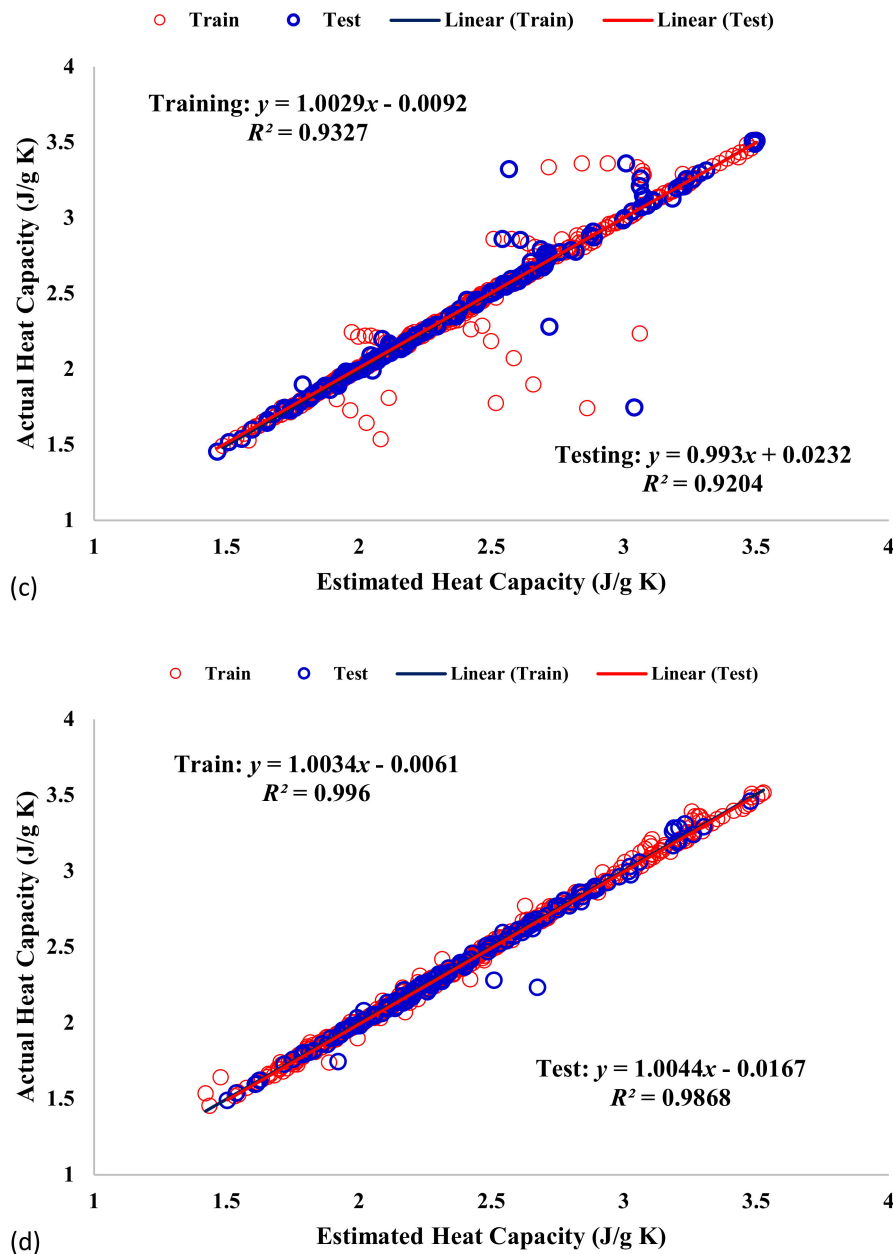


Figure 5. Regression diagram for predict C_p using different models in the training and testing steps: (a) ANFIS, (b) MLP-ANN, (c) RBF-ANN, and (d) SGB.

6. Outlier Detection

The correctness of the algorithms is strongly influenced by the precision of the laboratory values [83]. This study used a large amount of literature data, and it should be noted that some of these data may have a significant laboratory error. The trend of outliers is distinct from the general trend. It is necessary to find the exact procedure for detecting outliers to remove the imprecise experimental data [84]. In this work, the Leverage method was used to find the outliers. Based on the following equation, after obtaining the residual values, a hat matrix (H) was created for the input values [85]:

$$H = X(X^T X)^{-1} X^T. \tag{30}$$

X is an $m \times n$, where m and n are the number of model parameters and the number of samples, respectively. From the main diagonal of H , the hat values are obtained. Accordingly, a Williams plot

can graphically detect outliers that show the standard residual values versus the hat values. Because the predictive power of the SGB was better than the ANFIS, the SGB was used to analyze the outlier detection. Figure 6 illustrates Williams plots for the various models investigated. The critical leverage value (H^*) was calculated according to this equation:

$$H^* = \frac{3(n + 1)}{m}, \tag{31}$$

where the blue lines in the figures indicate the leverage limit and the data points that have lower critical leverage value (H^*) than the hat values (H) are known as outliers. Moreover, the red lines $y = \pm 3$ are borders, where data points with standardized residuals outside these two lines are regarded as outliers.

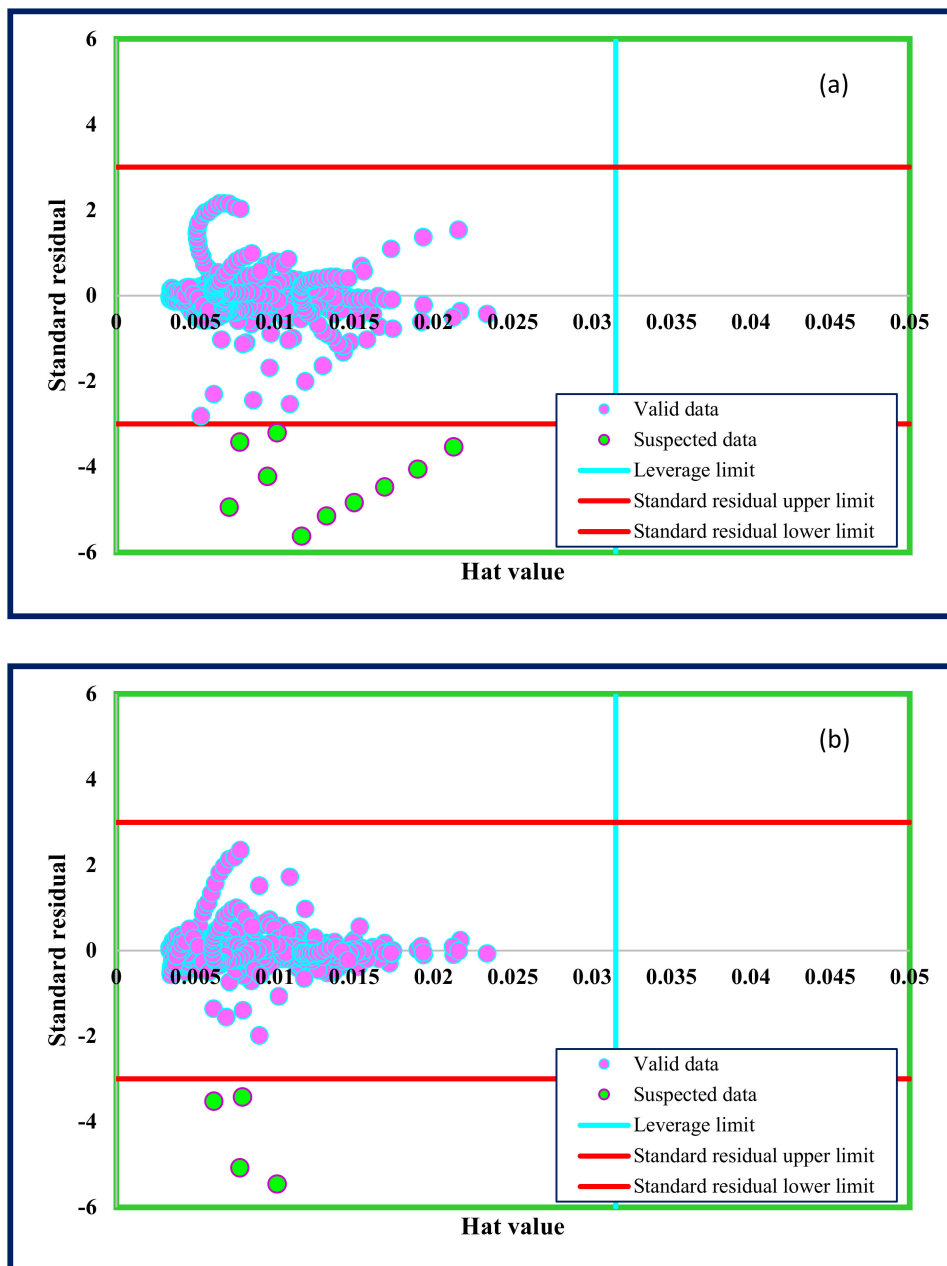


Figure 6. Cont.

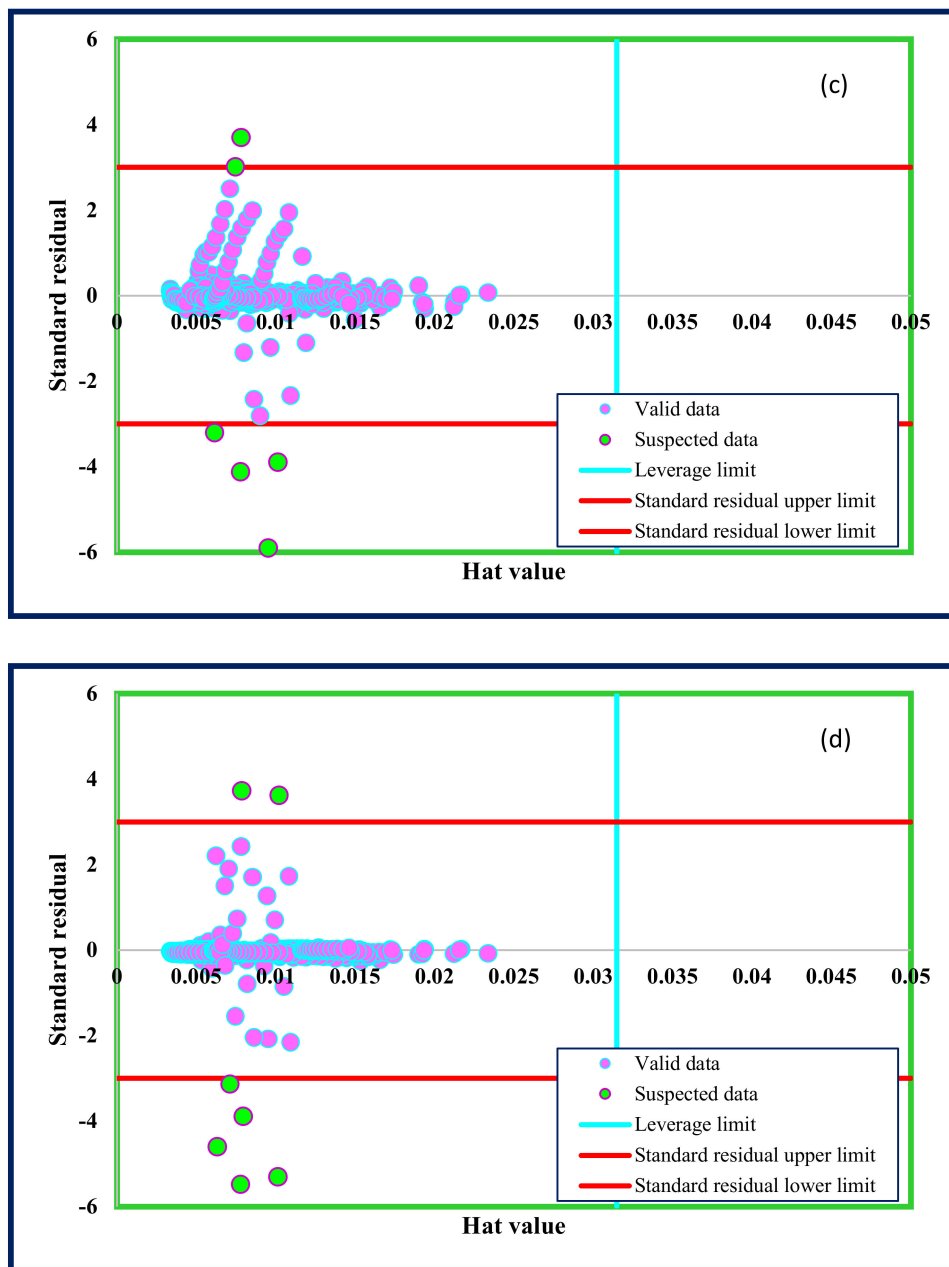


Figure 6. The detection of outlying data for the different models: (a) ANFIS, (b) MLP-ANN, (c) RBF-ANN, and (d) SGB.

Importance of the Input Parameters

To determine which inputs had the greatest impact on C_p , we used the relevancy factor (r), where r is in the range of $[-1, 1]$. The r values are calculated using Equation (32) [84]:

$$r = \frac{\sum_{i=1}^n (X_{k,i} - \bar{X}_k)(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_{k,i} - \bar{X}_k)^2 \sum_{i=1}^n (Y_i - \bar{Y})^2}} \tag{32}$$

where $X_{k,i}$ and Y_i are the i th input and output, respectively; \bar{Y} and \bar{X}_k are the average value of output and k th input, respectively; n denotes the total number of data points. As shown in Figure 7, C_p was directly related to T , M_w , and T_c , and had an inverse relation with ω and x . M_w and T_c had the highest and ω had the lowest impact on C_p (with r equal to 0.451).

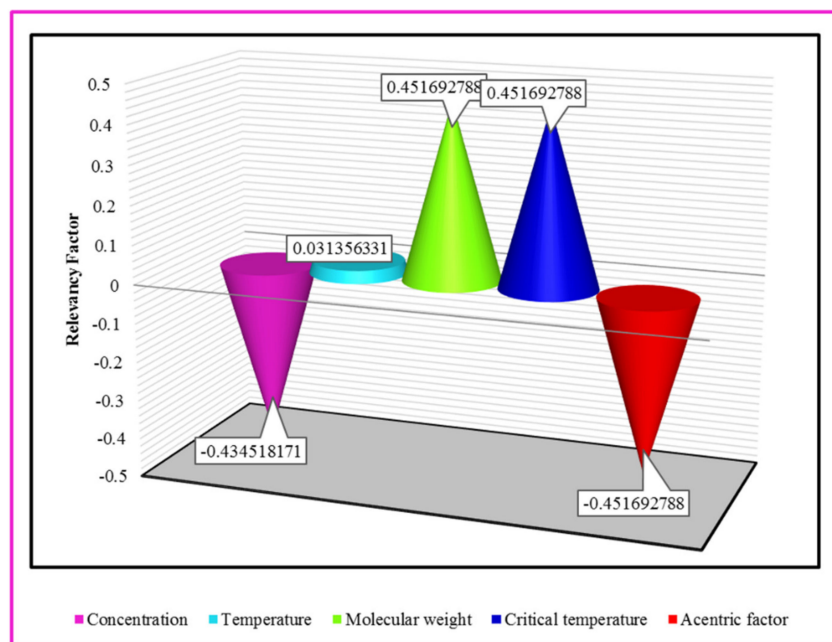


Figure 7. Sensitivity analysis to determine the effect of the inputs on the C_p values of ILs.

7. Conclusions

In this work, the predictive capability of four groups of intelligence models was evaluated to determine the C_p of ionic liquids in extensive conditions based on a wide database containing 571 data points gathered from literature reviews. C_p was estimated by considering the properties of ILs, the nanoparticle concentration, and operational temperature as input parameters. Moreover, the dependent parameters of the ANFIS were optimized using PSO. The LM algorithm was used to determine the tuning parameters of the ANN. The PSO displayed an excellent ability to determine the best values of the ANFIS parameters. The outstanding aspects of this study are its easy and quick calculation and the low number of adjustable parameters for the calculation methods. The statistical analyses of the SGB method showed highly satisfactory predictions compared with the other models. Fortunately, the SGB model presented here has simple calculations. Using it in commercial software or as an alternative tool when there is no empirical data is another of its applications.

Supplementary Materials: The following are available online at <http://www.mdpi.com/2076-3417/10/18/6432/s1>, Figures S1–S3, and Tables S1 and S2.

Author Contributions: Data curation, M.H. and M.S.; formal analysis, R.D., A.B., and H.M.A.; investigation, H.M.A., I.M., and T.A.; methodology, R.D., M.H. and A.B.; project administration, M.H., M.S., and H.M.A.; resources, I.M., T.A., and H.M.A.; software, M.H., M.S., and R.D.; supervision, M.S. and T.A.; writing—original draft, R.D.; writing—review and editing, A.B. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rehman, T.-U.; Ali, H.M.; Janjua, M.M.; Sajjad, U.; Yan, W.-M. A critical review on heat transfer augmentation of phase change materials embedded with porous materials/foams. *Int. J. Heat Mass Transf.* **2019**, *135*, 649–673. [CrossRef]
2. Sajid, M.U.; Ali, H.M. Thermal conductivity of hybrid nanofluids: A critical review. *Int. J. Heat Mass Transf.* **2018**, *126*, 211–234. [CrossRef]
3. Arshad, W.; Ali, H.M. Experimental investigation of heat transfer and pressure drop in a straight minichannel heat sink using TiO₂ nanofluid. *Int. J. Heat Mass Transf.* **2017**, *110*, 248–256. [CrossRef]

4. Qureshi, Z.A.; Ali, H.M.; Khushnood, S. Recent advances on thermal conductivity enhancement of phase change materials for energy storage system: A review. *Int. J. Heat Mass Transf.* **2018**, *127*, 838–856. [[CrossRef](#)]
5. Abbas, F.; Ali, H.M.; Shah, T.R.; Babar, H.; Janjua, M.M.; Sajjad, U.; Amer, M. Nanofluid: Potential evaluation in automotive radiator. *J. Mol. Liq.* **2019**, *297*, 112014. [[CrossRef](#)]
6. Tariq, R.; Hussain, Y.; Sheikh, N.A.; Afaq, K.; Ali, H.M. Regression-Based Empirical Modeling of Thermal Conductivity of CuO-Water Nanofluid using Data-Driven Techniques. *Int. J. Thermophys.* **2020**, *41*, 1–28. [[CrossRef](#)]
7. Ali, H.M. In tube convection heat transfer enhancement: SiO₂ aqua based nanofluids. *J. Mol. Liq.* **2020**, *308*, 113031. [[CrossRef](#)]
8. Wasserscheid, P.; Welton, T. *Ionic Liquids in Synthesis*; John Wiley & Sons: Hoboken, NJ, USA, 2008.
9. Ferreira, A.; Simões, P.; Ferreira, A.; Fonseca, M.; Oliveira, M.; Trino, A. Transport and thermal properties of quaternary phosphonium ionic liquids and Ionanofluids. *J. Chem. Thermodyn.* **2013**, *64*, 80–92. [[CrossRef](#)]
10. Cui, X.; Zhu, H.; He, C.-H.; Wu, K.-J. Measurement of the thermal conductivity of 1-butyl-3-methylimidazolium l-tryptophan+ water+ ethanol mixtures at T = (283.15 to 333.15) K. *J. Chem. Eng. Data* **2019**, *64*, 1586–1593. [[CrossRef](#)]
11. Sánchez-Badillo, J.; Gallo, M.; Guirado-López, R.A.; Lopez-Lemus, J. Thermodynamic, structural and dynamic properties of ionic liquids [C 4 mim][CF 3 COO],[C 4 mim][Br] in the condensed phase, using molecular simulations. *RSC Adv.* **2019**, *9*, 13677–13695. [[CrossRef](#)]
12. Hasani, M.; Varela, L.M.; Martinelli, A. Short-Range Order and Transport Properties in Mixtures of the Protic Ionic Liquid [C2HIm][TFSI] with Water or Imidazole. *J. Phys. Chem. B* **2020**, *124*, 1767–1777. [[CrossRef](#)]
13. Van Valkenburg, M.E.; Vaughn, R.L.; Williams, M.; Wilkes, J.S. Thermochemistry of ionic liquid heat-transfer fluids. *Thermochim. Acta* **2005**, *425*, 181–188. [[CrossRef](#)]
14. Paul, T.C.; Morshed, A.; Fox, E.B.; Visser, A.E.; Bridges, N.J.; Khan, J.A. Thermal performance of ionic liquids for solar thermal applications. *Exp. Therm. Fluid Sci.* **2014**, *59*, 88–95. [[CrossRef](#)]
15. Bridges, N.J.; Visser, A.E.; Fox, E.B. Potential of nanoparticle-enhanced ionic liquids (NEILs) as advanced heat-transfer fluids. *Energy Fuels* **2011**, *25*, 4862–4864. [[CrossRef](#)]
16. Nieto de Castro, C.; Lourenço, M.; Ribeiro, A.; Langa, E.; Vieira, S.; Goodrich, P.; Hardacre, C. Thermal properties of ionic liquids and ionanofluids of imidazolium and pyrrolidinium liquids. *J. Chem. Eng. Data* **2009**, *55*, 653–661. [[CrossRef](#)]
17. Liu, J.; Wang, F.; Zhang, L.; Fang, X.; Zhang, Z. Thermodynamic properties and thermal stability of ionic liquid-based nanofluids containing graphene as advanced heat transfer fluids for medium-to-high-temperature applications. *Renew. Energy* **2014**, *63*, 519–523. [[CrossRef](#)]
18. Wang, F.; Han, L.; Zhang, Z.; Fang, X.; Shi, J.; Ma, W. Surfactant-free ionic liquid-based nanofluids with remarkable thermal conductivity enhancement at very low loading of graphene. *Nanoscale Res. Lett.* **2012**, *7*, 314. [[CrossRef](#)]
19. Dehaghani, A.H.S.; Daneshfar, R. How much would silica nanoparticles enhance the performance of low-salinity water flooding? *Pet. Sci.* **2019**, *16*, 591–605. [[CrossRef](#)]
20. Paul, T.C.; Morshed, A.M.; Khan, J.A. Nanoparticle enhanced ionic liquids (NEILs) as working fluid for the next generation solar collector. *Procedia Eng.* **2013**, *56*, 631–636. [[CrossRef](#)]
21. Paul, T.C.; Morshed, A.; Fox, E.B.; Khan, J.A. Thermal performance of Al₂O₃ nanoparticle enhanced ionic liquids (NEILs) for concentrated solar power (CSP) applications. *Int. J. Heat Mass Transf.* **2015**, *85*, 585–594. [[CrossRef](#)]
22. Waghole, D.; Warkhedkar, R.; Kulkarni, V.; Shrivastva, R. Studies on heat transfer in flow of silver nanofluid through a straight tube with twisted tape inserts. *Heat Mass Transf.* **2016**, *52*, 309–313. [[CrossRef](#)]
23. De Castro, C.N.; Murshed, S.S.; Lourenço, M.; Santos, F.; Lopes, M.; França, J. Enhanced thermal conductivity and specific heat capacity of carbon nanotubes ionanofluids. *Int. J. Therm. Sci.* **2012**, *62*, 34–39. [[CrossRef](#)]
24. Iqbal, M.; Sheikh, N.; Ali, H.; Khushnood, S.; Arif, M. Comparison of empirical correlations for the estimation of conjugate heat transfer in a thrust chamber. *Life Sci. J.* **2012**, *9*, 708–716.
25. Ali, H.M.; Ali, A. Measurements and semi-empirical correlation for condensate retention on horizontal integral-fin tubes: Effect of vapour velocity. *Appl. Therm. Eng.* **2014**, *71*, 24–33. [[CrossRef](#)]
26. Ali, H.M.; Briggs, A. A semi-empirical model for free-convection condensation on horizontal pin-fin tubes. *Int. J. Heat Mass Transf.* **2015**, *81*, 157–166. [[CrossRef](#)]

27. Ali, H.M. An analytical model for prediction of condensate flooding on horizontal pin-fin tubes. *Int. J. Heat Mass Transf.* **2017**, *106*, 1120–1124. [[CrossRef](#)]
28. Chaudhary, G.Q.; Ali, M.; Ashiq, M.; Ali, H.M.; Amber, K.P. Experimental and model based performance investigation of a solid desiccant wheel dehumidifier in subtropical climate. *Therm. Sci.* **2019**, *23*, 975–988. [[CrossRef](#)]
29. Siddiqui, A.M.; Arshad, W.; Ali, H.M.; Ali, M.; Nasir, M.A. Evaluation of nanofluids performance for simulated microprocessor. *Therm. Sci.* **2017**, *21*, 2227–2236. [[CrossRef](#)]
30. Anwar, M.; Tariq, H.A.; Shoukat, A.A.; Ali, H.M.; Ali, H. Numerical study for heat transfer enhancement using CuO-H₂O nano-fluids through mini-channel heat sinks for microprocessor cooling. *Therm. Sci.* **2019**, *22*–22. [[CrossRef](#)]
31. Tariq, H.A.; Anwar, M.; Ali, H.M.; Ahmed, J. Effect of dual flow arrangements on the performance of mini-channel heat sink: Numerical study. *J. Therm. Anal. Calorim.* **2020**. [[CrossRef](#)]
32. Dehaghani, A.H.S.; Taleghani, M.S.; Badizad, M.H.; Daneshfar, R. Simulation study of the Gachsaran asphaltene behavior within the interface of oil/water emulsion: A case study. *Colloid Interface Sci. Commun.* **2019**, *33*, 100202. [[CrossRef](#)]
33. Soltanian, M.R.; Hajirezaie, S.; Hosseini, S.A.; Dashtian, H.; Amooie, M.A.; Meyal, A.; Ershadnia, R.; Ampomah, W.; Islam, A.; Zhang, X. Multicomponent reactive transport of carbon dioxide in fluvial heterogeneous aquifers. *J. Nat. Gas Sci. Eng.* **2019**, *65*, 212–223. [[CrossRef](#)]
34. Ershadnia, R.; Amooie, M.A.; Shams, R.; Hajirezaie, S.; Liu, Y.; Jamshidi, S.; Soltanian, M.R. Non-Newtonian fluid flow dynamics in rotating annular media: Physics-based and data-driven modeling. *J. Pet. Sci. Eng.* **2020**, *185*, 106641. [[CrossRef](#)]
35. Jiang, B.; Zhao, F. Combination of support vector regression and artificial neural networks for prediction of critical heat flux. *Int. J. Heat Mass Transf.* **2013**, *62*, 481–494. [[CrossRef](#)]
36. Peng, H.; Ling, X. Predicting thermal-hydraulic performances in compact heat exchangers by support vector regression. *Int. J. Heat Mass Transf.* **2015**, *84*, 203–213. [[CrossRef](#)]
37. Kamble, L.; Pangavhane, D.; Singh, T. Neural network optimization by comparing the performances of the training functions-Prediction of heat transfer from horizontal tube immersed in gas–solid fluidized bed. *Int. J. Heat Mass Transf.* **2015**, *83*, 337–344. [[CrossRef](#)]
38. Nabipour, N.; Daneshfar, R.; Rezvanjou, O.; Mohammadi-Khanaposhtani, M.; Baghban, A.; Xiong, Q.; Li, L.K.; Habibzadeh, S.; Doranehgard, M.H. Estimating biofuel density via a soft computing approach based on intermolecular interactions. *Renew. Energy* **2020**, *152*, 1086–1098. [[CrossRef](#)]
39. Vanani, M.B.; Daneshfar, R.; Khodapanah, E. A novel MLP approach for estimating asphaltene content of crude oil. *Pet. Sci. Technol.* **2019**, *37*, 2238–2245. [[CrossRef](#)]
40. Karimi, H.; Yousefi, F.; Rahimi, M.R. Correlation of viscosity in nanofluids using genetic algorithm-neural network (GA-NN). *Heat Mass Transf.* **2011**, *47*, 1417–1425. [[CrossRef](#)]
41. Atashrouz, S.; Mozaffarian, M.; Pazuki, G. Modeling the thermal conductivity of ionic liquids and ionanofluids based on a group method of data handling and modified Maxwell model. *Ind. Eng. Chem. Res.* **2015**, *54*, 8600–8610. [[CrossRef](#)]
42. Sadi, M. Prediction of Thermal Conductivity and Viscosity of Ionic Liquid-Based Nanofluids Using Adaptive Neuro Fuzzy Inference System. *Heat Transf. Eng.* **2017**, *38*, 1561–1572. [[CrossRef](#)]
43. Salehi, H.; Zeinali-Heris, S.; Esfandyari, M.; Koolivand, M. Neuro-fuzzy modeling of the convection heat transfer coefficient for the nanofluid. *Heat Mass Transf.* **2013**, *49*, 575–583. [[CrossRef](#)]
44. Mehrabi, M.; Sharifpur, M.; Meyer, J.P. Application of the FCM-based neuro-fuzzy inference system and genetic algorithm-polynomial neural network approaches to modelling the thermal conductivity of alumina–water nanofluids. *Int. Commun. Heat Mass Transf.* **2012**, *39*, 971–977. [[CrossRef](#)]
45. Golzar, K.; Amjad-Iranagh, S.; Modarress, H. Prediction of density, surface tension, and viscosity of quaternary ammonium-based ionic liquids ([N222 (n)] Tf2N) by means of artificial intelligence techniques. *J. Dispers. Sci. Technol.* **2014**, *35*, 1809–1829. [[CrossRef](#)]
46. Soriano, A.N.; Ornedo-Ramos, K.F.P.; Muriel, C.A.M.; Adornado, A.P.; Bungay, V.C.; Li, M.-H. Prediction of refractive index of binary solutions consisting of ionic liquids and alcohols (methanol or ethanol or 1-propanol) using artificial neural network. *J. Taiwan Inst. Chem. Eng.* **2016**, *65*, 83–90. [[CrossRef](#)]

47. Lashkarblooki, M.; Hezave, A.Z.; Al-Ajmi, A.M.; Ayatollahi, S. Viscosity prediction of ternary mixtures containing ILs using multi-layer perceptron artificial neural network. *Fluid Phase Equilibria* **2012**, *326*, 15–20. [[CrossRef](#)]
48. Hezave, A.Z.; Lashkarblooki, M.; Raeissi, S. Using artificial neural network to predict the ternary electrical conductivity of ionic liquid systems. *Fluid Phase Equilibria* **2012**, *314*, 128–133. [[CrossRef](#)]
49. Friedman, J.H. Stochastic gradient boosting. *Comput. Stat. Data Anal.* **2002**, *38*, 367–378. [[CrossRef](#)]
50. Aertsen, W.; Kint, V.; Van Orshoven, J.; Özkan, K.; Muys, B. Comparison and ranking of different modelling techniques for prediction of site index in Mediterranean mountain forests. *Ecol. Model.* **2010**, *221*, 1119–1130. [[CrossRef](#)]
51. Moisen, G.G.; Freeman, E.A.; Blackard, J.A.; Frescino, T.S.; Zimmermann, N.E.; Edwards, T.C., Jr. Predicting tree species presence and basal area in Utah: A comparison of stochastic gradient boosting, generalized additive models, and tree-based methods. *Ecol. Model.* **2006**, *199*, 176–187. [[CrossRef](#)]
52. Soleimani, R.; Mahmood, T.; Bahadori, A. Assessment of compressor power and condenser duty per refrigeration duty in three-stage propane refrigerant systems using a new ensemble learning tool. In *Chemeca 2016: Chemical Engineering-Regeneration, Recovery and Reinvention*; Engineers Australia: Melbourne, Australia, 2016; p. 23.
53. Stevens, K.B.; Pfeiffer, D.U. Spatial modelling of disease using data-and knowledge-driven approaches. *Spat. Spatio Temporal Epidemiol.* **2011**, *2*, 125–133. [[CrossRef](#)] [[PubMed](#)]
54. Elith, J.; Leathwick, J.R.; Hastie, T. A working guide to boosted regression trees. *J. Anim. Ecol.* **2008**, *77*, 802–813. [[CrossRef](#)]
55. Müller, D.; Leitão, P.J.; Sikor, T. Comparing the determinants of cropland abandonment in Albania and Romania using boosted regression trees. *Agric. Syst.* **2013**, *117*, 66–77. [[CrossRef](#)]
56. Filippi, A.M.; Güneralp, İ.; Randall, J. Hyperspectral remote sensing of aboveground biomass on a river meander bend using multivariate adaptive regression splines and stochastic gradient boosting. *Remote Sens. Lett.* **2014**, *5*, 432–441. [[CrossRef](#)]
57. Mohanraj, M.; Jayaraj, S.; Muraleedharan, C. Applications of artificial neural networks for thermal analysis of heat exchangers—a review. *Int. J. Therm. Sci.* **2015**, *90*, 150–172. [[CrossRef](#)]
58. Xie, G.; Sunden, B.; Wang, Q.; Tang, L. Performance predictions of laminar and turbulent heat transfer and fluid flow of heat exchangers having large tube-diameter and large tube-row by artificial neural networks. *Int. J. Heat Mass Transf.* **2009**, *52*, 2484–2497. [[CrossRef](#)]
59. Zendehboudi, A.; Saidur, R.; Mahbubul, I.; Hosseini, S. Data-driven methods for estimating the effective thermal conductivity of nanofluids: A comprehensive review. *Int. J. Heat Mass Transf.* **2019**, *131*, 1211–1231. [[CrossRef](#)]
60. Benli, H. Determination of thermal performance calculation of two different types solar air collectors with the use of artificial neural networks. *Int. J. Heat Mass Transf.* **2013**, *60*, 1–7. [[CrossRef](#)]
61. Yao, X.; Panaye, A.; Doucet, J.-P.; Zhang, R.; Chen, H.; Liu, M.; Hu, Z.; Fan, B.T. Comparative study of QSAR/QSPR correlations using support vector machines, radial basis function neural networks, and multiple linear regression. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1257–1266. [[CrossRef](#)]
62. Girosi, F.; Poggio, T. Networks and the best approximation property. *Biol. Cybern.* **1990**, *63*, 169–176. [[CrossRef](#)]
63. Jang, J.-S.R.; Sun, C.-T.; Mizutani, E. Neuro-fuzzy and soft computing; a computational approach to learning and machine intelligence. *IEEE Trans. Autom. Control* **1997**, *42*, 1482–1484. [[CrossRef](#)]
64. Afshar, M.; Gholami, A.; Asoodeh, M. Genetic optimization of neural network and fuzzy logic for oil bubble point pressure modeling. *Korean J. Chem. Eng.* **2014**, *31*, 496–502. [[CrossRef](#)]
65. Baghban, A.; Ahmadi, M.A.; Shahraki, B.H. Prediction carbon dioxide solubility in presence of various ionic liquids using computational intelligence approaches. *J. Supercrit. Fluids* **2015**, *98*, 50–64. [[CrossRef](#)]
66. Baghban, A.; Kahani, M.; Nazari, M.A.; Ahmadi, M.H.; Yan, W.-M. Sensitivity analysis and application of machine learning methods to predict the heat transfer performance of CNT/water nanofluid flows through coils. *Int. J. Heat Mass Transf.* **2019**, *128*, 825–835. [[CrossRef](#)]
67. Ahmadi, M.H.; Tatar, A.; Nazari, M.A.; Ghasempour, R.; Chamkha, A.J.; Yan, W.-M. Applicability of connectionist methods to predict thermal resistance of pulsating heat pipes with ethanol by using neural networks. *Int. J. Heat Mass Transf.* **2018**, *126*, 1079–1086. [[CrossRef](#)]

68. Ahangari, K.; Moeinossadat, S.R.; Behnia, D. Estimation of tunnelling-induced settlement by modern intelligent methods. *Soils Found.* **2015**, *55*, 737–748. [[CrossRef](#)]
69. Jang, J.-S.; Sun, C.-T. Neuro-fuzzy modeling and control. *Proc. IEEE* **1995**, *83*, 378–406. [[CrossRef](#)]
70. Hu, X.; Shi, Y.; Eberhart, R. Recent advances in particle swarm. In Proceedings of the 2004 Congress on Evolutionary Computation, Portland, OR, USA, 19–23 June 2004; pp. 90–97.
71. Kennedy, J. *Particle Swarm Optimization Encyclopedia of Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 760–766.
72. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS'95), Nagoya, Japan, 4–6 October 1995; pp. 39–43.
73. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization. *Swarm Intell.* **2007**, *1*, 33–57. [[CrossRef](#)]
74. Hastie, T.; Tibshirani, R.; Friedman, J.; Franklin, J. The elements of statistical learning: Data mining, inference and prediction. *Math. Intell.* **2005**, *27*, 83–85.
75. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
76. Whiting, D.G.; Hansen, J.V.; McDonald, J.B.; Albrecht, C.; Albrecht, W.S. Machine learning methods for detecting patterns of management fraud. *Comput. Intell.* **2012**, *28*, 505–527. [[CrossRef](#)]
77. Paul, T.C.; Morshed, A.; Fox, E.B.; Khan, J.A. Enhanced thermophysical properties of NEILs as heat transfer fluids for solar thermal applications. *Appl. Therm. Eng.* **2017**, *110*, 1–9. [[CrossRef](#)]
78. Paul, T.C. Investigation of Thermal Performance of Nanoparticle Enhanced Ionic Liquids (NEILs) for Solar Collector Applications. Ph.D. Thesis, University of South Carolina, Columbia, SC, USA, 2014.
79. Paul, T.C.; Morshed, A.; Fox, E.B.; Khan, J.A. Experimental investigation of natural convection heat transfer of Al₂O₃ nanoparticle enhanced ionic liquids (NEILs). *Int. J. Heat Mass Transf.* **2015**, *83*, 753–761. [[CrossRef](#)]
80. Valderrama, J.; Robles, P. Critical properties, normal boiling temperatures, and acentric factors of fifty ionic liquids. *Ind. Eng. Chem. Res.* **2007**, *46*, 1338–1344. [[CrossRef](#)]
81. Valderrama, J.O.; Sanga, W.W.; Lazzús, J.A. Critical properties, normal boiling temperature, and acentric factor of another 200 ionic liquids. *Ind. Eng. Chem. Res.* **2008**, *47*, 1318–1330. [[CrossRef](#)]
82. Chen, S.; Billings, S.; Grant, P. Non-linear system identification using neural networks. *Int. J. Control* **1990**, *51*, 1191–1214. [[CrossRef](#)]
83. Rousseeuw, P.J.; Leroy, A.M. *Robust Regression and Outlier Detection*; John Wiley & Sons: Hoboken, NJ, USA, 2005; Volume 589.
84. Hosseinzadeh, M.; Hemmati-Sarapardeh, A. Toward a predictive model for estimating viscosity of ternary mixtures containing ionic liquids. *J. Mol. Liq.* **2014**, *200*, 340–348. [[CrossRef](#)]
85. Mohammadi, A.H.; Gharagheizi, F.; Eslamimanesh, A.; Richon, D. Evaluation of experimental data for wax and diamondoids solubility in gaseous systems. *Chem. Eng. Sci.* **2012**, *81*, 1–7. [[CrossRef](#)]

