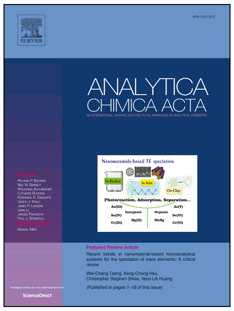
Accepted Manuscript



Strategies to develop robust neural network models: prediction of flash point as a case study

Amin Alibakshi

PII: S0003-2670(18)30604-4

DOI: 10.1016/j.aca.2018.05.015

Reference: ACA 235952

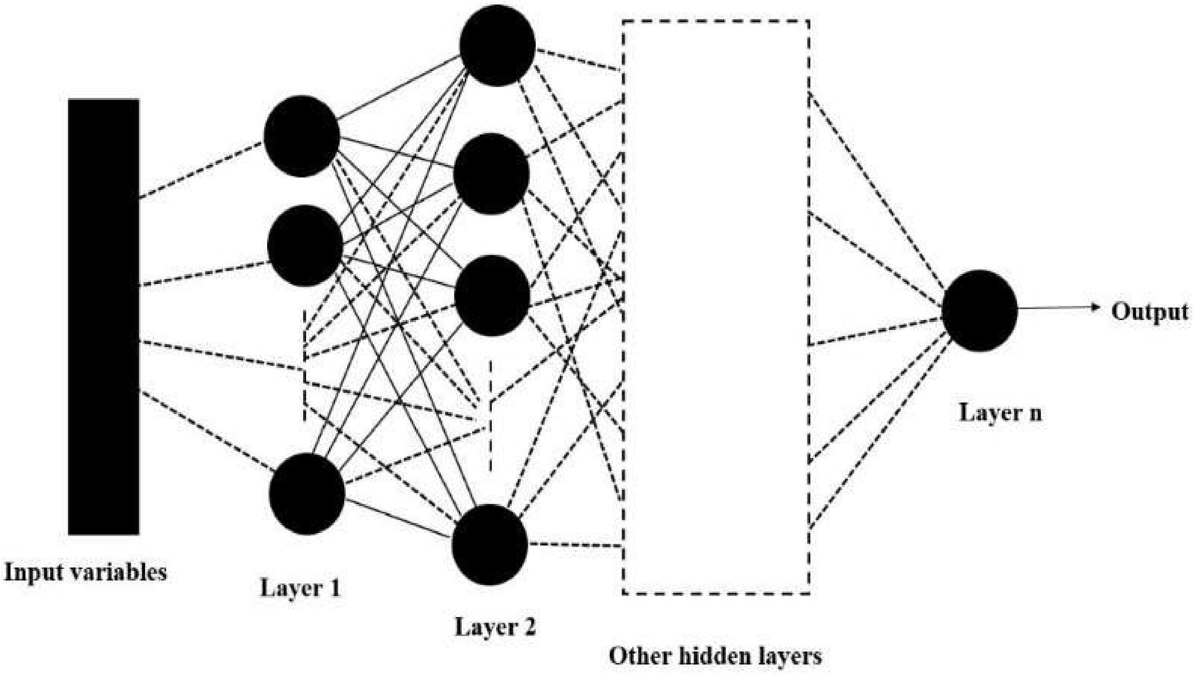
To appear in: *Analytica Chimica Acta*

Received Date: 7 August 2017

Revised Date: 2 May 2018

Accepted Date: 3 May 2018

Please cite this article as: A. Alibakshi, Strategies to develop robust neural network models: prediction of flash point as a case study, *Analytica Chimica Acta* (2018), doi: 10.1016/j.aca.2018.05.015.


Input variables
Layer 1
Layer 2
Other hidden layers
Layer n
Output

Strategies to develop robust neural network models: prediction of flash point as a case study

$Amin\ Alibakhshi^{1,*}$

Amin.alibakhshi@hotmail.com;aalibakhshi@geomar.de

Geomar Helmholtz Center for Ocean Research Kiel, Wischhofstrasse 1-3, 24148 Kiel, Germany

Abstract

Artificial neural network (ANN) is one of the most widely used methods to develop accurate predictive models based on artificial intelligence and machine learning. In the present study, the important practical aspects of developing a reliable ANN model e.g. appropriate assignment of the number of neurons, number of hidden layers, transfer function, training algorithm, dataset division and initialization of the network are discussed. As a case study, predictability of the flash point for a dataset of 740 organic compounds using ANNs was investigated. A total number of 484220ANNs were studied to allow covering a wide range of parameters affecting the performance of an ANN. Among all studied parameters, the number of neurons or layers was found to be the most important parameters to develop a reliable ANN with low overfitting risk. To evaluate appropriate number of neurons and layers, a value of equal or greater than 10 for the ratio of the training samples to the ANN constants was suggested as a rule of thumb. More ever, a strategy for evaluation of the authentic performance of ANNs and deciding about the reliability of an ANN model was proposed.

Based on the introduced considerations, an ANN model was proposed for predicting the flash point of pure organic compounds. According to the results, the new model was found to produce the lowest error compared to other available models.

Keywords—*Artificial neural networks, Predictive models, Group contribution method, QSPR, QSAR, Flash point*

1. Introduction

Artificial neural network (ANN) is one of the most efficient tools that work based on artificial intelligence and machine learning. ANNs are capable of doing several tasks such as function approximation [1], pattern recognition [2], data clustering [3], prediction of time series [4], and so on. To provide the best performance, various types of neural networks are developed and characterized depending on the application. However, despite their slight differences, all of them follow the same basics taken from the learning mechanisms of the biological neural networks [5].

Model development which is a function approximation problem, is probably the most widely used application of the ANNs in chemistry and chemical engineering [6-11]. The most appropriate ANN for model development is the multilayer network shown in figure 1, known as the feedforward neural network. For model development using a multilayer feedforward neural network, the input variables are introduced to the network as a vector and are processed by the neurons of the first layer. Each neuron in the first layer is connected to all of the input variables and for each connection, a weight constant is assigned. The summation of all input variables multiplied by their respective weights and a bias constant yields the input of each neuron. A transfer function modifies the inputs to result the output of each neuron which is then transmitted to the neurons of the next layer to be processed further in the same way.

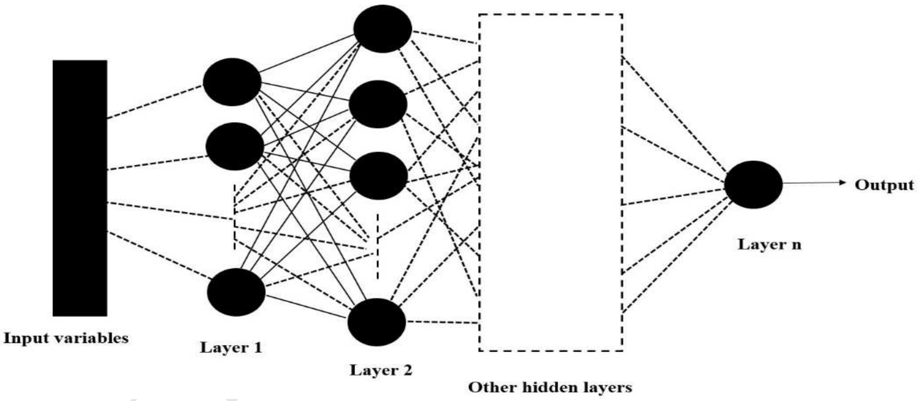


Figure 1- the configuration of a feedforward neural network

To develop an ANN model, the weights, biases and transfer functions are determined in a way that each set of inputs result in a final output equivalent to the required property. To do so, using a dataset with known input and output data (training dataset), the optimum values for the constants are determined through a procedure which is called the training of the network. Various training algorithms are developed and can be used for this purpose. With appropriate number of neurons, transfer functions and training algorithm, a multilayer feedforward ANN is capable of modeling any linear or non-linear correlation between the input and output variables[12].

Two widely used methods which apply ANNs to model chemical compounds properties in chemistry and chemical engineering are group contribution method (GCM) and quantitative structure property relationship (QSPR).

According to the GCM, properties of a compound are predicted based on the number and types of its constituting functional groups. The simplest form of a GCM based model has the form of:

$$\boldsymbol{\varphi} = \boldsymbol{c} + \sum \boldsymbol{n_i}\, \boldsymbol{\varphi_i}, \qquad (1)$$

where φ is the required property, n_i and φ_i are the number of presence and amount of contribution of functional group i, respectively, and c is a constant.

Prediction of properties via equation (1) is known as the Joback method. The Joback method typically produces poor results for large datasets. However, these results become considerably more accurate when the correlation between functional groups and the required property are mapped via ANNs. For example, using a feedforward neural network with one hidden layer containing 7 neurons, Albahri could predict the flash points of 375 transportation fuels based on the GCM with an average absolute relative error (*AARE*) of 1.1 %, while the Joback method resulted an *AARE* of 4.3 % for the same dataset and functional groups [13].

The group contribution based models which use ANNs have been widely used to predict various properties such as liquid viscosity [14-15], thermal conductivity [16], infinite dilution activity [17], and density of ionic liquids [18], normal boiling point (NBP) [19], flash point (FP) [13] and melting point [20].

Contrary to the classic GCM which only considers the functional groups as contributors to a property, the QSPR applies a more extensive set of structure based quantities, known as molecular descriptors, to model a property. To develop a QSPR model, the most effective molecular descriptors are screened from a pool of numerous calculated descriptors and are used as the inputs of the model. ANN based QSPR models have also been extensively used to predict various properties e.g. NBP [21], FP [22-24], surface tension [25], ideal gas entropy [26], aqueous solubility [27], Hildebrand solubility parameter [28] and so on.

Using several available software tools e.g. Matlab, R, and Neurosolutions, developing an ANN model has become considerably straightforward, without requiring any knowledge of its extensive theoretical details. While simplified in practice, ANN models soon become unreliable without full consideration of important details e.g. appropriate assignment of the number of neurons, number of layers, and training of the network. More ever, selecting appropriate transfer function, training algorithm, dataset division and initialization of the network can also considerably improve the reliability and performance of an ANN model. The present study discusses such details and introduces the practical aspects of developing a robust ANN model. As a case study, predicting the flash point (FP) for an extensive dataset via a two layer feedforward neural network is investigated. The FP is one of the most important flammability properties of chemical compounds in assessment of fire hazards [29], and its predictability via ANNs has been widely studied in many works [13, 30-32].

2. Practical aspects of developing ANN models

2.1. Dataset division

The first step in developing an ANN model is to divide the dataset into three subsets, namely training, validation and test datasets. The training dataset is used to train the network, where an error function which is usually the average absolute relative error or mean squared error is minimized with respect to the weight and bias constants in successive iterations. The number of compounds needed for training, as discussed later, is the first important factor affecting the reliability of an ANN model and determines the number of neurons and layers.

As the training goes on, the performance of the ANN is continuously improved for the training dataset, which simultaneously increases the risk of overfitting as well. Overfitting causes a model to yield accurate results for the dataset used for developing that model but poor results for the compounds out of this dataset. To prevent overfitting, the performance of the studied ANN is simultaneously monitored and validated for an independent dataset which is called the validation dataset. An increase in the error function of the validation dataset in several successive iterations is an indicator of overfitting and is used as a condition to stop the training. Once a neural network is trained, i.e. the optimum values of the weights and biases are determined, the performance of the ANN is examined using another independent dataset, known as the test dataset. Usually, 60-80 % of the dataset is assigned for training, 10-20% for validation and 10-

20% to test the model.

2.2. Assigning the number of neurons and hidden layers

After specifying the training, validation and test datasets, selecting the number of layers, number of neurons in each layer, type of transfer functions and training algorithm and assigning the initial values for the weights and biases are the subsequent steps to develop an ANN model.

The number of layers and neurons in each layer is one of the most crucial parameters affecting the performance and reliability of an ANN model. With a higher number of layers or neurons, an ANN typically yields more accurate results and can model more complicated relationships. However, this increase in the number of neurons or layers can also highly increase the risk of overfitting, simultaneously.

There are some recommendations to evaluate the appropriate number of hidden layer neurons, e.g. setting a number of hidden layer neurons equal to 2/3 of the number of input layer neurons [33], between the number of neurons in the input and output layers [34], or lower than twice the number of neurons in the input layer [35]. However, such recommendations don't seem to be very robust as they totally neglect the number of training samples and details of the ANN configuration as the most important factors.

Considering an ANN model as a regression problem, the ratio of the training samples to the total number of ANN constants as suggested by Jackson for regression models [36], can be used as an index for determining the appropriate number of neurons and layers. For a multilayer feedforward neural network, if the number of input variables is V_n, the number of neurons in the hidden layer i is N_i, and the number of layers is n_L, the total number of weight and bias constants (S) can be calculated via:

$$\boldsymbol{S = V_n\, N_1 + \sum_{i=1}^{n_L-1} N_i N_{i+1} + \sum_{i=1}^{n_L} N_i}\,. \tag{2}$$

Obviously, the higher the ratio of the training samples to *S*, the higher the reliability of the results obtained by the model. It will be further discussed in section 4.2.

2.3. Assigning training algorithm and transfer function

The applied transfer functions and training algorithm are also other ANN parameters which can have impacts on the performance of an ANN [37-38]. Some commonly used transfer functions in ANN models which are also considered in our study are hyperbolic tangent sigmoid (*tansig*), log-sigmoid (*logsig*), Hard-limit (*hardlim*), Positive linear (*poslin*), and Radial basis (*radbas*) transfer functions for the hidden layers, and linear transfer function (*purelin*) for the output layer. The most widely used training algorithms are reported in table 1.

Table 1- Appropriate training algorithms in model development [39]

Training algorithm	**Abbreviation**
Levenberg-Marquardt backpropagation	*trainlm*
Gradient descent backpropagation	*traingd*
Resilient backpropagation	*trainrp*
Scaled conjugate gradient backpropagation	*trainscg*
BFGS quasi-Newton backpropagation	*trainbfg*
Conjugate gradient backpropagation with Fletcher-Reeves updates	*traincgf*
Gradient descent with momentum backpropagation	*traingdm*

2.4. Initialization of the network

Assignment of the initial values for the weight and bias constants (initialization) is the last step before we can start training of the network. Initialization is a necessary task as the efficient training algorithms, all require some initial values for the weights and biases to optimize them in successive iterations via minimizing the error function. The error function which should be minimized with respect to the weight and bias constants, typically has several local minima. As a result, starting from different initialization states determined by the initial values of the weights and biases, we may get to a quite different local minimum and obtain considerably different results. Some theoretical approaches on appropriate initialization of an ANN is reviewed by Yam and Chow [40]. An easy and yet very efficient approach to overcome the problems originated by initialization is to repeat the training of the network for various initialization states [41], which will be taken in this work to show the effect and importance of initialization and also to overcome the discussed problems resulted by initializations.

2.5. Evaluating the reliable performance of an ANN model

As discussed before and will be seen in the results, the performance of an ANN highly depends on the ANN specifications (applied dataset division, transfer function, training algorithm, initialization state and so on). Now the crucially important questions that arise are which ANN provides the most reliable model and among all different results which can be obtained via each ANN configuration using different initialization or dataset division states, which one represent the authentic performance of that configuration? One may argue that the ANN which provides the most accurate results for the overall or test dataset is the one to choose, however, such data won't be reliable. An excellent performance for the overall dataset may be affected by overfitting and for the test dataset it can be just the result of a lucky dataset division. Another option is to use the average of results obtained for all studied initialization and dataset division states for an ANN. However, average of all results is not always informative because as discussed before, due to the high number of constants in many ANN models, error function typically has several local minima and therefore, initialization plays an important role. As a result, it is quite plausible that only a few initialization states may result in an accurate model (see e.g. the results obtained for developed FP predictive ANNs with 4 neurons, reported in the supplementary material) and therefore, the average of all results won't represent the authentic performance of an ANN in most cases. In other words, a good initialization state may get lost

Table 2- The constituting functional groups used in group contribution method

	Functional Group
1	–CH_3
2	–CH_2–
3	>CH–
4	>C<
5	═CH_2
6	═CH–
7	═C<
8	═C═
9	≡CH
10	≡C–
11	–OH
12	–O–
13	>C═O
14	–CHO (aldehyde)
15	–COOH (acid)
16	–COO– (ester)
17	HCOO– (formate)
18	–NH_2
19	–NH–
20	>N–
21	═N–
22	–C≡N
23	–NO_2
24	–F
25	–Cl
26	–Br
27	–I
28	–SH
29	–S–
30	–CH_2– (ring)
31	–HC< (ring)
32	═CH– (ring)
33	>C< (ring)
34	═C< (ring)
35	–O– (ring)
36	–OH (ring)
37	>C═O (ring)
38	–NH– (ring)
39	>N– (ring)
40	═N– (ring)
41	–S– (ring)
42	-CO-O-CO- (anidride)

among several other inappropriate initialization states by averaging.

To overcome this issue, in the present study instead of averaging the results of all initialization and dataset division states, for each individual initialization state we retrain the model for several different dataset division states. The average of all obtained results are then considered as the authentic performance of that configuration. To find the most reliable model, the ANN for which in most of the repeats the observed errors of the training and test sets are not significantly different confirmed by available statistical tests, can be considered as the appropriately trained and reliable models..

3. Material and method

3.1. Dataset

To develop a robust model, reliability of the dataset used for model development plays an important role. In the present study, the DIPPR 801 [42] database was used to evaluate the predictability of the FP for an extensive dataset of pure organic compounds. DIPPR 801 provides evaluated data for several properties is one of the most widely used databases to develop FP predictive models.

To implement a GCM based model, number of presence of the functional groups listed in table 2 and the experimentally determined data of the NBP were used as the inputs of the model.

Normal boiling point and enthalpy of vaporization have been used is many FP predictive models [43-46], as both of them represent the volatility and hence, flammability of a fuel [47]. In previous studies, it was shown that considering a contribution for the NBP in addition to the same functional groups listed in table 2, can significantly improve the predictability of the FP [43, 48]. Among the organic compounds available in the DIPPR database, 740 compounds were available for which the reported data for both FP and NBP were experimentally determined. For other compounds, as a predicted data were reported for at least one of those properties, they were not considered in model development and evaluation. The full list of studied compounds can be found as supplementary material.

3.2. Initial implementation of the ANNs

To study the various parameters affecting the performance of an ANN, the predictability of the FP for the dataset of 740 pure organic compounds from diverse families was initially investigated using feedforward neural networks with 1 to 10 neurons in the hidden layer. 75% of the dataset was assigned for training, 13% for validation, and 12% to test the ANN models.

To study the impacts of dataset division on the performance of ANNs, randomly division of the dataset was repeated 20 times. To investigate the impacts of various training algorithms, transfer functions and initialization states, for each dataset division the *trainlm, traingd, trainrp, trainscg, trainbfg, traincgf,* and *traingdm* training algorithms and *tansig, logsig, hardlim, poslin,* and *radbas* transfer functions were examined for 20 different initialization states. An increase in the mean squared error of the validation dataset in 6 successive iterations was considered as the condition to stop the training.

Therefore, considering 20 different dataset division states and for each one, assigning 1 to 10 neurons for the hidden layer, 5 different transfer functions, 7 different training algorithms, and 20 different initialization states, a total number of 140000 neural networks were initially implemented and their performance for FP prediction were evaluated using a Matlab code. The performance of the ANNs were reported as percentage average absolute relative errors (*AARE%*) and correlation coefficients (R) defined as:

$$AAD = \frac{1}{N}\sum\left(\left|y_i^{ref} - y_i^{pred}\right|\right), \quad (3)$$

$$AARE\% = \frac{1}{N}\sum\left(\left|\frac{y_i^{exp} - y_i^{pred}}{y_i^{exp}}\right|\right) \times 100, \quad (4)$$

$$R = \frac{N\sum y_i^{exp} y_i^{pred} - \sum y_i^{exp}\sum y_i^{pred}}{\sqrt{N\sum (y_i^{exp})^2 - (\sum y_i^{exp})^2}\sqrt{N\sum (y_i^{pred})^2 - (\sum y_i^{pred})^2}}, \quad (5)$$

where y_i^{exp} and y_i^{pred} are the experimentally determined and predicted values of FP, respectively.

For the ANNs which resulted an overall *AARE%* of lower than 1.5%, the initially assigned constants of the network were recorded to be used for evaluation of their authentic performances as explained in section 2.5 in the next step.

3.3. Evaluation of the authentic performance of ANNs

After initially developing 140000 ANN models in the previous step, the models which resulted an *AARE%* of less than 1.5% were selected to study their reliable performance, based on the approach discussed in section 2.5. To do so, the initially assigned weight and bias constants recorded in the previous step were used to retrain the selected models for 20 different random dataset division states. The average of the results for 20 repeats were considered as the reliable performance for each configuration and initialization state. The two sample t-test method was used to compare the results obtained for the training and test datasets. The models for which the relative errors of the training and test datasets were not significantly different with 95% of significance level in at least 19 repeats, were considered as reliable models.

4. Results and discussion

4.1 Implementation of ANNs

The effect of various parameters on the performance of an ANN can be observed in the results of 140000 initially developed ANN models reported in the supplementary material. A quick overview of the initially obtained results shows that for each dataset division state changing one of the studied parameters i.e. the initialization state, number of hidden layer neurons, training algorithm or transfer function while the other parameters remain unchanged can yield considerably different results.

Among all initially studied ANNs, 17211 models yielded an overall *AARE%* of lower than 1.5% and retrained for 20 different dataset division states to evaluate their authentic performance, as discussed before. For each model, the average of 20 repeat results were calculated and are used in the next sections to evaluate the effect of each parameters. The details of the repeated results for each configuration are reported in the supplementary material.

4.2. The effect of assigned number of hidden layer neurons

As the first important factor, we consider the effect of the number of hidden layer neurons. To do so, we exploited the percentage of the initially selected ANNs with overall *AARE%* of lower than 1.5% which after retraining for 20 different dataset division states and averaging resulted an overall *AARE%* of lower than 1.5% again. Based on the results reported in table 3, a remarkable difference between the ANNs with one neuron in the hidden layer and other ANNs with higher number of neurons can be clearly observed. According to the results, 99.91% of the ANNs with 1 neuron in the hidden layer which yielded an *AARE%* of lower than 1.5% in the first step, after retraining and averaging yielded an average *AARE%* of lower than 1.5% again, while for other ANNs with higher number of neurons this value was lower than 42%. The same remarkable difference can also be observed for the average of standard deviations of *AARE%* calculated for the 20 repeats of each configuration, reported in table 3.

Table 3- The percentage of initially implemented ANNs which after retraining yielded *AARE%* of lower than 1.5% again and the average of standards deviations obtained for 20 repeat of each configuration

Nr. of hidden layer neurons	Ratio of the ANNs which reproduce the initially obtained results	Mean of average standard deviations in 20 repeats (overall)	Mean of average standard deviations in 20 repeats (training)	Mean of average standard deviations in 20 repeats (validation)	Mean of average standard deviations in 20 repeats (test)
1	99.91	0.015289	0.039167	0.13143	0.15645
2	41.35	1.1976	1.2049	1.286	1.3726
3	40.74	0.99861	1.0024	1.0813	1.1486
4	40.3	0.90107	0.90617	0.96768	1.0179
5	42.79	0.82164	0.82722	0.87261	0.91792
6	42.92	0.80169	0.80691	0.8547	0.90352
7	42.93	0.78356	0.79009	0.82401	0.86634
8	44.69	0.73719	0.74234	0.7822	0.82574
9	46.85	0.78592	0.79338	0.82743	0.85918
10	46.66	0.76154	0.76925	0.80203	0.83607

Using the average results of 20 repeats and applying the t-test method, for each number of hidden layer neurons the percentage of reliable models were calculated and are depicted in figure 2.

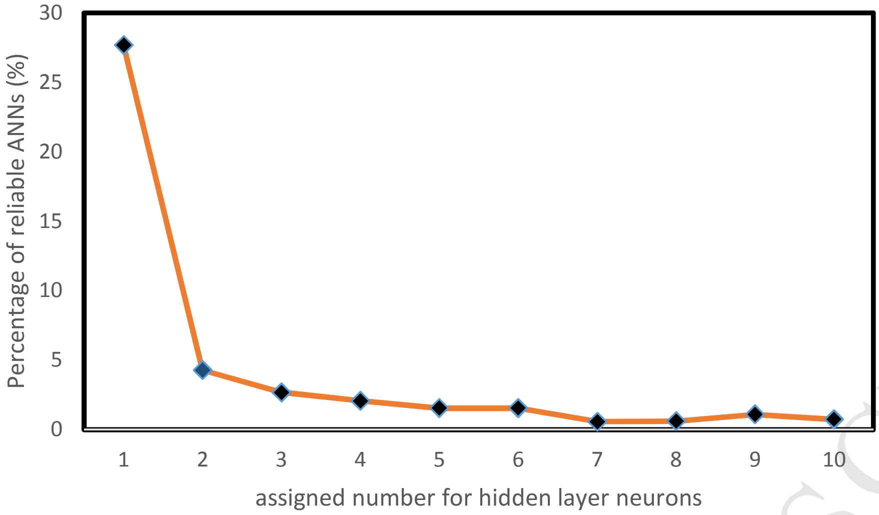


Fig 2- The percentage of reliable ANNs for each assigned number of hidden layer neurons.

According to the results, 27.66% of ANNs with 1 neuron in the hidden layer were efficiently trained, while this value for other ANNs with higher number of neurons is lower than 4.3%.

Based on the abovementioned observations, we can conclude that for our case study the optimum number of hidden layer neurons is one. Considering the number of weight and bias constants for our case study, for ANNs with one neuron in the hidden layer the ratio of the number of training samples to the ANN constants calculated using equation (2) is 12.07. This value for ANNs with 2 neurons in hidden layer is 6.1 and for higher number of neurons reduces further.

The observed ratio of the training samples to the number of constants for ANNs with one hidden layer neuron is consistent with the value recommended by Kline who suggested a value of least 10 for this ratio to avoid overfitting in regression models [49].

Therefore for developing reliable ANN models, considering the number of training samples and ANN constants, existence of at least 10 samples in the training set for each constant can be used as a rule of thumb to evaluate the efficient configuration of ANN model. In other words, considering the number of training samples, the number of layers and neurons should be selected in a way that the number of constants calculated via equation (2) is one tenth or smaller than the number of training samples. A literature survey shows many models which considerably deviate from this condition. Some of those models are reported in table 4.

Table 4- Examples of lack of sufficient data in the training dataset in ANN based models

	Number of weight and bias constants	Number of compounds in the training dataset	Ratio of the training samples to constants
Gharagheizi[50]	1481	821	0.55
Lazzús [51]	353	328	0.93
Lazzús [30]	369	350	0.95
Valderrama , et.al., [18]	421	399	0.95
Albahri [13]	274	335	1.22
Fathi et. al. [52]	316	550	1.53
Gharagheizi [53]	536	846	1.58

4.3. Impact of training algorithm and transfer function

The impact of the training algorithm and transfer function was investigated via analyzing the performance of efficiently trained models in FP prediction for each combination of training algorithm and transfer function. The results are reported in table 5 and show that different combinations of training algorithms and transfer functions can yield quite different results. More ever, for prediction of FP applying the *trainlm* training algorithm and *logsig* transfer function seems to be more efficient.

Table 5- details of reliable ANNs for each combination of transfer function and training algorithm

Training algorithm	Transfer function	Nr. reliable models	*AARE%* (overal)	*AARE %* (training)	*AARE %* (validation)	*AARE %* (test)	Training algorithm	Transfer function	Nr. reliable models	*AARE %* (overal)	*AARE %* (training)	*AARE %* (validation)	*AARE %* (test)
trainlm	logsig	70	1.57	1.57	1.57	1.58	trainbfg	logsig	51	2.08	2.06	2.12	2.12
trainlm	tansig	56	1.88	1.87	1.89	1.89	trainbfg	tansig	3	8.74	8.74	8.6	8.88
trainlm	hardlim	0					trainbfg	hardlim	0				
trainlm	poslin	73	1.61	1.61	1.62	1.61	trainbfg	poslin	46	8.01	8	8.02	8.05
trainlm	radbas	25	5.09	5.07	5.15	5.18	trainbfg	radbas	1	5.61	5.61	5.61	5.61
traingd	logsig	0					traincgf	logsig	23	4.86	4.86	4.78	4.92
traingd	tansig	0					traincgf	tansig	16	4.18	4.18	4.16	4.24
traingd	hardlim	0					traincgf	hardlim	0				
traingd	poslin	0					traincgf	poslin	80	4.84	4.84	4.81	4.88
traingd	radbas	0					traincgf	radbas	5	6.34	6.34	6.24	6.48
trainrp	logsig	6	3.58	3.57	3.62	3.59	traingdm	logsig	0				
trainrp	tansig	0					traingdm	tansig	0				
trainrp	hardlim	0					traingdm	hardlim	0				
trainrp	poslin	12	6.94	6.95	6.89	6.95	traingdm	poslin	0				
trainrp	radbas	0					traingdm	radbas	0				
trainscg	logsig	23	1.76	1.75	1.78	1.81							
trainscg	tansig	13	2.94	2.94	2.93	2.94							
trainscg	hardlim	0											
trainscg	poslin	44	3.08	3.07	3.11	3.12							
trainscg	radbas	1	1.47	1.47	1.47	1.47							

4.4. The selected ANN for FP prediction

Among the efficiently trained models, the best performance was observed for an ANN model with 1 neuron in the hidden layer, *trainlm* training algorithm and *logsig* transfer function. For this configuration the average of 20 repeats resulted an *AARE%* of 1.198, 1.194, 1.195 and 1.122 and correlation coefficients of 0.9933, 0.9933, 0.9933 and 0.9934, for the overall, training, validation and test datasets, respectively. More ever, the relative errors of the training and test datasets in none of the 20 repeats were significantly different. For this ANN, the initial values of the weight and bias constants are reported in supplementary material.

The obtained results are compared with those of the most accurate ever proposed models in table 6. As can be seen in table 6, only one model provides better results than the current model which was proposed by Albahri [13]. However, the Albahri's model is developed for a smaller set of chemicals which are transportation fuels and the diversity of the chemical families is lower than the dataset studied in the present work. Furthermore, as discussed before for the Albahri's model the ratio of training samples to constants is 1.22 which implies a high risk of overfitting in that model.

To remind the importance of the approach used to evaluate the authentic performance of ANNs, it should be noted that among the initially developed models an overall *AARE%* as low as 0.72% was also observed for an ANN with 10 neurons in the hidden layer which is resulted by overfitting.

Among initially developed ANNs, we can also find a model with 9 neurons in hidden layer for which the *AARE%* of the test and training dataset were not significantly different and yielded an overall *AARE*% as low as 0.84%. However after retraining this model using other dataset division states, we find that this model wouldn't reproduce such excellent results anymore, which implies that the initially obtained results is clearly affected by overfitting. This confirms the importance of the proposed approach for evaluating the authentic performance of each model.

Table 6- Comparison of the results of the developed model with other accurate models reported in the literature

Model	Method	No. data	AAD (k)	*AARE (%)*	Max. *AARE* (%)	R
The new model (overall)	GCM+ ANN	740	-	1.198	-	0.9933
Alibakhshi et.al. [43]	Semi- empirical	740	4.066	1.225	9.81	0.9934
Alibakhshi et. al. [48]		740	4.11	1.23	9.49	0.9935
Albahri[13]	GCM+ANN	375	3.55	1.1	6.62	0.9961
Rowley et al. [54]	Correlation (ΔH_v+NBP)	1062	4.65	1.32	–	–
Lazzús [30]	GCM+ANN+ PSO	505	6.2	1.8	8.6	–
Catoire &Naudet [55]	Correlation (ΔH_v+NBP)	600	6.36	1.84	–	–
Mathieu [56]	Correlation	92	3.75	1.37	5.4	0.9922
Pan et al. [57]		92	3.75	1.38	10.18	0.9907
Keshavarz and Ghanbarzadeh[58]	Correlation	173	6.35	2.21	12.8	0.9899
Mathieu and Alaime [59]	-	488	8.6	-	-	-
Rowley et al. [60]	Correlation (ΔH_v+NBP)	1062	9.68	2.84	–	–
Tetteh et al. [32]	QSPR+ANN	400	9.59	–	–	–
Hukkerikar et al.[61]	GC^{+}	512	10.66	3.27	–	0.89

Mathieu[44]	QSPR	230	12	–	–	0.943
Keshavarz et al. [62]	Correlation	548	12.1	–	–	–
Katritzky et al. [31]	QSPR+ANN	758	12.6	–	–	0.989
Khaje and Modarres [63]	ANFIS	95	11.5	31.1	1500	0.986
Khaje and Modarres [63]	GFA	95	13.08	25.8	966.75	0.98
Chen et. al. [64]	QSPR	230	–	–	22.9	0.964
Hshieh[65]	Correlation (NBP)	494	–	–	–	0.966
Bagheri et al. Bagheri, 2012 #33}	QSPR	1651	19.31	5.94	–	0.94
Katritzky et al. [66]	QSPR	271	–	–	–	0.91
Patil[67]	Correlation (NBP)	593	–	–	7.5	0.90

5. Conclusion

In the present study we studied various parameters which can affect the reliability and performance of ANN models. Among all affecting parameters, appropriate selection of the number of neurons and hidden layers seems to be the highest priority which should be determined based on the number of training samples. The results show that considering the ratio of the training samples to ANN constants can be used as an index for evaluating the appropriate number of neurons and layers. Our results suggest a value of greater than 10 for this ratio which has already been suggested for regression models elsewhere too. The second crucially important strategy in developing reliable ANN models is to study various initialization states for the ANNs and for each one, repeating the randomly division of the dataset and use the average of those repeats as the authentic performance obtained for each initialization state. Comparing the relative errors of training and test datasets in different repeats using appropriate statistical tests can be used to find the efficiently trained ANNs.

Considering various combinations of training algorithm and transfer functions to find the best combination of those parameters is also another efficient strategy to find more efficient ANN models.

6. Acknowledgment

The author sincerely thanks an anonymous reviewer whose careful reading and insightful comments caused a substantial improvement of the manuscript.

7. References

1. Tripp, B.; Eliasmith, C., Function approximation in inhibitory networks. *Neural Networks* **2016,** *77*, 95-106.

2. Alfaro-Ponce, M.; Argüelles, A.; Chairez, I., Pattern recognition for electroencephalographic signals based on continuous neural networks. *Neural Networks* **2016,** *79*, 88-96.

3. Hajjar, C.; Hamdan, H., Interval data clustering using self-organizing maps based on adaptive Mahalanobis distances. *Neural Networks* **2013,** *46*, 124-132.

4. Cui, Y.; Shi, J.; Wang, Z., Complex Rotation Quantum Dynamic Neural Networks (CRQDNN) using Complex Quantum Neuron (CQN): Applications to time series prediction. *Neural Networks* **2015,** *71*, 11-26.

5. da Silva, I. N.; Spatti, D. H.; Flauzino, R. A.; Liboni, L. H. B.; dos Reis Alves, S. F., *Artificial Neural Networks: A Practical Course*. Springer: 2016.

6. Acquarelli, J.; van Laarhoven, T.; Gerretzen, J.; Tran, T. N.; Buydens, L. M.; Marchiori, E., Convolutional neural networks for vibrational spectroscopic data analysis. *Analytica chimica acta* **2017,** *954*, 22-31.

7. Kuzmanovski, I.; Wagner, A.; Novič, M., Development of models for prediction of the antioxidant activity of derivatives of natural compounds. *Analytica chimica acta* **2015,** *868*, 23-35.

8. Fjodorova, N.; Novič, M., Searching for optimal setting conditions in technological processes using parametric estimation models and neural network mapping approach: A tutorial. *Analytica chimica acta* **2015,** *891*, 90-100.

9. Ni, W.; Nørgaard, L.; Mørup, M., Non-linear calibration models for near infrared spectroscopy. *Analytica chimica acta* **2014,** *813*, 1-14.

10. Ouyang, Q.; Zhao, J.; Chen, Q., Instrumental intelligent test of food sensory quality as mimic of human panel test combining multiple cross-perception sensors and data fusion. *Analytica chimica acta* **2014,** *841*, 68-76.

11. Minovski, N.; Župerl, Š.; Drgan, V.; Novič, M., Assessment of applicability domain for multivariate counter-propagation artificial neural network predictive models by minimum Euclidean distance space analysis: a case study. *Analytica chimica acta* **2013,** *759*, 28-42.

12. Hornik, K.; Stinchcombe, M.; White, H., Multilayer feedforward networks are universal approximators. *Neural networks* **1989,** *2* (5), 359-366.

13. Albahri, T. A., MNLR and ANN structural group contribution methods for predicting the flash point temperature of pure compounds in the transportation fuels range. *Process Safety and Environmental Protection* **2015,** *93*, 182-191.

14. Suzuki, T.; Ebert, R.-U.; Schüürmann, G., Application of neural networks to modeling and estimating temperature-dependent liquid viscosity of organic compounds. *Journal of chemical information and computer sciences* **2001,** *41* (3), 776-790.

15. Paduszyński, K.; Domańska, U., Viscosity of ionic liquids: an extensive database and a new group contribution model based on a feed-forward artificial neural network. *Journal of chemical information and modeling* **2014,** *54* (5), 1311-1324.

16. Hezave, A. Z.; Raeissi, S.; Lashkarbolooki, M., Estimation of thermal conductivity of ionic liquids using a perceptron neural network. *Industrial & Engineering Chemistry Research* **2012,** *51* (29), 9886-9893.

17. Paduszyński, K., In Silico Calculation of Infinite Dilution Activity Coefficients of Molecular Solutes in Ionic Liquids: Critical Review of Current Methods and New Models Based on Three Machine Learning Algorithms. *Journal of Chemical Information and Modeling* **2016,** *56* (8), 1420-1437.

18. Valderrama, J. O.; Reátegui, A.; Rojas, R. E., Density of ionic liquids using group contribution and artificial neural networks. *Industrial & Engineering Chemistry Research* **2009,** *48* (6), 3254-3259.

19. Palatinus, J. A.; Sams, C. M.; Beeston, C. M.; Carroll, F. A.; Argenton, A. B.; Quina, F. H., Kinney revisited: an improved group contribution method for the prediction of boiling points of acyclic alkanes. *Industrial & engineering chemistry research* **2006,** *45* (20), 6860-6863.

20. Lazzús, J. A., Hybrid method to predict melting points of organic compounds using group contribution+ neural network+ particle swarm algorithm. *Industrial & Engineering Chemistry Research* **2009,** *48* (18), 8760-8766.

21. Chalk, A. J.; Beck, B.; Clark, T., A quantum mechanical/neural net model for boiling points with error estimation. *Journal of chemical information and computer sciences* **2001,** *41* (2), 457-462.

22. Jiao, L.; Zhang, X.; Qin, Y.; Wang, X.; Li, H., QSPR study on the flash point of organic binary mixtures by using electrotopological state index. *Chemometrics and Intelligent Laboratory Systems* **2016**.

23. Phoon, L. Y.; Mustaffa, A. A.; Hashim, H.; Mat, R., A review of flash point prediction models for flammable liquid mixtures. *Industrial & Engineering Chemistry Research* **2014,** *53* (32), 12553-12565.

24. Patel, S. J.; Ng, D.; Mannan, M. S., QSPR flash point prediction of solvents using topological indices for application in computer aided molecular design. *Industrial & Engineering Chemistry Research* **2009,** *48* (15), 7378-7387.

25. Albahri, T. A.; Alashwak, D. A., Modeling of pure compounds surface tension using QSPR. *Fluid Phase Equilibria* **2013,** *355*, 87-91.

26. Fazeli, A.; Bagheri, M.; Ghaniyari-Benis, S.; Aslebagh, R.; Kamaloo, E., Prediction of absolute entropy of ideal gas at 298K of pure chemicals through GAMLR and FFNN. *Energy Conversion and Management* **2011,** *52* (1), 630-634.

27. Jiao, L.; Li, H., QSPR studies on the aqueous solubility of PCDD/Fs by using artificial neural network combined with stepwise regression. *Chemometrics and Intelligent Laboratory Systems* **2010,** *103* (2), 90-95.

28. Goodarzi, M.; Duchowicz, P. R.; Freitas, M. P.; Fernández, F. M., Prediction of the Hildebrand parameter of various solvents using linear and nonlinear approaches. *Fluid Phase Equilibria* **2010,** *293* (2), 130-136.

29. Alibakhshi, A.; Mirshahvalad, H.; Alibakhshi, S., Investigating the Mechanism of Effect of Carbon Nanotubes on Flame Spread Over Liquid Fuels. *Fire Technology* **2015,** *51* (4), 759-770.

30. Lazzús, J. A., Prediction of flash point temperature of organic compounds using a hybrid method of group contribution+ neural network+ particle swarm optimization. *Chinese Journal of Chemical Engineering* **2010,** *18* (5), 817-823.

31. Katritzky, A. R.; Stoyanova-Slavova, I. B.; Dobchev, D. A.; Karelson, M., QSPR modeling of flash points: An update. *Journal of Molecular Graphics and Modelling* **2007,** *26* (2), 529-536.

32. Tetteh, J.; Suzuki, T.; Metcalfe, E.; Howells, S., Quantitative structure-property relationships for the estimation of boiling point and flash point using a radial basis function neural network. *Journal of chemical information and computer sciences* **1999,** *39* (3), 491-507.

33. Boger, Z.; Guterman, H. In *Knowledge extraction from artificial neural network models,* Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on, IEEE: 1997; pp 3030-3035.

34. Blum, A., *Neural networks in C++: an object-oriented framework for building connectionist systems*. John Wiley & Sons, Inc.: 1992.

35. Berry, M. J.; Linoff, G., *Data mining techniques: for marketing, sales, and customer support*. John Wiley & Sons, Inc.: 1997.

36. Jackson, D. L., Revisiting sample size and number of parameter estimates: Some support for the N: q hypothesis. *Structural equation modeling* **2003,** *10* (1), 128-141.

37. Yonaba, H.; Anctil, F.; Fortin, V., Comparing sigmoid transfer functions for neural network multistep ahead streamflow forecasting. *Journal of Hydrologic Engineering* **2010,** *15* (4), 275-283.

38. Karlik, B.; Olgac, A. V., Performance analysis of various activation functions in generalized MLP architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems* **2011,** *1* (4), 111-122.

39. Demuth, H.; Beale, M., Neural network toolbox for use with MATLAB. **1993**.

40. Yam, J. Y.; Chow, T. W., A weight initialization method for improving training speed in feedforward neural network. *Neurocomputing* **2000,** *30* (1), 219-232.

41. MacLeod, C., An introduction to practical neural networks and genetic algorithms for engineers and scientists. *Jerhi Wahyu Fernanda* **2013,** *157*.

42. American Institute of Chemical Engineers (AIChE), E. P. D. D., Public Release, Documentation, Design Institute for Physical Properties (DIPPR), Project 801. **2016**.

43. Alibakhshi, A.; Mirshahvalad, H.; Alibakhshi, S., Prediction of flash points of pure organic compounds: Evaluation of the DIPPR database. *Process Safety and Environmental Protection* **2017,** *105*, 127-133.

44. Mathieu, D., Flash points of organosilicon compounds: how data for alkanes combined with custom additive fragments can expedite the development of predictive models. *Industrial & Engineering Chemistry Research* **2012,** *51* (43), 14309-14315.

45. Carroll, F. A.; Lin, C.-Y.; Quina, F. H., Simple method to evaluate and to predict flash points of organic compounds. *Industrial & Engineering Chemistry Research* **2011,** *50* (8), 4796-4800.

46. Riazi, M.; Daubert, T., Predicting flash and pour points. *Hydrocarbon processing* **1987,** *66* (9), 81-83.

47. Alibakhshi, A., Enthalpy of vaporization, its temperature dependence and correlation with surface tension: a theoretical approach. *Fluid Phase Equilibria* **2017,** *432*, 62-69.

48. Alibakhshi, A.; Mirshahvaland, H.; Alibakhshi, S., A Modified Group Contribution Method for Accurate Prediction of Flash Points of Pure Organic Compounds. *Industrial & Engineering Chemistry Research* **2015,** *54* (44), 11230-11235.

49. Kline, R. B., *Principles and practice of structural equation modeling*. Guilford publications: 2015.

50. Gharagheizi, F., An accurate model for prediction of autoignition temperature of pure compounds. *Journal of hazardous materials* **2011,** *189* (1), 211-221.

51. Lazzús, J. A., Prediction of flammability limit temperatures from molecular structures using a neural network–particle swarm algorithm. *Journal of the Taiwan Institute of Chemical Engineers* **2011,** *42* (3), 447-453.

52. Fatehi, M.-R.; Raeissi, S.; Mowla, D., Estimation of viscosity of binary mixtures of ionic liquids and solvents using an artificial neural network based on the structure groups of the ionic liquid. *Fluid Phase Equilibria* **2014,** *364*, 88-94.

53. Gharagheizi, F., A new group contribution-based model for estimation of lower flammability limit of pure compounds. *Journal of hazardous materials* **2009,** *170* (2), 595-604.

54. Rowley, J. R.; Rowley, R. L.; Wilding, W. V., Prediction of pure-component flash points for organic compounds. *Fire and Materials* **2011,** *35* (6), 343-351.

55. Catoire, L.; Naudet, V., A unique equation to estimate flash points of selected pure liquids application to the correction of probably erroneous flash point values. *Journal of physical and chemical reference data* **2004,** *33* (4), 1083-1111.

56. Mathieu, D., Inductive modeling of physico-chemical properties: Flash point of alkanes. *Journal of hazardous materials* **2010,** *179* (1), 1161-1164.

57. Pan, Y.; Jiang, J.; Wang, Z., Quantitative structure–property relationship studies for predicting flash points of alkanes using group bond contribution method with back-propagation neural network. *Journal of hazardous materials* **2007,** *147* (1), 424-430.

58. Keshavarz, M. H.; Ghanbarzadeh, M., Simple method for reliable predicting flash points of unsaturated hydrocarbons. *Journal of hazardous materials* **2011,** *193*, 335-341.

59. Mathieu, D.; Alaime, T., Insight into the contribution of individual functional groups to the flash point of organic compounds. *Journal of hazardous materials* **2014,** *267*, 169-174.

60. Rowley, J.; Rowley, R.; Wilding, W., Estimation of the flash point of pure organic chemicals from structural contributions. *Process Safety Progress* **2010,** *29* (4), 353-358.

61. Hukkerikar, A. S.; Kalakul, S.; Sarup, B.; Young, D. M.; Sin, G. r.; Gani, R., Estimation of environment-related properties of chemicals for design of sustainable processes: development of group-contribution+ (GC+) property models and uncertainty analysis. *Journal of chemical information and modeling* **2012,** *52* (11), 2823-2839.

62. Keshavarz, M. H.; Jafari, M.; Kamalvand, M.; Karami, A.; Keshavarz, Z.; Zamani, A.; Rajaee, S., A simple and reliable method for prediction of flash point of alcohols based on their elemental composition and structural parameters. *Process Safety and Environmental Protection* **2016,** *102*, 1-8.

63. Khajeh, A.; Modarress, H., QSPR prediction of flash point of esters by means of GFA and ANFIS. *Journal of hazardous materials* **2010,** *179* (1), 715-720.

64. Chen, C.-C.; Liaw, H.-J.; Tsai, Y.-J., Prediction of flash point of organosilicon compounds using quantitative structure property relationship approach. *Industrial & Engineering Chemistry Research* **2010,** *49* (24), 12702-12708.

65. Hshieh, F. Y., Correlation of closed-cup flash points with normal boiling points for silicone and general organic compounds. *Fire and Materials* **1997,** *21* (6), 277-282.

66. Katritzky, A. R.; Petrukhin, R.; Jain, R.; Karelson, M., QSPR analysis of flash points. *Journal of chemical information and computer sciences* **2001,** *41* (6), 1521-1530.

67. Patil, G., Estimation of flash point. *Fire and materials* **1988,** *12* (3), 127-131.

initial weights (input to hidden layer)	initial weight (hidden to out	initial bias (hidden layer)	initial bias (output layer)
0.25664109	0.753820576	0	0.600984171
-0.089856267			
-0.026214469			
0.233793939			
0.000730538			
-0.132677763			
0.204736078			
-0.111603444			
0.171586573			
0.217845522			
0.158490122			
0.085816088			
0.107618943			
-0.048025702			
-0.251767873			
-0.249378562			
0.19979295			
0.220617871			
-0.321063932			
0.150536464			
0.201334458			
-0.239071437			
0.340050043			
-0.224779078			
0.295832302			
0.164472357			
0.112653827			
-0.03721664			
0.210485864			
0.345716969			
-0.270950582			
-0.052974562			
-0.285945606			
-0.186141492			
-0.044006425			
-0.257247693			
-0.337605767			
0.134659311			
-0.214740052			
-0.282797827			
-0.070716288			
-0.328017535			
0.328173127			

- The present study introduces the important practical aspects of developing a reliable artificial neural network (ANN) model including appropriate assignment of the number of neurons, number of hidden layers, transfer functions, training algorithm, dataset division and initialization of the network.

- Predictability of the flash point for a dataset of 740 organic compounds is studied for a total number of 484220 different ANNs to allow covering a wide range of parameters affecting the performance of an ANN.

- As a rule of thumb for evaluating the appropriate number of neurons and layers, it is suggested to select the number of neurons and layers in a way that the ratio of the training samples to the ANN constants be equal or greater than 10.

- A strategy for evaluation of the authentic performance of an ANN model is proposed which suggests repeating the training of each configuration with same initially assigned weight and bias constants several times for different dataset division states and using the average of all repeats as the authentic performance of that initialization and configuration.

- The models retrained for similar initialization and configuration but different data set division states for which the observed errors of the training and test sets are not significantly different in most of the repeats confirmed by statistical tests (two sample t-test in the current work) can be considered as the appropriately trained models and their results can be used reliably.

- Based on the introduced considerations, an ANN model is proposed for predicting the flash point of pure organic compounds from diverse chemical families. According to the results obtained for 740 compounds, the new model was found to produce the lowest errors for flash point prediction compared to other available models.