Development and comparison of Extreme Learning machine and multi-layer perceptron neural network models for predicting optimum coagulant dosage for water treatment

Jayaweera CD, Aziz N. *  
School of Chemical Engineering, Engineering Campus, Universiti Sains Malaysia, Seri Ampangan, 14300 Nibong Tebal, Seberang Perai Selatan, Penang, Malaysia.

chamanthidj@gmail.com, chnaziz@usm.my*

Abstract. Artificial neural networks have been extensively used in modelling the coagulation process in water treatment. Multi-layer Perceptron neural networks (MLP) are commonly used for coagulation modelling. However, a major drawback of MLPs is the high computational effort required due to its iterative nature. The Extreme learning machine neural network has a prediction accuracy comparable to MLPs and consumes far less computational effort. In this study an ELM-single layer feedforward neural network (ELM-SLFN), ELM-radial basis neural network (ELM-RBF) and a MLP was developed to predict the optimum coagulant dosage. All neural networks performed well with correlation coefficients exceeding 0.97. However, the ELM-RBF network performed better than the MLP model with higher prediction accuracy. Thus ELM-RBF neural network is a more efficient model for prediction of coagulant dosage for water treatment.

1. Introduction
Coagulation is a mandatory process employed in water treatment primarily for the reduction of turbidity and natural organic matter. The dosage of coagulant required is conventionally determined by carrying out jar tests, which consume time and chemicals. Jar tests are not suitable for responding to rapid changes of water qualities. Studies have been carried out to develop neural network models to predict the optimum coagulant dosage for water treatment [1-6]. Multi-layer Perceptron neural networks (MLP ANNs) are commonly used for the modelling of coagulation process for water treatment. The major drawback of multi-layer perceptron neural networks is the time consumed for training. The multiplicity of parameters to be optimized in MLP ANNs makes the modelling procedure is more time consuming. Other types of neural networks such as General Regression Neural Networks and Radial Basis Neural Networks have been developed [7-8] and compared with the MLP architecture. Although the use of the said networks is more convenient than MLPs, their test performance is inferior to MLPs.

A more efficient type of neural networks named ‘Extreme Learning Machine’ was introduced by [9] in 2004. ELMs are Single Layer Feed forward neural networks (SLFN) which consume a fraction of the time consumed by MLPs. [10] developed an ELM-SLFN model to predict optimum coagulant dosage which performed well as the MLP model, but with far less computational effort.
1.1 ELM-SLFN

In an ELM-SLFN, weights connecting the inputs to the hidden layer and hidden layer biases are not optimized, but randomly generated. The weights connecting the hidden layer are estimated analytically using the Moore Penrose inverse. The mathematical model for SLFNs with an output layer - linear transfer function is given by equation 1.

$$\sum_{i=1}^{n} \beta_i g(w_i x_j + b_i) = T$$  \hspace{1cm} (1)

Where \(n\) – Number of hidden neurons, \(x\) – Input vector, \(w\) – Input weight vector, \(T\) – Output vector, \(\beta\) – Output layer transfer function, \(g(x)\) – Hidden layer transfer function

In an ideal situation, where the number of hidden neurons (\(n\)) is equal to the number of training samples (\(N\)), assuming there exist \(\beta\) and \(g(x)\) such that equation 1 could be represented in matrix form as given in equation 2.

$$H \beta = T$$  \hspace{1cm} (2)

Where, \(H\) = 

\[
\begin{bmatrix}
g(w_1 x_1 + b_1) & g(w_2 x_1 + b_2) & \cdots & g(w_n x_1 + b_n) \\
g(w_1 x_N + b_1) & g(w_2 x_N + b_2) & \cdots & g(w_n x_N + b_n)
\end{bmatrix}
\]

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}, \quad T = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix}$$

Since the number of hidden neurons \(n \ll N\), the optimization of parameters is carried out as given by equation 3.

$$\|H - \beta T\| = \min \|H - \beta T\|$$  \hspace{1cm} (3)

[9] suggested using the smallest norm least square solution to calculate output weights using equation 4, instead of using gradient descent methods.

$$\hat{\beta} = H^\dagger T$$  \hspace{1cm} (4)

Where \(H^\dagger\) is Moore Penrose inverse

The only parameter that requires adjusting is the number of hidden neurons.

1.2 ELM-RBF

In 2004, [11] suggested an extension of the ELM-SLFN case to radial basis function neural networks. The mathematical model of a RBF is given by equation 5.

$$\sum_{i=1}^{n} \beta_i \varphi_i(x) = T_j$$  \hspace{1cm} (5)

Where \(\varphi_i(x) = \exp \left( \frac{\|x - \mu_i\|^2}{\sigma_i^2} \right)\)

\(\mu_i\) \(^{\text{th}}\) radial basis center, \(\sigma_i\) \(^{\text{th}}\) smoothing parameter

If all \(N\) training vectors are used as RBF kernels, equation 5 could be represented by the matrix form given by equation 6.

$$H \beta = T$$  \hspace{1cm} (6)
Where 
\[ H = \begin{bmatrix} \varphi_1(x_1, \mu_1, \sigma_1) & \varphi_2(x_1, \mu_2, \sigma_2) & \ldots & \varphi(x_1, \mu_n, \sigma_n) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(x_N, \mu_1, \sigma_1) & \varphi(x_N, \mu_2, \sigma_2) & \ldots & \varphi(x_N, \mu_n, \sigma_n) \end{bmatrix} \]

\[ \beta \] and \( T \) are similar to the SLFN case.

As \( n<<N \), the optimization is carried out by equation 7.

\[ \|H - \beta T\| = \min \|H - \beta T\| \]

Thus, as in the case of SLFN, [11] suggests using equation 4. For ELM-RBF model development, RBF kernels and smoothing parameters are randomly generated values. The only parameter that requires adjusting is the number of radial basis kernels. In their study, they have observed that the generalization behaviour of ELM-RBF on a number of datasets was similar to support vector machines.

Network architectures of MLP and RBF networks are shown in Figs 1 and 2. MLP is a supervised network whereas RBF is an unsupervised network.

Although MLP neural networks have been successfully used in prediction of coagulant modelling, they require adjusting multiple parameters with the possibility of obtaining solutions corresponding to a local minimum. On the other hand, ELM-RBF neural networks produce results of high prediction accuracy with only having to adjust the number of radial basis kernels. However, the application of ELM-RBF on coagulation modelling for water treatment has yet to be explored. The objective of this study is to develop ELM-RBF, ELM-SLFN and commonly used MLP neural network models to predict the optimum coagulant dosage for water treatment and compare their performance and efficiency.

Figure 1 - Multi-Layer perceptron network architecture [12]
2. Methodology
Matlab 2017 was used for the development of the ANN models, with data provided by the Segama Water treatment plant corresponding to the year 2005, constituting over 8200 data points. The ranges of values of water qualities are as shown in table 1. The coagulant used by the Segama water treatment plant is Alum. Most part of the source river runs through vegetation and is exposed to less industrial activity.

The data set was divided into training (76%) and testing (24%), for the development of ELM-SLFN and ELM-RBF models. The hidden layer activation function used for the ELM-SLFN model is the tansigmoid. The performance of the ELM-RBF model was tested for the radial basis function kernel and the gaussian kernel. The optimum number of hidden neurons (for the ELM-SLFN) and radial basis centers (for the ELM-RBF) was found by trial and error, based on the performance (MSE and correlation coefficient) of the model.

The data set was divided for training (60%), validation (16%) and testing (24%), for the development of the MLP model. The network was trained using the Levenberg-Marquardt algorithm. The MLP contained a single hidden layer with tansigmoid as the activation function. A linear activation function was employed for the output layer. The number of hidden neurons was determined using the hypothesis employed by [6] to use a maximum of 2I+1 neurons, where I is the number of input variables. The optimum number of hidden neurons was determined by varying the number of neurons from I to 2I+1 and testing the performance of the model in each case.

Table 1 - Water qualities

| Raw water     | Treated water     |
|---------------|-------------------|
|               | pH | Turb (NTU) | Col (HU) | TDS (mg/l) | Alkalinity (mg/l) | pH | Turb (NTU) | Color (HU) | TDS (mg/l) | Alkalinity (mg/l) | Al residual (mg/l) | Alum dosage |
| Min           | 6.6 | 11 | 36 | 60 | 50 | 6.5 | 0.19 | 0 | 90 | 20 | 0.01 | 20 |
| Max           | 7.9 | 3405 | 656 | 160 | 170 | 7.5 | 5.24 | 6 | 170 | 100 | 0.2 | 170 |
Step by step forms of the algorithms used in the three models are as follows.

**ELM-SLFN:**
1. Record CPU time
2. Set parameters (number of hidden neurons, number of variables) and generate random input weights and biases
3. Generate the Hessian matrix according to equation 2.
4. Calculate the output weight matrix according to equation 4.
5. Simulate training data
6. Calculate MSE and Correlation coefficient (R)
7. If performance is not satisfactory, return to step 2. Else, record CPU time and calculate total time.

**ELM-RBF:**
1. Record CPU time
2. Set parameters (number of hidden neurons, number of variables) and generate random input weights and smoothing parameters
3. Generate the Hessian matrix according to equation 6.
4. Calculate the output weight matrix according to equation 4.
5. Simulate training data
6. Calculate MSE and Correlation coefficient (R)
7. If performance is not satisfactory, return to step 2. Else, record CPU time and calculate total time.

The MLP network was modeled using Matlab neural networking toolbox. The procedure for determining the optimum model is as follows.

1. Generate neural network
2. Set parameters
3. Initialize weights and record initial weights
4. Train neural network
5. Record performance and training time
6. Repeat from step 3 onwards till the number of initial weights equal the number of tries required by the former networks
7. Choose best set of initial weights

Data normalization was carried out using equation 8.

\[
x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}
\]  

(8)

Input parameter selection was carried out using an exhaustive search algorithm.

3. **Results and Discussion**

Input parameters selected for use in all models are Turbidity, Color and Al (t-1). The optimum number of hidden neurons and radial basis centers in ELM-SLFN and ELM-RBF was found to be 20 in both cases. The optimum architecture of the MLP model consisted of 7 hidden neurons.

The performance of the ELM models highly depend on the weights/centers randomly generated. Therefore the performance (eg. Correlation coefficient) of the models fluctuated at each trial between 0.8-0.97. The fluctuation of the performance of the ELM-RBF model with each trial, was higher than the ELM-SLFN model. [11] have recommended using the Gaussian kernel for the ELM-RBF. However, in this study, it was noted that the said fluctuation was higher with the use of the Gaussian kernel than with the Radial Basis Function kernel, while the latter tend to produce better results. Therefore the RBF kernel was used to model the final ELM-RBF model. Table 2 represents the performance of the models developed.

As shown in Table 2, the ELM-RBF model has the highest correlation coefficient and the lowest MSE, thus, outperforming the other models. The comparison of results of ELM-SLFN model and the MLP model is consistent with the results of [10], where MLP has slightly outperformed the ELM-SLFN model. The cause of ELM-SLFN’s higher MSE could be noted in figure 4.
Table 2 - Performance of models predicting optimum coagulant dosage

|          | ELM-RBF | ELM-SLFN | MLP   |
|----------|---------|----------|-------|
| Test     | Correlation coefficient | 0.9752   | 0.9733 | 0.9740 |
|          | Mean Squared error      | 4.56×10⁻⁴ | 4.94×10⁻⁴ | 4.83×10⁻⁴ |
| Train    | Correlation coefficient | 0.9680   | 0.9733 | 0.9676 |
|          | Mean Squared error      | 7.52×10⁻⁴ | 7.53×10⁻⁴ | 7.6×10⁻⁴ |
| Time (s) | Train               | 0.7-0.9  | 0.6-0.7 | 5 – 20 |
|          | Selection of optimum initial weights | 11.65 (20 sets of input weights) | 9.05 (20 sets of input weights) | Approximately 200 (20 sets of initial weights) |

ELM-SLFN has consumed the least amount of time for training the model. However, the difference between the time taken by ELM-SLFN and ELM-RBF to train is negligible. In comparison to the ELM models, the time taken by the MLP model is significantly high.

The models’ performance when simulated with test data was higher than train data, as all test data consisted of inliers.

Figures 3-5 demonstrate plots of predicted vs actual values of coagulant dosage for each of the models. When comparing the data distribution of the ELM-SLFN plot (fig 4) with the MLP (fig 3) and the ELM-RBF (fig 5), it could be noted that from 0 to 0.25, the points in fig 2 have clustered closer to the y=x line than in other two plots. The distribution of data points between 0.25 to 0.6 are similar in all three plots. According to the regression plots, ELM-SLFN has performed well. However, a few points may have caused the decrease in the relative performance, as demonstrated in figure 6. Such points that have been distributed astray could be noted between 0.1 and 0.3. The distribution of the results of ELM-RBF plot (fig 5) is remarkably similar to the MLP (fig 3) model. However it could be noted that distribution of data points above 0.6 are closer to the y=x line in the ELM models (fig 4&5) than in the MLP model (fig 3). Therefore the ELM models have performed better than the MLP.

Figure 3 - Regression plot of MLP test results (y-predicted, x- actual)

Figure 4 - Regression plot of ELM-SLFN test results (y-predicted, x- actual)
Figure 5 - Regression plot of ELM-RBF test results (y-predicted, x-actual)

Figure 6 - Predicted vs. actual regression plots of results of ELM-SLFN, ELM-RBF and MLP models

As observed by the regression plots, the performance of all three models is remarkably similar. However, there are merits and drawbacks inherent to each model. A comparison of advantages and disadvantages of each model is given in Table 3.

Table 3 - Advantages and disadvantages of the three models

|                             | MLP   | ELM-SLFN | ELM-RBF |
|-----------------------------|-------|----------|---------|
| Computational time          | High  | Low      | Low     |
| Variability of performance depending on initial | Low   | High     | High    |

The performance of ELM models depended greatly on the initial input weights. It was also noted that the data preprocessing technique affected the performance of ELM models. Normalization of data enhanced the performance of models. The presence of outliers affected the performance of ELM-RBF significantly. The inability of ELM-RBF model to extrapolate was due to its exponentially decaying localized non-linearities. Therefore the ELM-RBF model developed is strictly applicable to a specified range of values (as shown in Table 1), while MLP and ELM-SLFN could tolerate some degree of extrapolation. However, a satisfactory performance of all three models within the given limits of operation could be guaranteed.

4. Conclusion

In this study, three models were developed for the prediction of optimum coagulant dosage in water treatment plants: A Multi-Layer Perceptron model, A single layer feed forward extreme learning and a Radial Basis Function-extreme learning machine neural network. The ELM-RBF model outperformed all the other models demonstrating excellent generalization. The ELM models performed well while
consuming lesser computational effort compared to the MLP model. All three models performed satisfactorily with correlation coefficients exceeding 0.97.

Acknowledgement
The cooperation of Sabah Water Supply Department and LDWS for supplying the Segama Water Treatment Plant data is greatly acknowledged.

References
[1]. F. N. Valentin, "An Hybrid Neural Network Based System for Optimization of Coagulant Dosing in a Water Treatment Plant", Citeseerx.ist.psu.edu, 1999.
[2]. K. Griffiths and R. Andrews, "The application of artificial neural networks for the optimization of coagulant dosage", Water Science & Technology: Water Supply, vol. 11, no. 5, p. 605, 2011.
[3]. G. Wu and S. Lo, "Effects of data normalization and inherent-factor on decision of optimal coagulant dosage in water treatment by artificial neural network", Expert Systems with Applications, vol. 37, no. 7, pp. 4974-4983, 2010.
[4]. A. Robenson, S. Shukor and N. Aziz, "Development of Process Inverse Neural Network Model to Determine the Required Alum Dosage at Segama Water Treatment Plant Sabah, Malaysia", 2009.
[5]. G. Wu and S. Lo, "Predicting real-time coagulant dosage in water treatment by artificial neural networks and adaptive network-based fuzzy inference system", Engineering Applications of Artificial Intelligence, vol. 21, no. 8, pp. 1189-1195, 2008.
[6]. H. Maier, "Use of artificial neural networks for predicting optimal alum doses and treated water quality parameters", Environmental Modelling & Software, vol. 19, no. 5, pp. 485-494, 2004.
[7]. M. Kennedy, A. Gandomi and C. Miller, "Coagulation modeling using artificial neural networks to predict both turbidity and DOM-PARAFAC component removal", Journal of Environmental Chemical Engineering, vol. 3, no. 4, pp. 2829-2838, 2015.
[8]. C. Kim and M. Parnichkun, "MLP, ANFIS, and GRNN based real-time coagulant dosage determination and accuracy comparison using full-scale data of a water treatment plant", Journal of Water Supply: Research and Technology - Aqua, vol. 66, no. 1, pp. 49-61, 2016.
[9]. G. Huang, Q. Zhu and C. Siew, "Extreme learning machine: Theory and applications", Neurocomputing, vol. 70, no. 1-3, pp. 489-501, 2006.
[10]. X. Deng and C. Lin, "Application of ELM to predict the coagulant dosing in water treatment plant", 2016.
[11]. "Extreme learning machine: RBF network case - IEEE Conference Publication", Ieeexplore.ieee.org, 2004. [Online]. Available: http://ieeexplore.ieee.org/document/1468985/.
[12]. H. Demuth, M. Beale and M. Hagan, Neural network toolbox. Natick, Mass.: MathWorks, 2007.
[13]. "Extreme learning machine: a new learning scheme of feedforward neural networks - IEEE Conference Publication", Ieeexplore.ieee.org, 2004. [Online]. Available: http://ieeexplore.ieee.org/document/1380068/.