Uncertainty Analysis of Developed ANN in Prediction of Daily Electrical Conductivity

Nhat Minh Megan Nguyen

Maryam Zeinolabedini Rezaabd

Abstract

One of the critical water quality indicators is Electrical Conductivity (EC) parameter. It is useful in mineralization and Total Dissolved Solids (TDS) estimation. This project aims to model daily EC values in Barwon River, Australia, by using Artificial Neural Network (ANN). For this purpose, three observed data including Turbidity, PH, and Temperature were used as predictors and observed EC values as predictand. In order to obtain the best structure, trial and error procedure was used and different number of nodes, hidden layers with different activation functions were constructed. Afterward, bootstrapping method and Mont Carlo (MC) were employed respectively for parameter estimation and uncertainty analysis. The performance of best structures for test stage was 0.82 $\mu s/cm$ and 0.8 in terms of RMSE and R, respectively. It was found that the ANN was not able to estimate EC values accurately. Moreover, the uncertainty extent of each selected model for first 10 test data was estimated. Results showed that the extent of uncertainty of both selected models was low, with less than 20% of observed data lying within 10^{th} and 90^{th} percentile of MC outputs.

Introduction and Related Work

Water quality is influenced by rapid urbanization and economic development. Therefore, water quality deterioration has become a global concern.

Water quality grade can be evaluated by various parameters among which Electrical Conductivity (EC) is a critical one[1]. EC measurements are a simple and fast way for Total Dissolved Solids (TDS), salinity, and electrolytes accessibility estimation in water. Although EC is a measurable parameter, its direct measurement is time-consuming and costly[2].

Water quality models are vigorous tools to assess pollution, plan aquatic projects, and manage the best practices for conserving water[3,4]. Currently, different models including conventional methods (i.e., linear models) and Neural Network (NN)-based methods have been employed for water quality prediction[5]. As real systems, like water quality patterns, are complicated, non-linear models are required to deal with the complexity. The restricted ability of conventional methods in accurate prediction of water quality highlights the usefulness of NN-based models[6]. In comparison with conventional methods, NN technique has a potential to learn and find the hidden relationships in observed data[4]. The inclusive feature of NN in massively parallel processing makes it the most common method in processing and handling sophisticated computations[7].

The performance and efficiency of NN-based methods have been studied in various water bodies in current studies[5]. However, the uncertainty of these methods is infrequently considered by researchers[8]. In water quality modelling, the uncertainty which is related to data variation and model parameters is not avoidable[9]. With growth of parameter numbers, the computational costs and model complexity will increase. Moreover, the model reliability and accuracy will decrease if the data accessibility and model complexity are not balanced[10]. Hence, uncertainty analysis of NN needs more attention to evaluate the model reliability.

There are two main techniques for uncertainty analysis: 'local' and 'global'[11,12]. Although the local approach is simpler than global approach and requires less computational cost, it is not able to consider whole effect of model parameters uncertainty on its output. Global approach uses probability density function for uncertainty analysis and thus, is more reliable.

A Monte Carlo (MC) simulation, as a global approach, is an impressive and novel method for uncertainty analysis. Recently, MC is the most widely used for uncertainty analysis[13] in water field.

Yullianti et al.[14] investigated the sensitivity of management policies to inputs parameters, for erosion control. Aqil et al.[15] evaluated the uncertainty in two NN-base models' outputs to predict weekly streamflow in the river by using MC.

Noori et al studied the efficiency of MC for uncertainty propagation in prediction of solid waste generation forecasting[16] and daily carbon monoxide by using Artificial Neural Network (ANN) and Adaptive Neuro-Fuzzy Inference System (ANFIS)[17]. Godo-pla et al.[18] performed parameter estimation and uncertainty analysis to evaluate the reliability of the ANN for potassium permanganate demand prediction.

According to the above-mentioned explanations of the importance of EC in water quality and NN-based models' capability, the objective of this study is evaluating the impact of uncertainty of model parameters on EC prediction. For this purpose, a model by using ANN is developed for EC prediction. Next, model parameters are estimated by bootstrapping technique, and the reliability of proposed model is assessed by MC. More details of used methods in this project will be explained in Methodology section.

Materials and Methods

Case study

The Barwon River is in the north-west slopes and Orana regions of New South Wales, Australia. Barwon River is a part of the Murray-Darling basin and has a constant stream throughout the year. Some of characteristics of Barwon River and the selected station are shown in Table 1. Figure 1 shows the geographical location of used station.

Table 1. Barwon River features.

River Length	700 km
River Source	Macyntire River
Station name	Barwon-Geelong
Station number	233217
Station Latitude	-38.16
Station Longitude	144.35
Data Owner	VIC - Department of Environment, Land, Water and Planning

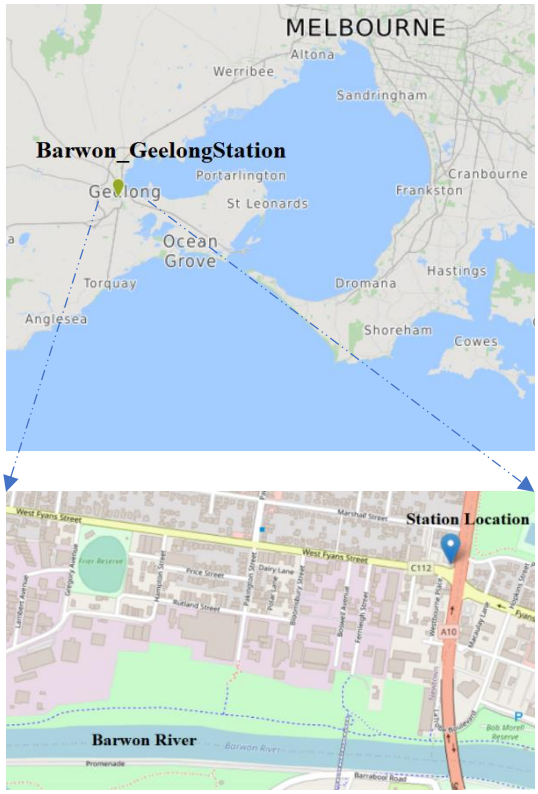

Figure 1. Geographical location of Barwon-Geelong station.

In this project, water quality data (2010-2021) related to Barwon-Geelong Station were used. For this project, daily observed dataset including Electrical Conductivity (*EC*), *PH*, Temperature (T_e), and Turbidity (T_u) were selected in which EC is dependent variable and others are independent variables. *PH* is a measure of the water acidity and is based specifically on its hydrogen ion concentration. As ions carry positive or negative charges, EC will occur. Temperature

affects EC by increasing ionic mobility as well as the solubility of many salts and minerals. Turbidity can increase water temperature because suspended particles absorb more heat. Table 2 shows some statistical features of dataset.

Table 2. Statistical features of observed variables.

Variable	*PH*	T_e	T_u	*EC*
Unit	-	℃	*NTU**	$\mu s/cm$
Min	6.73	8.4	-1.7	320
Max	12.54	27	405.3	3643
Mean	7.78	16.32	26.36	1930.7
Std	0.49	4.96	29.35	731.83
Correlation	0.06	0.47	-0.60	1
**NTU*: Nephelometric Turbidity Units				
*$\mu s/cm$:Micro siemens per centimeter				

Preprocessing: outlier treatment

In observed dataset, there were individual values falling outside of the overall pattern of a data set. These values, which affect the generalization of the data and model, are called outliers. In this project, first, both dependent (output) and independent (inputs) variables were standardized by using Z-score transformation (Eq.1)[19].

$$x_{i,norm} = \frac{x_i - x_{mean}}{x_{std}} \qquad \text{Eq.1}$$

Next, standardized values whose absolute values exceeded 1.96, representing around 5% of total dataset, were eliminated.

Methodology

ANN for water systems modeling

ANN, as a soft computing technique, is useful for dealing with inaccurate, nonlinear and complex data. It has high precision in modeling and recognizing pattern by estimation of optimum weights and biases[20]. More details of ANN structure are explained by Jiang et al.[13].

PyTorch has become the preferrable deep learning framework among both academic and industrial researchers. The main advantage of this PyTorch is that it provides a flexible and programmatic runtime interface that facilitates the construction and modification of systems by connecting operations. In PyTorch, by importing Torch, many convenient modules are provided. For instance, torch.nn, torch.optim are to create and train our neural networks based on the training dataset. NN.Sequential is a module that contains other modules and applies them in sequence to generate output. A PyTorch implementation of an ANN looks exactly like a NumPy implementation. The first step is parameter initialization. In this purpose, tensors, as the base data structures of PyTorch, are used for building different types of ANN. Next, ANN must be defined and trained in four key steps as follows:

Forward Propagation: Computation of activation function at every layer (with j nodes) using two below steps:

$$u_j = \sum_{i=1}^{n} x_i * w_{ij} + bias \qquad \text{Eq.2}$$

$$\hat{y}_j = f(u_j) \qquad \text{Eq.3}$$

Where $\hat{y}$ is outputs estimated by model at node j, n is number of inputs, and f is the activation function.

Loss computation: Computation of the difference between the observed (y) and the predicted ($\hat{y}$) values. The loss function is computed as follows:

$$l(y, \hat{y}) = L = \{l_1, \ldots., l_N\}^T \qquad \text{Eq.4}$$

Where N is the batch size.

When torch.nn is implemented, the $l(y, \hat{y})$ will be defined differently based on reduction default (Eq.5). In this project, it was defined based on mean.

$$l(y, \hat{y}) = \text{mean(L)} \qquad \text{Eq.5}$$

Back Propagation: Minimize the error in the output layer by making marginal changes in the bias and the weights. These marginal

changes are computed using the derivatives of the error term.

Parameters Update: Update weights and bias using the delta changes received from the above backpropagation step.

Finally, when these steps are executed for several epochs with training dataset, the loss is reduced to a minimum value. The final weight and bias values are obtained which can then be used to make predictions on the unseen (test) dataset.

Bootstrap method and parameter estimation

In modeling, residual (e_i) is the difference between target and predicted values by model, i.e., Eq.6[18]:

$$e_i = y_i - \hat{y}_i \quad \text{Eq.6}$$

and, the optimal parameter values can be obtained by minimizing the sum of squared residuals, which is called least squares method and is shown in Eq.7:

$$\theta = argmin \sum_{i=1}^{n} {e_i}^2 \quad \text{Eq.7}$$

Where, θ is the unknow model parameter set, y_i is targets and $\hat{y}_i$ is the model outputs (predicted values).

For parameter estimation, a statistical/ computational method, called Bootstrap, is applied. If the underlying distribution of parameters is not determined, this method can be practical to evaluate model parameters. Some studies such as[18,21–23] proved the capability of bootstrap in quantifying uncertainty for prediction using NN-based models. The bootstrap steps are indicated as follows[24]:

1. ANN is trained with input and target values. Model parameters, predicted values and sets of residuals are obtained.
2. New residual subsets are generated by random sampling and replacement of original residual set, and then added to model prediction obtained in Step 1 to create synthetic data.
3. Step 2 is repeated for N_{BT}, which is the number of bootstrapping samples.
4. Parameter estimation is repeated by using Step 1 and synthetic dataset obtained in Step 2. The re-estimated model parameters are saved in a Matrix ($\hat{\theta}$) with $N_{BT} \times p$ dimension, where p is the number of re-estimated parameters in each bootstrap resampling.
5. The distribution function, mean and covariance of the parameter estimators can be estimated by using ($\hat{\theta}$) matrix. The $cov(\hat{\theta})$ and $\mu(\hat{\theta})$ functions are defined as Eq.8 and Eq.7, respectively.

$$cov(\hat{\theta}) = \frac{\frac{\min \sum (y_i - \hat{y}_i)^2}{\sigma^2}}{n-p} (F' \cdot F)^{-1} \quad \text{Eq.8}$$

where, F is Jacobian matrix: $\frac{\partial \hat{y}}{\partial \hat{\theta}}$

$$\mu(\hat{\theta}) = \frac{1}{n} \sum_{i=1}^{n} {e_i}^2 \quad \text{Eq.9}$$

Monte Carlo (MC) and uncertainty analysis

To analyze uncertainty of parameter estimation, which is effective on model outputs uncertainty, MC method is employed. Through estimated mean and covariance functions in the last section, distribution (which is multivariate normal distribution in this project) can be assumed for parameter sets. Afterwards, following steps are followed to analyze the parameter uncertainty on model outputs[24]:

1. Sample N_{MC} possible parameter sets from assumed multivariate normal distribution. N_{MC} is indicating the number of MC samples and the dimension of sampling matrix is $N_{BT} \times p$. p is the number of parameters.
2. Compute ANN model for each parameter set.
3. Estimate integral of interest (i.e., mean and variance of model outputs). If f is a

function of θ, where $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)$ is the parameter vector, and p is the number of parameters, the multidimensional integration operation is given by Eq.10 and 11:

$$E = \frac{1}{N_{MC}} \sum_I^{N_{MC}} f(X_{N_{MC}}) \quad \text{Eq.10}$$

$$I = \lim_{N_{MC}\to\infty} \frac{1}{N_{MC}} \sum_I^{N_{MC}} f(X_{N_{MC}}) \quad \text{Eq.11}$$

E is the MC estimate, and it is ensured to converge to true values of I based on law of large numbers. However, if N_{MC} is finite, there will be an error in MC integration.

4. Estimate the average of MC integration error as following:

$$MCerror = \frac{\sigma(f)}{\sqrt{N_{MC}}} \quad \text{Eq.12}$$

$\sigma(f)$ is the error standard deviation and is estimated by Eq.13.

$$\sigma(f)^2 \sim \frac{1}{N_{MC}-1} \sum_I^{N_{MC}} (f(X_{N_{MC}} - E)^2 \quad \text{Eq.13}$$

5. Estimate the uncertainty in predicted outputs $\hat{y}$: Given the uncertainty in the vector of $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)$ characterized by the respective distribution functions $D = (D_1, \dots, D_p)$, the uncertainty in predicted outputs $\hat{y}$ is expressed as follow:

$$var(\hat{y}) = \int (\hat{y} - f(\theta))^2 \, d\theta \quad \text{Eq.14}$$

$$E(\hat{y}) = \int f(\theta) d\theta \quad \text{Eq.15}$$

$var(\hat{y})$ and $E(\hat{y})$ are the variance and expected value of $\hat{y}$ which is obtained by MC sampling.

6. Compute the 90% confidence interval by using $var(\hat{y})$ and $E(\hat{y})$ as below:

$$\hat{y}_{0.90} = \hat{y} \pm t_{N-p}^{0.05} \sqrt{diag(\, var(\hat{y}))} \quad \text{Eq.16}$$

where, $cov(\hat{y}) = (F \cdot F) cov(\hat{\theta}) (F \cdot F)^{-1}$ Eq.17

Results and Discussion

ANN development and parameter estimation

This project was carried out on quality parameters from Barwon River of Australia to predict *EC* by using ANN. Input dataset included T_e, T_u, and *PH* and output contained EC values. For model development, observed data must be divided to training and testing dataset [20] . In this project, training and testing dataset were 75% and 25% respectively.

An ANN model constructed for predicting daily EC values. The number of iterations was 200 (code submitted was run on 100 epochs to save running time), and the performance function was MSE. The significant issue in ANN learning is overfitting prevention. Larger portion of data for training can prevent model from overfitting. However, using too many nodes in model construction may be one of the reasons for overfitting[25]. In this project, in order to find the optimum structure for *EC* prediction, different structures were built by changing the number of hidden layers, number of nodes and type of activation function. Activation functions used in this project were "Linear", "Tanh", "Sigmoid", "ReLU", and "Hardtanh".

Table 3 and 4 show some of the results for each structure in training and testing stage respectively. For comparing adequacy of developed structures, some statistical criteria such as Root mean square error (RMSE), Mean absolute percentage error (MAE) and Coefficient of correlation (R)[2,15] were used.

Table 3. performance of different structures in training stage.

No.	Layer Nodes	Activation Function	RMSE	MAE	R
1	1	Linear	0.82	0.68	0.74
2	4	Linear	0.82	0.68	0.74
3	10	Linear	0.82	0.68	0.74
4	4	ReLU	0.82	0.68	0.78
5	4	Sigmoid	0.82	0.68	0.75

6	10	Sigmoid	0.82	0.68	0.76
7	10	Hardtanh	0.81	0.67	0.79
8	4	Tanh	0.81	0.67	0.76
9	10	Tanh	0.82	0.68	0.75
10	10 5	Tanh Tanh	0.81	0.67	0.80
11	30 10	Tanh Hardtanh	0.81	0.67	0.80
12	10 **5**	Tanh ReLU	0.82	0.68	0.79
13	15 10 5	Tanh Tanh Tanh	0.82	0.67	0.79
14	15 10 5	Sigmoid Tanh Hardtan	0.82	0.67	0.79

Table 4. performance of different structures in testing stage.

No.	Layer Nodes	Activation Function	RMSE	MAE	R
1	1	Linear	0.82	0.69	0.75
2	4	Linear	0.83	0.70	0.75
3	10	Linear	0.83	0.69	0.74
4	4	ReLU	0.83	0.69	0.77
5	4	Sigmoid	0.83	0.69	0.75
6	10	Sigmoid	0.82	0.69	0.76
7	10	Hardtanh	0.82	0.68	0.79
8	4	Tanh	0.82	0.68	0.76
9	10	Tanh	0.82	0.68	0.73
10	10 5	Tanh Tanh	0.83	0.69	0.77
11	30 10	Tanh Hardtanh	0.84	0.70	0.78
12	**10** **5**	**Tanh** **ReLU**	**0.82**	**0.68**	**0.80**
13	**15** **10** **5**	**Tanh** **Tanh** **Tanh**	**0.82**	**0.68**	**0.80**
14	15 10 5	Sigmoid Tanh Hardtan	0.83	0.69	0.76

Comparison of table 3 and 4 results indicates that error values are in the same range. In other words, type of activation function or number of hidden layers and nodes could not be effective in improving results significantly. As testing data are unseen for model, making decision for selecting the best structure was done based on test results. Thus, both structures 12 and 13 were selected as the best ones. ½

Results show that the best performance was obtained from 2 hidden layer with 10 and 5 nodes using "Tanh" and "ReLU" activation functions and 3 hidden layers with 15, 10 and 5 nodes using "Tanh" activation function. One of the best structures (structure 12) is shown in Figure 2.

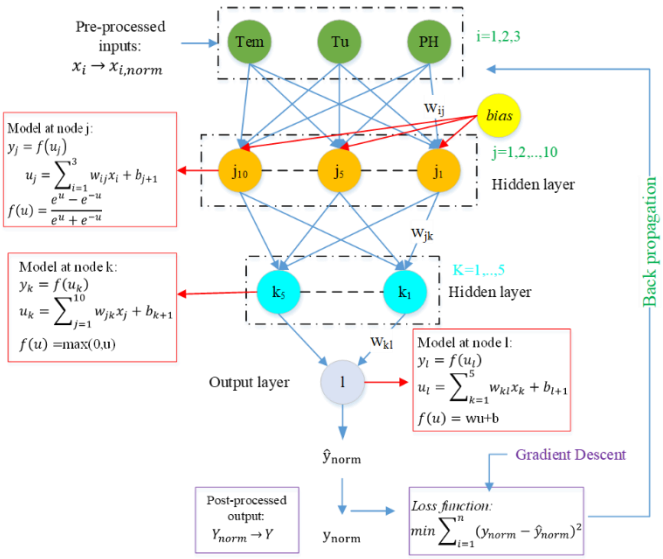

Figure 2. Details of the ANN with structure 12.

Observed and predicted values of structure 12 were compared visually in Figure 3 and 4.

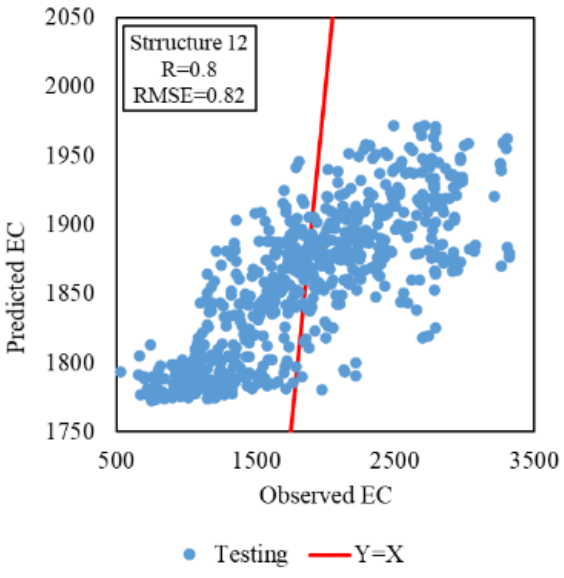

Figure 3. Scatter plot for unstandardized data in testing stage relative to best fit (Y=X).

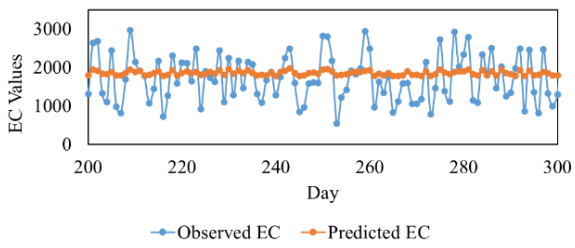

Figure 4. Comparison of differences between unstandardized predicted and observed EC values, in testing stage.

Figure 3 depicts the scatterplot of observed and predicted EC values. As it is obvious in figure 3, most of data points do not coincide with the best fit line. A low correlation between observed and predicted values may be due to inconsistent nature of water quality variables as they have been measured over a long period (11 years) [25].

Figure 4 compares predicted values relative to observed data for just 100 days. Results show that ANN was not able to model the variation of EC values precisely. All EC values were underestimated relative to real ones.

The selected structures (12 and 13) were candidates to further exploration through uncertainty analysis. In this purpose, first, bootstrap method was applied on selected structures. Details of bootstrapping method are depicted in Figure 5.

In this project, the number of bootstrapping samples (N_{BT}) was 120 and 300 for structure 12 and 13, respectively. After applying bootstrapping, mean, standard deviation and correlation matrix of the parameter estimators could be estimated from covariance matrix. As a result, the distribution function of parameter estimators was obtained.

Parameter identifiability and uncertainty analysis

To infer precisely, the parameters need to be identifiable. In other words, for different set of weights the output values follow different distribution. However, generally, neural networks are not identifiable. It can be proved by a simple linear example. Imagine, there is an ANN structure, like that presented in Figure 2, but with only one hidden layer and linear function. For linear -single layer neural network with k nodes in hidden layer, the model output at node k will be equal to Eq.18.

$$\hat{y}_k = \sum_{k=1}^{K} w_{kl} \sum_{i=1}^{I} w_{ik} x_i + b_{k+1} \quad \text{Eq.18}$$

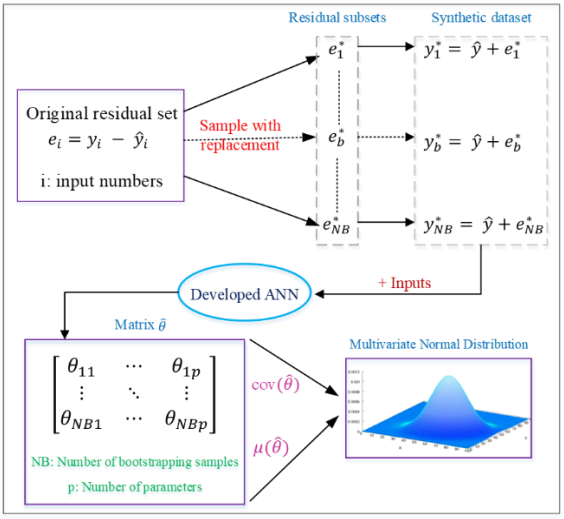

Figure 5. Details of bootstrapping method.

Where, i is number of inputs, k is number of nodes in hidden layer, l is the node of output layer, w_{kl} is the hidden-to-output weights, and w_{ik} is the input-to-hidden weights.

As there are many number of possible configurations for which two pairs of matrices w_{kl} and w_{ik} have the same product, the elements of the weight matrices are not identifiable in general.

The same reasoning can be used to infer that non-linear neural networks are also non-identified. The structure in Figure 2, is non-identified. Since, for any loss value, a permutation of the node weights at one layer, and their corresponding nodes at the next layer, will also result in the same loss values[26]. Hence, when MC simulation was performed with the ANN, both the mean and covariance matrix of the parameters should

be considered[18]. The outputs of MC simulation show the interquartile range of the ANN predictions.

Based on the mean and covariance matrix of parameter sets generated from bootstrapping, 100 different sets of parameters were randomly sampled from multivariate normal distribution. These 100 sets of parameters were fitted to both selected models. In this project, box plot was utilized to represent visually MC results with fitted parameter sets. Figure 6 demonstrates details of a box plot. Box plot visually shows the distribution of data by using data quartiles (or percentiles).

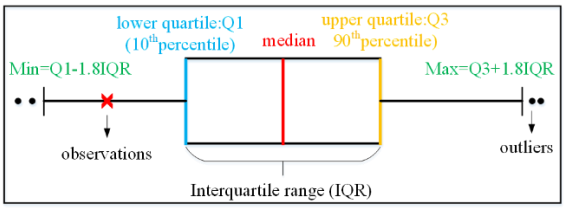

Figure 6. Details of a box plot.

Box plot displays the five-number summary of a dataset, including the minimum score, first (lower) quartile, median, third (upper) quartile, and maximum score.

The most important point about box plot is that the larger spread of MC outputs on the box plots, the higher extent of uncertainty in parameter estimation.

Figures 7 and 8 illustrate the model outputs for structures 12 and 13 respectively, with first 10 test data. Figures display 10th and 90th percentile and mean value of MC simulations to estimate prediction quality.

Results of figure 7 indicate that the uncertainty extent for structure 12 is relatively low, possibly because of high correlations among model parameters.

In structure 13, the increase in model parameters and the reduced likelihood of model non-identifiability expanded the uncertainty bounds and captured observations better. However, out of 10 boxes demonstrated for both suggested structures, only 20% of the observations lied within the 10th and 90th percentile of MC outputs. Such a low percentage, which will decrease even further as more simulations are plotted, is due to narrow uncertainty band.

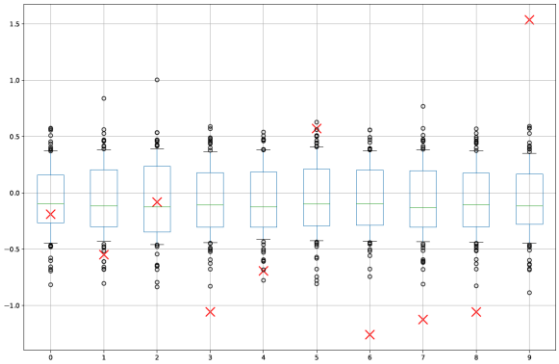

Figure 7. Uncertainty of EC prediction with structure 12.

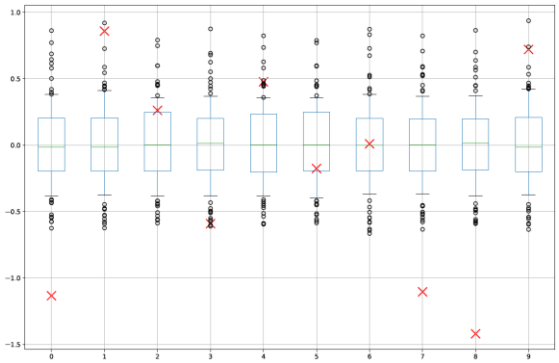

Figure 8. Uncertainty of EC prediction with structure 13.

Conclusion

In this project, different structures were developed by using ANN for EC values prediction. The main aim of this project was evaluating the reliability of ANN in EC prediction. In this purpose, two structures with better accuracy compared to others in testing stage was selected for uncertainty analysis. Results showed that the increasing number of nodes expanded the model's extent of uncertainty and boosted model's precision by capturing more observations. However, complicated structures are computationally expensive. This problem

proves to be more prominent in parameter estimation and uncertainty analysis. For instance, structure12 (with 10 and 5 nodes in hidden layers) contained $3 \times 10 + 10 + 10 \times 5 + 5 + 5 \times 1 + 1 = 101$ parametrs. The number of bootstrapping samples needs to exceed the number of parameters to avoid singular matrix error when multivariate normal distribution of bootstrapped sets of parameters is projected for MC simulations. 120 bootstrapped simulations (> 101) were performed, which took an average of 45 seconds per simulation. Once the structure is complicated, the number of bootstrapped simulations and their running time will increase exponentially. Moreover, there was an increased chance of overfitting as structure got more complicated. For further improvements on this combined method, cross-validation should be performed, and dropout of nodes could be used to mitigate the overfitting problem. Markov-chain Monte Carlo (MCMC) instead of stochastic gradient descent should be implemented directly on the neural network (assuming Gaussian priors) to update parameters for more effective predictive models.

GitHub link

References

1. Ekemen Keskin, T., Özler, E., Şander, E., Düğenci, M. & Ahmed, M. Y. Prediction of electrical conductivity using ANN and MLR: a case study from Turkey. *Acta Geophysica* **68**, 811–820 (2020).

2. Ghorbani, M. A., Aalami, M. T. & Naghipour, L. Use of artificial neural networks for electrical conductivity modeling in Asi River. *Applied Water Science* **7**, 1761–1772 (2017).

3. Jia, H., Xu, T., Liang, S., Zhao, P. & Xu, C. Bayesian framework of parameter sensitivity, uncertainty, and identifiability analysis in complex water quality models. *Environmental Modelling and Software* **104**, 13–26 (2018).

4. Najah, A., El-Shafie, A., Karim, O. A. & El-Shafie, A. H. Application of artificial neural networks for water quality prediction. *Neural Computing and Applications* **22**, 187–201 (2013).

5. Singha, S., Pasupuleti, S., Singha, S. S., Singh, R. & Kumar, S. Prediction of groundwater quality using efficient machine learning technique. *Chemosphere* **276**, 130265 (2021).

6. Rajaee, T., Khani, S. & Ravansalar, M. Artificial intelligence-based single and hybrid models for prediction of water quality in rivers: A review. *Chemometrics and Intelligent Laboratory Systems* vol. 200 103978 (2020).

7. Sarkar, A. & Pandey, P. River Water Quality Modelling Using Artificial Neural Network Technique. *Aquatic Procedia* **4**, 1070–1077 (2015).

8. Wu, W., Dandy, G. C. & Maier, H. R. Protocol for developing ANN models and its application to the assessment of the quality of the ANN model development process in drinking water quality modelling. *Environmental Modelling and Software* vol. 54 108–127 (2014).

9. Franceschini, S. & Tsai, C. W. Assessment of uncertainty sources in water quality modeling in the Niagara River. *Advances in Water Resources* **33**, 493–503 (2010).

10. Freni, G. & Mannina, G. Uncertainty estimation of a complex water quality model: The influence of Box-Cox transformation on Bayesian approaches and comparison with a non-Bayesian method. *Physics and Chemistry of the Earth* **42–44**, 31–41 (2012).

11. Vandenberghe, V., Bauwens, W. & Vanrolleghem, P. A. Evaluation of uncertainty propagation into river water quality predictions to guide future monitoring campaigns. *Environmental Modelling and Software* **22**, 725–732 (2007).

12. Reder, K., Alcamo, J. & Flörke, M. A sensitivity and uncertainty analysis of a continental-scale water quality model of pathogen pollution in African rivers. *Ecological Modelling* **351**, 129–139 (2017).

13. Jiang, Y., Nan, Z. & Yang, S. Risk assessment of water quality using Monte Carlo simulation and artificial neural network method. *Journal of Environmental Management* **122**, 130–136 (2013).

14. Yulianti, J. S., Lence, B. J., Johnson, G. v. & Takyi, A. K. Non-point source water quality management under input information uncertainty. *Journal of Environmental Management* **55**, 199–217 (1999).

15. Aqil, M., Kita, I., Yano, A. & Nishiyama, S. Analysis and prediction of flow from local source in a river basin using a Neuro-fuzzy modeling tool. *Journal of Environmental Management* **85**, 215–223 (2007).

16. Noori, R., Abdoli, M. A., Farokhnia, A. & Abbasi, M. Results uncertainty of solid waste generation forecasting by hybrid of wavelet transform-ANFIS and wavelet transform-neural network. *Expert Systems with Applications* **36**, 9991–9999 (2009).

17. Noori, R., Hoshyaripour, G., Ashrafi, K. & Araabi, B. N. Uncertainty analysis of developed ANN and ANFIS models in prediction of carbon monoxide daily concentration. *Atmospheric Environment* **44**, 476–482 (2010).

18. Godo-Pla, L. *et al.* Predicting the oxidant demand in full-scale drinking water treatment using an artificial neural network: Uncertainty and sensitivity analysis. *Process Safety and Environmental Protection* **125**, 317–327 (2019).

19. Olden, J. D. & Jackson, D. A. Illuminating the "black box": A randomization approach for understanding variable contributions in artificial neural networks. *Ecological Modelling* **154**, 135–150 (2002).

20. Dawood, T., Elwakil, E., Novoa, H. M. & Gárate Delgado, J. F. Toward urban sustainability and clean potable water: Prediction of water quality via artificial neural networks. *Journal of Cleaner Production* **291**, 125266 (2021).

21. Jia, Y. & Culver, T. B. Bootstrapped artificial neural networks for synthetic flow generation with a small data sample. *Journal of Hydrology* **331**, 580–590 (2006).

22. Tiwari, M. K. & Chatterjee, C. A new wavelet-bootstrap-ANN hybrid model for daily discharge forecasting. *Journal of Hydroinformatics* **13**, 500–519 (2011).

23. Sharma, S. K. & Tiwari, K. N. Bootstrap based artificial neural network (BANN) analysis for hierarchical prediction of monthly runoff in Upper Damodar Valley Catchment. *Journal of Hydrology* **374**, 209–222 (2009).

24. Sin, G. & Gernaey, K. Data Handling and Parameter Estimation. in *Experimental Methods in Wastewater Treatment* (eds. van Loosdrecht, M. C. M., Nielsen, P. H., Lopez-Vazquez, C. M. & Brdjanovic, D.) 201–234 (IWA Publishing, 2016).

25. Ravansalar, M. & Rajaee, T. Evaluation of wavelet performance via an ANN-based electrical conductivity prediction model. *Environmental Monitoring and Assessment* doi:10.1007/s10661-015-4590-7.

26. Ran, Z. Y. & Hu, B. G. Parameter identifiability in statistical machine learning: A review. *Neural Computation* vol. 29 1151–1203 (2017).