Hindawi
Journal of Environmental and Public Health
Volume 2022, Article ID 7104752, 15 pages
https://doi.org/10.1155/2022/7104752

Research Article

Multiple Statistical Model Ensemble Predictions of Residual Chlorine in Drinking Water: Applications of Various Deep Learning and Machine Learning Algorithms

Charles Onyutha

Department of Civil and Environmental Engineering, Kyambogo University, P.O. Box 1, Kyambogo, Kampala, Uganda

Correspondence should be addressed to Charles Onyutha; conyutha@kyu.ac.ug

Received 25 July 2022; Accepted 7 September 2022; Published 28 September 2022

Academic Editor: Giovanna Deiana

Applicability of statistical models in predicting chlorine decay remains minimally explored. This study predicted residual chlorine using six deep learning and nine machine learning techniques. Suitability of multimodel ensembles (MMEs) including arithmetic mean of all the models (Ens1), average of the best three performing models (Ens2), and weighted mean of outputs from all the 15 models was investigated. A total of nine "goodness-of-fit" measures (such as distance correlation (r_d) and Taylor skill score) were used to rank the models. The two best deep learning methods were the nonlinear autoregressive model with exogenous input (NARX) ($r_d = 0.51$) and feedforward backpropagation (FFB) ($r_d = 0.61$). The two best machine learning algorithms included random forests (RF) ($r_d = 0.64$) and Gaussian process regression (GPR) ($r_d = 0.59$). Eventually, Ens2 was obtained using RF, FFB, and GPR. Ens2 performed better than Ens1 and Ens3. The amount of variance explained by individual models and MMEs was over the ranges of 13–66% and 51–74%, respectively. Ens2 explained 74% of the total variance in observed residual chlorine. Remarkably, the appropriateness of the MMEs depends on the approach for combining model outputs, and the number of models considered. This study demonstrated the acceptability of statistical MMEs in predicting chlorine residual concentration in drinking water.

1. Introduction

Several drinking water disinfectants exist [1]. Examples of such disinfectants include chlorine dioxide, ozone, ultraviolet light, chloramine, and free chlorine [2]. Free chlorine has several advantages; it is efficacious in disinfection, easy to apply, cheap, and long-lasting, thereby disinfecting water up to the consumer points [1]. Drinking water is recommended to have chlorine residual concentrations in the range 0.2–5 mg/l [3]. Drinking water with concentrations of residual chlorine below 0.2 mg/l tends to be susceptible to regrowth of microbials which infect consumers. On the other hand, overdosage of chlorine can lead to the formation of harmful byproducts such as chloro-organics, haloacetic acids, and trihalomethanes (THM). THM chloroform is known to be carcinogenic to animals [4] and human beings. Other health complications which can result from overdosage of chlorine in drinking water include birth defects and damages to liver and kidney [5]. In places where water distribution networks are ineffectively managed and when treatment plants are outdated or not regularly maintained, careful activities should be designed for monitoring of the system to guard against possible health issues which may arise, for instance, from byproducts of chlorine following overdosage of the chlorine as a disinfectant [6].

For analysis of how to keep concentrations of residual chlorine in drinking water within the recommended range 0.2–5 mg/l, modelling tends to be conducted. Both physical and statistical models exist for modelling of chlorine decay. Examples of process-based and statistical models include EPANET [7] and artificial neural network, respectively. A recent review showed that 87% and 17% of the studies on modelling chlorine residuals in drinking water applied process-based and statistical models, respectively [2]. Physical models are characterized by scale effects. Furthermore, nontrivial mathematics and several assumptions are

required in capturing the behavior of chlorine concentrations under different circumstances of temperature, pH, and other factors. In cases where predictions of residual chlorine concentration (*RCC*) are required given the little available time, data-driven statistical models can be used.

As statistical methods, both machine learning and deep learning techniques can be applied for making predictions of chlorine residual concentrations. Deep learning makes use of deep neural networks. Machine learning comprises the use of computer to train and automate tasks which would turn out to be unbearable or impossible for human beings to perform. Essentially, deep learning is a subset of machine learning. However, both deep learning and machine learning are subsets of artificial intelligence. Worth noting is that the performance of deep learning can differ, to some extent, from that of the machine learning.

Tackling the problem of uncertainties on the modelling outcomes can be undertaken by combining the outputs from various models to obtain one set of modelled results. By combining many realizations from (i) one deterministic model or (ii) various models but of the same structure, we obtain single model ensemble. On the other hand, multimodel ensemble can be obtained from combination of outputs from models of different structural complexity. The application of the concept of multimodel ensemble has been common in hydrological modelling (see, e.g., [8–10]) but not for prediction of chlorine residual in drinking water.

By the time of conducting this research, no any studies could be found in literature to have conducted an in-depth analysis of the prediction of RCC using an array of both machine learning and deep learning techniques while investigating the suitability of multimodel ensemble. Thus, this study is aimed at addressing this knowledge gap in scientific research.

2. Materials and Methods

2.1. Methods

2.1.1. Deep Learning

(i) Nonlinear Autoregressive Model with Exogenous Input (NARX)

To model sequential data, recurrent neural networks (RNNs) can be applied. This is because RNN eliminates the need for having many parameters since it considers each element within the given sequence to be assigned the same weight, something which is not the case for other models such as the deep feedforward model. In an RNN, the exhibition of the temporal dynamic behavior can be linked to the connections among the nodes. The input variable length sequence can be processed based on the internal memory.

NARX [11] is one of the commonly applied RNNs. NARX differs from other networks through its passing of information from one step to another, and it does this by adding loops which feed the preceding inputs and outputs back to the network. NARX also has the typical feature of characterizing lag time of the input and output through the feedback delays [12].

Consider n_x and n_y to denote the number of input and output delays, respectively, such that the actual values of the exogenous inputs include $x(t+1), x(t), \cdots, x(t-n_x)$. Furthermore, let the actual values of the time series be $y(t)$, $y(t-1), \cdots, y(t-n_y)$. If we represent the past estimated or predicted outputs from the NARX model as series $\widehat{y}(t), \widehat{y}(t-1), \cdots, \widehat{y}(t-n_y)$, the typical architectures of NARX [13, 14] can be given by

$$\widehat{y}(t+1) = f\big(y(t), y(t-1), \cdots, y\big(t-n_y\big), x(t+1), x(t), \cdots, x(t-n_x)\big), \tag{1}$$

$$\widehat{y}(t+1) = f\big(\widehat{y}(t), \widehat{y}(t-1), \cdots, \widehat{y}\big(t-n_y\big), x(t+1), x(t), \cdots, x(t-n_x)\big), \tag{2}$$

where $\widehat{y}(t+1)$ denotes the predicted output of NARX at the time $(t+1)$, and f represents the mapping function. Equation (1) represents the series-parallel (also called open-loop) architecture of NARX. On the other hand, Equation (2) indicates the parallel (also called closed loop) architecture of NARX.

(ii) Long Short-Term Memory (LSTM) Model

LSTM model [15] is another commonly applied RNN. LSTM which is used to effectively capture long-term temporal dependencies has a memory cell as its fundamental structure. The memory cell comprises cell states to remember temporal information. Thus, the memory cell is to remember and propagate unit outputs at various time steps. Flow of information from one time step to another is controlled by the forget gate, input gate, and output gate. Initially, a typical LSTM block comprised the input gate and output gate as well as the cells [15] and the forget gate was not included. Later, the forget gate was introduced into the LSTM architecture by Gers et al. [16], thereby allowing the LSTM to reset its own state.

Consider that i_t, o_t, and f_t denote the input gate, output gate, and forget gate, respectively. Furthermore, let h_t, c_t^v, and c_t represent the hidden state, candidate vector, and cell state, respectively. Let us also take b_i, b_f, b_o, and b_c to be the bias for the input gate, forget gate, output gate, and the new cell, respectively. If $x(t)$ denotes the input vector at time step t while h_{t-1} is the hidden state of the previous time step, the general memory block of the LSTM can be given by $i_t = \sigma(w_i[x(t), h_{t-1}] + b_i)\, f_t = \sigma(w_f[h_{t-1}, x(t)] + b_f)$, $o_t = \sigma(w_o[h_{t-1}, x(t)] + b_o), c_t^v = \sigma(w_c[h_{t-1}, x(t)] + b_c)$, $c_t = f \times c_{t-1} + i_t \times c_t^v$, and $h_t = o_t \times \tanh(c_t)$ where σ and tanh are activation functions.

(iii) Layer Recurrent (LR) Neural Network

LR is another RRN model and was introduced by Xie et al. [17] following the inspiration by the ResNet architecture [18]. LR can adaptively learn contextual information. In the LR architecture, the first step is to compute the local features using the convolutional neural network (CNN)

module. In the second step, two 1D spatial RNNs are applied to scan along each of the rows. Thirdly, the two 1D spatial RNNs are applied to scan along each column from two directions [17]. In the LR architecture (see [17] for details), dependencies are captured by the within layer recurrence.

(iv) Feedforward Neural Networks

The main goal of a feedforward neural network (FFN) is to approximate some function. The components of an FFN network include input layer, hidden layer, output layer, and neurons' weights. In FFN, connections among the nodes are not in a circular form. Nodes within the network are used to connect the various units. Information about the network depends on the weight of the connections. The flow of information through the nodes of FFN is in a forward direction. At the neuron output, we have an activation function which serves as the decision-making center.

After entering the network via the input point, the data passes through each and every layer of the network before obtaining the outputs. At first, the set of inputs are multiplied by their weights after entering through the input layer. The second step consists of summing up the weighted inputs such that we can have the output as 1 (if the sum of the value exceeds a specified threshold) or -1 (when the value is exceeded by the limit). The third step comprises classification and here, concepts of machine learning can be applied for the classification. In the fourth step, outputs can be compared with the predicted values. At this point, the training procedure ensures weights are optimized to enhance accuracy. The last step consists of backpropagation in which the weights are updated especially in a multilayered networks and hence, the name feedforward backpropagation (FFB) neural network. Instead of backpropagation, we can have a tap delay line associated with each input and weights and in this line, we talk about the feedforward distributed time delay (FFDT) neural network.

(v) Adaptive-Network-Based Fuzzy Inference System (ANFIS)

ANFIS is some kind of artificial neural network [19]. It integrates neural networks and fuzzy logic. The learning capability of ANFIS in approximating nonlinear functions can be linked to the set of fuzzy if–then rules.

ANFIS applies a hybrid learning rule to combine the least squares method and backpropagation gradient descent [19] in identifying an array of parameters for generating the previously obtained input-output pairs. The neural fuzzy control system used for the ANFIS was based on the Takagi-Sugeno-Kang fuzzy rules in the form

$$R_1 : \text{If } x_1 \text{ is } C_1^1 \text{ and } x_2 \text{ is } C_1^2, \text{ then } y = f_1 = c_0^1 + c_1^1 x_1 + c_2^1 x_2,$$
$$R_2 : \text{If } x_1 \text{ is } C_1^2 \text{ and } x_2 \text{ is } C_1^2, \text{ then } y = f_2 = c_0^2 + c_1^2 x_1 + c_2^2 x_2, \quad (3)$$

and for the two inputs, x_1 and x_2, the inferred output $y^{\#} = (\gamma_1 f_1 + \gamma_2 f_2)/(\gamma_1 + \gamma_2)$ where γ_j denotes the firing strengths of $R_j (j = 1, 2)$ and can be computed using $\gamma_j = \gamma_{C_1^j}(x_1) \times \gamma_{C_2^j}(x_2)$.

We can consider ANFIS to have 5 layers. The first layer is actually an adaptive node comprising a node function. In the second layer, all the incoming signals are multiplied to obtain an output. In layer 3, the ratio of the i^{th} rule's firing strength to the sum of all the various rules' firing strength is computed and in this layer we consider each node to be fixed. In layer 4, each node is taken to be an adaptive node and comprises a node function. It is in layer 5 that we compute the overall output by adding all the incoming signals and to do so, each node is considered fixed.

2.1.2. Machine Learning

(i) Randomforest (RF) Regression

RF developed by Breiman [20] comprises a methodology in which results from decision trees are averaged to obtain a better outcome. Here, the decision trees can be thought of in terms of a number of weak heterogeneous individual learners who are trained. Given a complex problem, we believe that these learners (as a group) can make better decision than (expert) individuals. Thus, the decision tree represents an outcome obtained by combining the results of the learners in a manner, for instance, through a voting scheme to ensure the overall result typifies an enhanced yield.

Recursive partitioning is applied to establish a regression tree. Bagging (which is the process of creating training data through random sampling with replacement) is used to minimize the diversity of trees. The out-of-bag (or the subset of trees which are not selected as training data in the bagging process) are used for checking model's performance.

Consider N to be the number of samples while M denotes the attributes or the number of input features. Furthermore, let ψ be the total number of trees to be grown to become a forest. Bootstrap sampling is performed for each tree in a forest to come up with N samples. In the sampling procedure, samples are selected randomly and this is done with replacement. To grow a new decision tree for every training set, the CART method of Breiman et al. [21] can be used. It is from the node that a new split can occur. To do so, we consider only q number of features such that q is less than M, for instance, $q = \sqrt{M}$. We repeat this process several times until a total of ψ trees are obtained to comprise a random forest. The last step is the decision which entails averaging the tree predictions.

(ii) Support Vector Machine (SVM)

The idea of the Support Vector Machine (also known as the classification and regression procedure) is that it should map the input vectors into high dimensional feature space [22] and this is to simplify classification problem in the feature space. In a high dimensional feature space mapped from the input data using the device called kernel mapping, the problem can become linearly separable [23]. SVM has commendable capacity in establishing unknown relationships which may exist between a set of various input variables

and the system output. The key aspect of SVM which makes it so advantageous is the use of a kernel trick for establishing knowledge about a given problem in a manner that can allow both model residuals (or prediction errors) and the model complexity to be minimized in a simultaneous way.

Let us assume that we have a training dataset $(\mathbf{x}, y)$ such that $\mathbf{x} \in \mathfrak{R}^n$ and $y \in \mathfrak{R}$ where $\mathfrak{R}$ denotes the input space with $\mathbf{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots\}$ as the input vector while y is the system output [24]. If $\mathbf{v}$ represents the matrix of regression weights while b is the bias term (which denotes the threshold in the support vector machines), to obtain a function f with small risk using an independent uniformly distributed dataset $(\mathbf{x}_1, y_1)$,...., $(\mathbf{x}_n, y_n)$, we can, by some nonlinear mapping Φ, *map* $\mathbf{x}$ into the feature space [25] with the nonlinear estimate function given by $y = f(\mathbf{x}, \mathbf{v}) = \mathbf{v}^{\mathrm{T}}\Phi(\mathbf{x}) + b$. Following Vapnik [22] and Cortes and Vapnik [26], regularized risk functional for obtaining small risk can be given in terms of the penalty parameter (D), number of support vectors (n^*), small positive number (G), and support vectors obtained from training data $(\mathbf{x}_i)$ such that

$$\frac{1}{2}\|\mathbf{v}\|^2 + \frac{D}{n^*}\sum_{i=1}^{n^*}|y_i - f(\mathbf{x}_i, \mathbf{v})|_G. \tag{4}$$

If $(\beta_i^{\#}, \beta_n)$ denotes coefficients which can be determined by training, while $K(\mathbf{x}_i, \mathbf{x}_j)$ is the support vector kernel, we can form a kernel to satisfy the Mercer's condition or a dot-product kernel by $K(\mathbf{x}, \mathbf{x}') = K(\langle \mathbf{x}.\mathbf{x}' \rangle)$ [27]. Equation (4) leads us to dual optimization problem in which we have to

$$\begin{aligned}&\text{Maximize } W\left(\beta^{\#}\right) = -G \times \sum_{i=1}^{n^*}\left(\beta_i^{\#} + \beta_i\right)\\ &+ \sum_{i=1}^{n^*}\left(\beta_i^{\#} - \beta_i\right)y_i - \frac{1}{2}\sum_{i=1}^{n^*}\sum_{j=1}^{n^*}\left(\beta_i^{\#} - \beta_i\right)\left(\beta_j^{\#} - \beta_j\right)K\left(\mathbf{x}_i.\mathbf{x}_j\right).\\ &\text{subject to}\\ &\sum_{i=1}^{n^*}\left(\beta_i^{\#} - \beta_i\right) = 0 \quad \beta_i^{\#} \in [0, D]\end{aligned} \tag{5}$$

Finally, the decision function can be given by $f(\mathbf{x}) = \sum_{i=1}^{n^*}(\beta_i^{\#} - \beta_i)K(\mathbf{x}, \mathbf{x}') + b$, such that the $K(\mathbf{x}, \mathbf{x}')$, as normally taken to be the Gaussian kernel, is computed using

$$K\left(\mathbf{x}, \mathbf{x}'\right) = \exp\left(-\frac{\left\|\mathbf{x} - \mathbf{x}'\right\|^2}{2\sigma^2}\right). \tag{6}$$

(iii) Generalized Linear Models (GLMs)

GLMs consist of a modelling framework which lead to a family of models. Each of the family members makes use of a particular link function and specific distribution. A link function is used to mathematically establish the linearity. The outcome variable can be fitted using a particular distribution. What is important here is the selection of the shape which can match the given randomness and in this way, we can eliminate the tough assumption of a constant variance. There are several distributions which tend to be used. For instance, this study employed GLMs based on beta distribution (GLMB), normal distribution (GLMN), poisson distribution (GLMP), gamma distribution (GLMG), and inverse Gaussian distribution (GLMI).

(iv) Discriminant Analysis Model (DAM)

Discriminant analysis is a natural technique to make forecast especially when the predictand comprises a finite set of discrete groups given that vectors of predictors are known. The discriminant analysis as a classification technique relies on the assumption that different classes can be generated based on different Gaussian distributions. Discriminant analysis can be applied to characterize differences between groups and use the characteristics for classification of a new member to be added to the groups, and this is based on the observations obtained from the member. In discriminant analysis, a member is characterized using a vector of variables which comprise a multivariate density function. The multidimensional characteristics of the density function of the population's variable are mapped onto a one dimensional measure using a discriminant function.

In a DAM, the predictor $\mathbf{x}$ is assumed to have a Gaussian mixture distribution. For DAM, two options exist including linear discriminant analysis and quadratic discriminant analysis. In the former case, we assume that only the means vary while each class has the same covariance matrix. In the latter case, both covariance and the mean of every class vary. In training a classifier, parameters of a Gaussian distribution for every class are estimated using a suitable fitting function. For predicting the classes of new data, the trained classifier determines the class which has the smallest misclassification cost.

(v) Gaussian Process Regression (GPR)

GPR model is a nonparametric kernel-based probabilistic model. Let n be the sample size. If we have a training dataset $\{(x_i, y_i)\,; i = 1, 2, \cdots, n\}$ such that t, $x_i \in \mathfrak{R}^n$, and $y_i \in \mathfrak{R}$ all drawn from an unknown distribution, we can use GPR to predict the system output y given the new input vector $\mathbf{x}$. We can make use of a linear regression of the form $y = x^T\alpha + \varepsilon$ such that $\varepsilon \sim N(0, \sigma^2)$. The data can be used to estimate the coefficient α and the error covariance σ^2. To explain the response of the system, GRP introduces (i) explicit basis function h to project the predictors or inputs x_i into p-dimensional feature space and (ii) the set of latent variables $f(x_i)\,; 1 \le i \le n$ from a Gaussian process for capturing the smoothness of the system response.

A GPR model may be represented as $h(x)^T\alpha + f(x)$, such that $f(x) \sim$ Gaussian process with mean $= 0$ and covariance $= K(x, x')$. The system response y can be modelled using

$$P(y_i|f(x_i), x_i) \sim N\left(y_i\middle|h(x_i)^T\alpha + f(x_i), \sigma^2\right). \tag{7}$$

2.1.3. Water Quality Data. To evaluate the performance of the various models, water quality datasets based on samples taken from the Lirima Gravity Flow Scheme (LGFS) in Uganda, East Africa, were used. The LGFS is owned and operated by the National Water and Sewerage Corporation (NWSC) as a utility parastatal under sole ownership of the Government of Uganda. NWSC is known for its commitment for the provision of clean and safe water in the various towns, cities, and urban centers across Uganda. Sampling and testing water from the LGFS (which is about 90 km in length) for this research followed a formal permission granted by the Research and Development Department from the Head Office of the NWSC in Kampala.

Several points were selected within the LGFS for water sampling. At each location or point, water was sampled several times (both in the morning and afternoon) within each day. From each water sample, several water quality parameters were tested and recorded including water temperature (°C), pH, electrical conductivity (μS), and RCC (mg/l). These factors were reported to influence decay of chlorine (see, e.g., [2, 28–31]), and they were deemed to be possible predictors of RCC.

It is important to recall that chlorine decay depends on time and this follows the kinetic of chlorine reaction with water as governed by the first order equation $C_{\text{res}}(t) = C_{\text{ini}} \times \exp(-K_b \times t)$ where $C_{\text{res}}(t)$ is the RCC (mg/l) at time t, C_{ini} denotes the chlorine concentration (mg/l) at time $t = 0$, and K_b refers to the chlorine bulk reaction constant (measured per hour). Thus, the time t at which the water was sampled was also recorded. Eventually, water temperature (°C), pH, electrical conductivity (μS), and the sampling time were considered as predictors (X_1, X_2, X_3, and X_4, respectively) of the RCC. To do so, X_4 was converted from the format of an hour (e.g., 13 : 00 hours) to become a number (i.e., 0.54) and this was done using an in-built function (or option for number formatting) in the Ms Excel.

2.1.4. Training and Testing of the Various Models. Performance of each of the various deep learning and machine learning algorithms was assessed using several "goodness-of-fit" metrics including the coefficient of determination (R^2 or R-squared), revised R-squared (RRS, [32]), Nash Sutcliffe efficiency [33], hydrological model skill score or Onyutha efficiency (OE, [32]), root mean square error (RMSE), percentage bias (PBIAS), symmetric mean absolute percentage error (SMAPE), index of agreement (IOA, [34]), Taylor skill score (TSS, [35]), and distance correlation (r_d, [36]). Based on the observed (x) and modelled (y) RCC, consider some of the relevant terms of these "goodness-of-fit" metrics to be sample size (n), mean of $x(\bar{x})$, mean of $y(\bar{y})$, variance of $x(\mathrm{s}_{\mathrm{x}}^2)$, variance of $y(\mathrm{s}_{\mathrm{y}}^2)$, standard deviation of x (s_x), standard deviation of $y(\mathrm{s}_y)$, normalized $s_y(\hat{s}_y)$, distance covariance of $x(\mathrm{d}_{xx})$, distance covariance of $y(\mathrm{d}_{yy})$, distance covariance of x and $y(\mathrm{d}_{xy})$, maximum correlation attainable (r_m) (in this study r_m was set to 0.99), and variance of y constrained to $\bar{x}$ as the comparison baseline (s_{yc}^2) such that $\mathrm{s}_{yc}^2 = (n-1)^{-1} \times \sum_{i=1}^{n} (y - \bar{x})^2$. The various measures to assess model performance were computed using the following

$$R^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{8}$$

$$\mathrm{RRS} = |r| \times \frac{\min\left(\mathrm{s}_{\mathrm{x}}, \mathrm{s}_{\mathrm{y}}\right)}{\max\left(\mathrm{s}_{\mathrm{x}}, \mathrm{s}_{\mathrm{y}}\right)} \times \frac{\min\left(\mathrm{s}_{\mathrm{x}}^2, \mathrm{s}_{yc}^2\right)}{\max\left(\mathrm{s}_{\mathrm{x}}^2, \mathrm{s}_{yc}^2\right)}, \tag{9}$$

$$\mathrm{NSE} = 1 - \frac{\sum_{i=1}^{n}(x_i - y_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \tag{10}$$

$$r_d = \frac{\mathrm{d}_{\mathrm{xy}}}{\sqrt{\mathrm{d}_{\mathrm{xx}} \times \mathrm{d}_{\mathrm{yy}}}}, \tag{11}$$

$$\mathrm{OE} = r_d \times \frac{\min\left(\mathrm{d}_{\mathrm{XX}}, \mathrm{d}_{\mathrm{YY}}\right)}{\max\left(\mathrm{d}_{\mathrm{XX}}, \mathrm{d}_{\mathrm{YY}}\right)} \times \frac{\min\left(\mathrm{s}_{\mathrm{x}}^2, \mathrm{s}_{yc}^2\right)}{\max\left(\mathrm{s}_{\mathrm{x}}^2, \mathrm{s}_{yc}^2\right)}, \tag{12}$$

$$\mathrm{IOA} = 1 - \frac{\sum_{i=1}^{n}(x_i - y_i)^2}{\sum_{i=1}^{n}\left(|(x_i - \bar{x})| + |(y_i - \bar{y})|\right)^2}, \tag{13}$$

$$\mathrm{TSS} = \frac{4\left(1 + \left(n \times s_x s_y\right)^{-1} \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})\right)}{\left(\hat{s}_y + \left(\hat{s}_y\right)^{-1}\right)^2 (1 + r_m)}, \tag{14}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2}, \tag{15}$$

$$\mathrm{SMAPE} = \frac{100}{n} \times \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|}, \tag{16}$$

$$\mathrm{PBIAS} = \frac{100 \times \sum_{i=1}^{n}(x_i - y_i)}{\sum_{i=1}^{n} x_i}. \tag{17}$$

In Equations (9) and (12), min () denotes whichever is smaller of the two values. Each of the modelling datasets was divided into training (70%) and testing (30%) subseries. Model performance evaluation was conducted using (i) training data, (ii) testing data, and (iii) entire data (when both training and testing series were combined). Each of the "goodness-of-fit" metrics (Equations (8)–(17)) was applied to evaluate the outputs of the various deep learning and machine learning models. To download the MATLAB codes for computing RRS (Equation 9) and OE (Equation 12), the reader is referred to https://doi.org/10.5281/zenodo.6570904 and the access is unrestricted.

The next step comprised ranking the models to indicate which one was the best or the worst in terms of performance to reproduce the available observations. Based on the values of a particular "goodness-of-fit" metric, models were ranked. Each of the metrics in Equations (8), (9), and (11)–(14) varies from zero to one. A value of zero shows the worst model performance. For an ideal model (or one with the best performance), each of these metrics (Equations (8), (9), and (11)–(14)) gives a value of one. The best and worst model

Model	R^2	RRS	OE	RMSE	PBIAS	NSE	sMAPE	TSS	IOA	r_d
a) Training										
NARX	0.55	0.31	0.16	0.07	−6.15	0.54	22.34	0.80	0.83	0.55
FFB	0.62	0.37	0.22	0.06	−4.63	0.62	20.71	0.84	0.87	0.61
FFDT	0.50	0.28	0.14	0.07	−8.77	0.48	22.67	0.77	0.81	0.52
LR	0.49	0.26	0.13	0.07	−3.03	0.49	22.16	0.76	0.81	0.51
ANFIS	0.57	0.33	0.17	0.06	−0.12	0.57	21.81	0.81	0.85	0.58
LTSM	0.51	0.20	0.08	0.07	−0.12	0.51	23.53	0.72	0.80	0.48
RF	0.65	0.30	0.19	0.06	−6.76	0.63	20.01	0.81	0.87	0.64
SVM	0.25	0.08	0.02	0.09	1.18	0.25	27.05	0.52	0.63	0.27
DAM	0.17	0.30	0.15	0.10	6.77	−0.07	30.34	0.70	0.66	0.19
GLMB	0.27	0.08	0.02	0.08	−8.33	0.25	27.68	0.52	0.64	0.28
GLMN	0.28	0.07	0.02	0.08	−8.71	0.26	27.63	0.49	0.63	0.28
GLMP	0.26	0.09	0.02	0.09	−8.37	0.25	27.73	0.53	0.64	0.27
GLMG	0.22	0.12	0.03	0.09	−8.78	0.18	28.10	0.59	0.62	0.25
GLMI	0.15	0.21	0.04	0.10	−9.87	−0.04	28.95	0.66	0.58	0.21
GPR	0.61	0.31	0.17	0.06	−5.86	0.60	21.74	0.81	0.86	0.59
b) Testing										
NARX	0.30	0.30	0.14	0.07	10.50	0.51	20.90	0.78	0.82	0.50
FFB	0.42	0.42	0.25	0.06	8.41	0.66	17.97	0.86	0.89	0.65
FFDT	0.27	0.27	0.11	0.07	8.19	0.51	21.39	0.76	0.82	0.49
LR	0.32	0.32	0.15	0.07	17.93	0.45	22.98	0.78	0.80	0.53
ANFIS	0.48	0.48	0.24	0.09	19.02	0.07	28.64	0.77	0.72	0.31
LTSM	0.29	0.29	0.12	0.09	−26.34	0.16	25.07	0.70	0.71	0.38
RF	0.36	0.36	0.22	0.06	11.10	0.67	16.16	0.84	0.89	0.69
SVM	0.18	0.18	0.08	0.09	23.58	0.20	25.35	0.61	0.66	0.38
DAM	0.23	0.23	0.14	0.12	29.58	−0.54	33.36	0.62	0.57	0.15
GLMB	0.11	0.11	0.04	0.08	16.11	0.25	23.46	0.53	0.65	0.37
GLMN	0.14	0.14	0.06	0.08	16.85	0.26	24.04	0.59	0.67	0.37
GLMP	0.10	0.10	0.04	0.08	16.19	0.24	23.46	0.52	0.64	0.37
GLMG	0.07	0.07	0.02	0.09	17.00	0.20	24.12	0.43	0.60	0.36
GLMI	0.04	0.04	0.01	0.09	19.11	0.14	25.01	0.29	0.54	0.36
GPR	0.36	0.36	0.22	0.06	11.33	0.64	17.22	0.83	0.88	0.66
c) Entire data (both validation and testing sub-series combined)										
NARX	0.54	0.29	0.14	0.07	−0.48	0.54	21.92	0.79	0.83	0.51
FFB	0.64	0.37	0.21	0.06	−0.20	0.64	19.87	0.84	0.87	0.61
FFDT	0.50	0.26	0.12	0.07	−2.98	0.50	22.28	0.76	0.81	0.49
LR	0.49	0.26	0.12	0.07	4.09	0.49	22.41	0.76	0.81	0.48
ANFIS	0.46	0.44	0.21	0.08	8.66	0.39	23.89	0.80	0.80	0.45
LTSM	0.44	0.28	0.12	0.08	−9.05	0.41	24.00	0.76	0.79	0.41
RF	0.66	0.30	0.18	0.06	−0.66	0.65	18.85	0.81	0.87	0.64
SVM	0.26	0.09	0.03	0.09	8.81	0.24	26.52	0.54	0.64	0.27
DAM	0.13	0.27	0.12	0.11	14.54	−0.19	31.25	0.67	0.62	0.15
GLMB	0.26	0.08	0.02	0.08	0.01	0.26	26.39	0.52	0.64	0.27
GLMN	0.27	0.07	0.02	0.08	−0.01	0.27	26.54	0.51	0.64	0.27
GLMP	0.26	0.08	0.02	0.09	−0.01	0.26	26.44	0.52	0.63	0.27
GLMG	0.21	0.09	0.02	0.09	−0.02	0.20	26.90	0.54	0.61	0.25
GLMI	0.14	0.13	0.02	0.10	−0.01	0.03	27.76	0.60	0.56	0.22
GPR	0.62	0.31	0.17	0.06	0.00	0.62	20.36	0.81	0.86	0.59

Figure 1: Values of "goodness-of-fit" metrics for (a) training, (b) validation, and (c) entire data.

performance is shown by NSE (Equation (10)) values of one and negative infinity, respectively. However, for RMSE, SMAPE, and PBIAS (Equations (15)–(17)), the best model performance can be obtained when the value of the "goodness-of-fit" metric is zero. When there are many models, the one with the largest value of RMSE indicates the worst model fit. The same is true when we consider each of the metrics SMAPE and PBIAS (Equations (16) and (17)). Therefore, for each of the metrics in Equations (8)–(14), ranks of 1, 2, ….., w were given to the model with the highest, second highest, ..., lowest value, respectively, of the "goodness-of-fit" metric under consideration. However, for statistical metrics in Equations (15)–(17), ranks of 1, 2, ….., w were given to the model with lowest, second lowest,

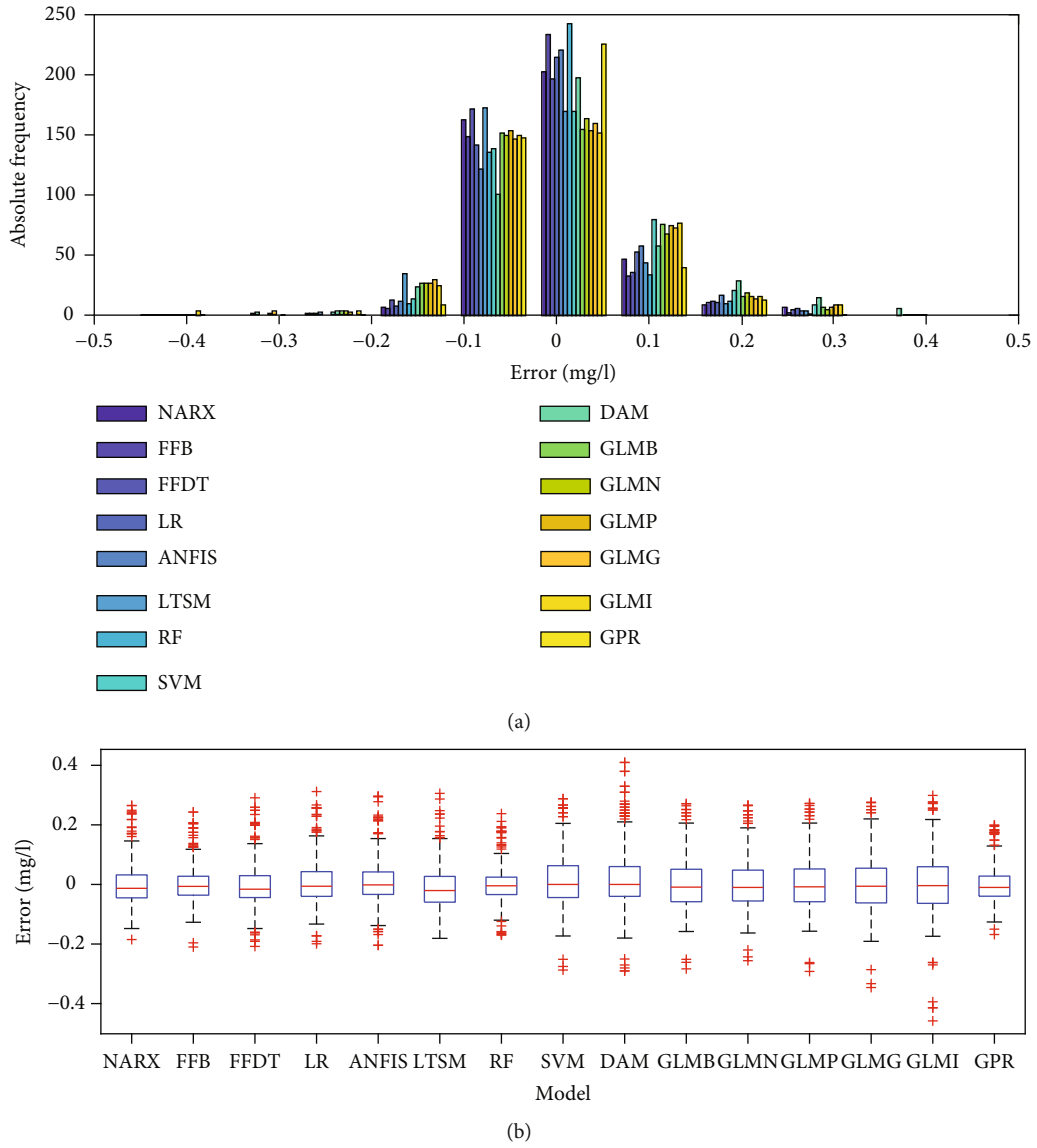


Figure 2: Error analysis based on (a) histogram and (b) boxplots.

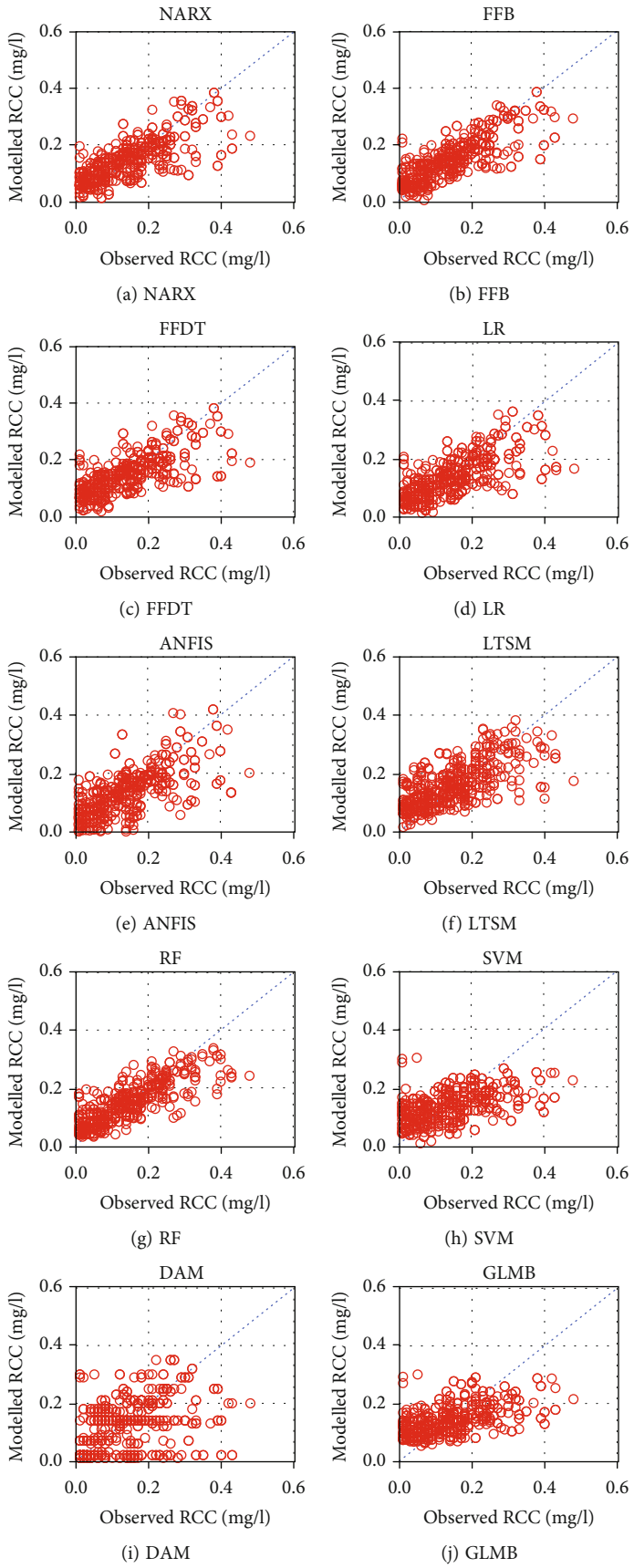
….., highest value, respectively, of a particular "goodness-of-fit" metric.

Let $\eta$to denote the number of "goodness-of-fit" metrics (and this was 10 in this study). Ranks of each of the fifteen models based on assessments from the various "goodness-of-fit" metrics were summed up. This resulted into a total of w (or 15) values which were expected to vary between $(\eta - 1)$ to $((\eta \times w) + 1)$. Finally, the sums of the ranks were sorted in ascending order. The model with the smallest sum, second smallest sum, ..., and the largest sum was given performance index 1, 2, ….., w, (as the best, second best, ….., worst model), respectively.

2.1.5. Developing Ensemble Predictions. There are various ways of obtaining model ensembles such as simple model average, weighted average method, multiple super model ensemble, and constrained multiple linear regression. For brevity, this study examined the suitability of three model ensembles including the arithmetic model average of all the modelled results (Ens1), arithmetic mean of results from the best three models (Ens2), and weighted mean of outputs from all the models (Ens3). Consider $y_i^{(j)}$ as the j^{th} output of the i^{th} model, Ens1 was computed using

$$\text{Ens1}_j = \frac{1}{w}\sum_{i=1}^{w} y_i^{(j)} \quad \text{for} \quad 1 \le j \le n. \tag{18}$$

Ens2 was also computed using Equation (18) with w set to 3 (i.e., the first, second, and third best performing models). Let $r_{d(i)}$ be the distance correlation between the observed data and outputs of the i^{th} model while p is the absolute value of an arbitrary power (and p was set to 15 in this study). It is worth noting that p depends on the magnitudes of the values in the series. Ens3 was computed using



(a) NARX (b) FFB

(c) FFDT (d) LR

(e) ANFIS (f) LTSM

(g) RF (h) SVM

(i) DAM (j) GLMB

Figure 3: Continued.

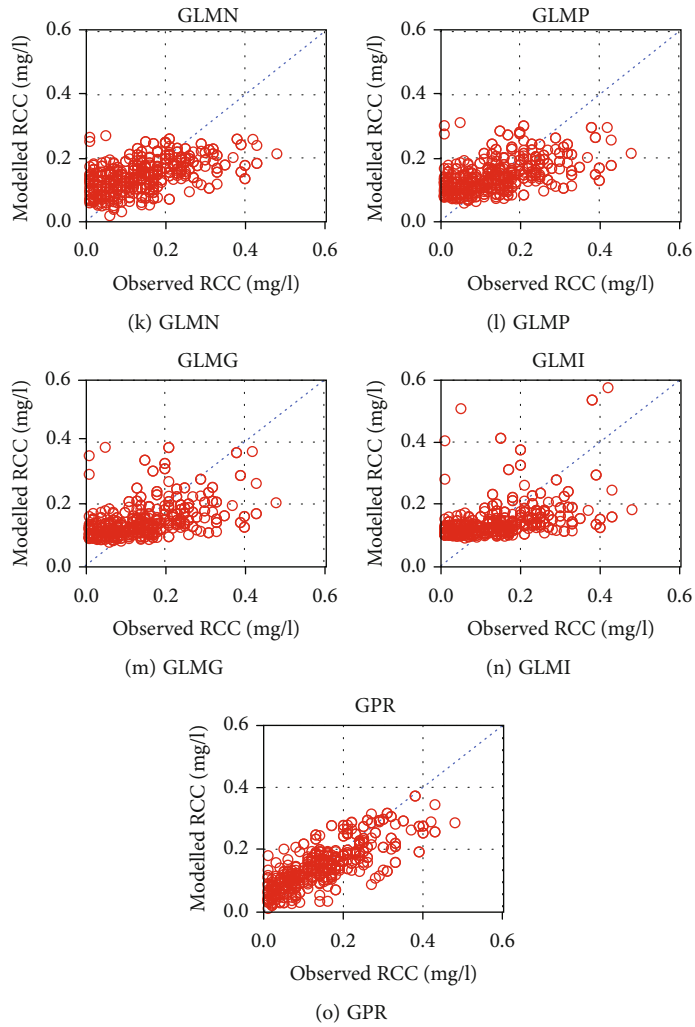


Figure 3: Observed versus modelled RCC based on (a)–(f) deep learning and (g)–(o) machine learning methods.

$$\text{Ens3}_j = \frac{\sum_{i=1}^{w}\left(y_i^{(j)} \times \left(r_{d(i)}\right)^p\right)}{\sum_{i=1}^{w}\left(r_{d(i)}\right)^p} \quad \text{for} \quad 1 \le j \le n. \tag{19}$$

3. Results

Figure 1 shows performance of the various statistical models in reproducing observed RCC. Notably, the performance depends on the selected "goodness-of-fit" metric (Figure 1). This is because of the differences among the "goodness-of-fit" metrics. For instance, some metrics like NSE, IOA, and RMSE are based on squared model residuals while RSS and OE consider combined measures of variability, bias, and correlation. Furthermore, the various objective functions differ in terms of their ranges over which their values occur. For instance, NSE ranges from negative infinity to one while R^2, RRS, OE, TSS, IOA, and r_d occur over the range 0–1. Values of RMSE range from zero to positive infinity. However, the RMSE values were all small in magnitude and one may associate these values to a possible excellent performance of the models. Nevertheless, it should be noted that the small values of RMSE were due to the small values of the RCC. For instance, the observed RCCs ranged from 0.01 to 0.48 mg/l with an average of 0.14 mg/l. For all the models, some metrics especially TSS and IOA exhibited values which were close to one (indicating best model performance). The sensitivity of IOA in yielding values close to one even for models which is characterized by poor fit was shown by Krause et al. [37]. The tendency of TSS to yield values close to 1 for models which are not perfect was also recently obtained by Onyutha [32].

Considering the average of each "goodness-of-fit" metric, the deep leaning methods (NARX, FFB, FFDT, LR, ANFIS, and LSTM) were generally better than the machine learning techniques (RF, SVM, DAM, GLMB, GLMN, GLMP, GLMG, GLMI, and GPR). The best two performing machine learning methods included RF and GPR. On the other hand, FFB and NARX were the two best deep learning methods. In most cases, model results of training were slightly better than those for testing. Furthermore, models evaluated using the entire data (both calibration and testing subseries) were found to perform better than when testing subseries were used.

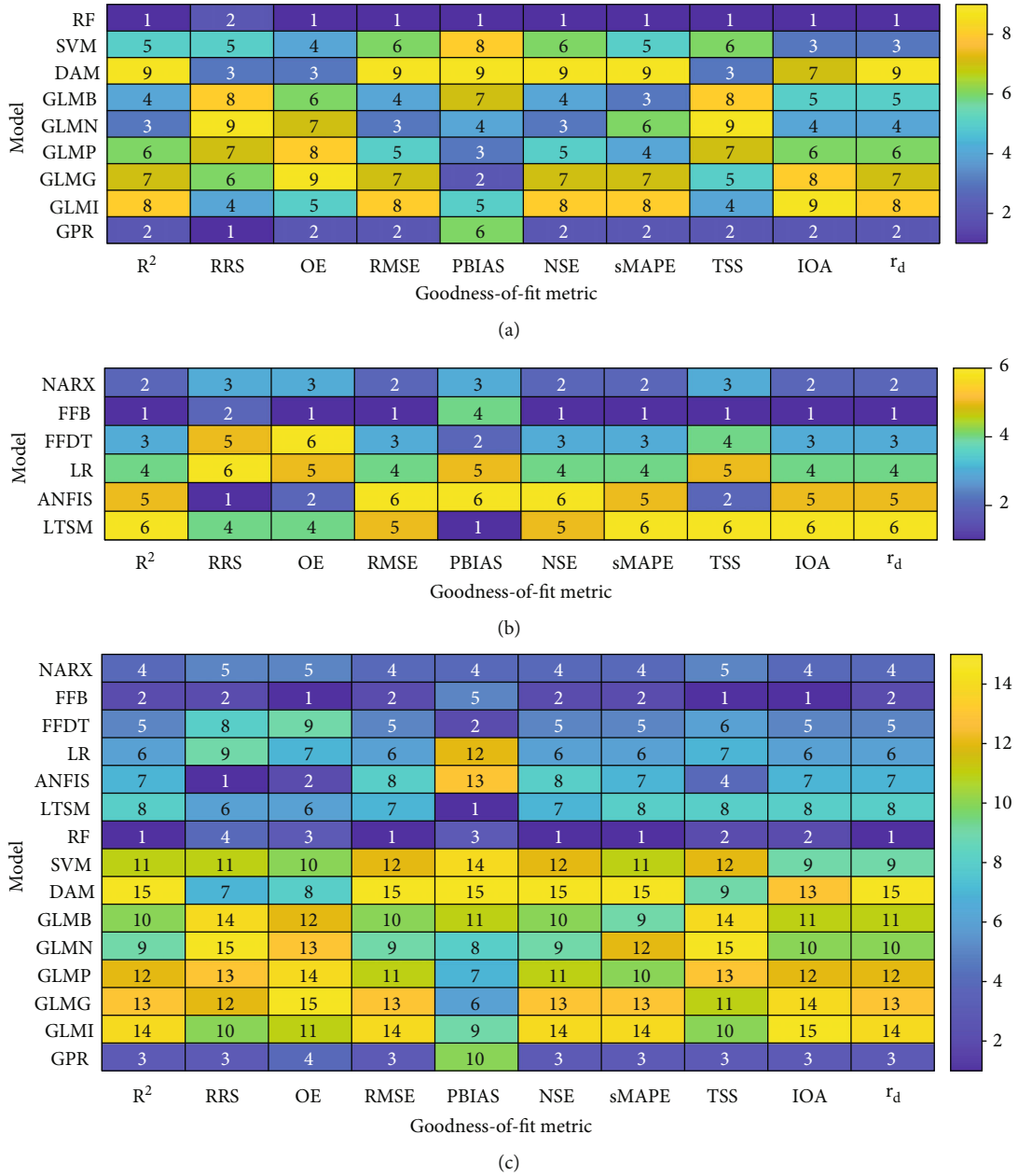


Figure 4: Performance of models considering (a) machine learning, (b) deep learning, and (c) combination of both machine learning and deep learning methods.

Figure 2 shows results of error analysis. For each model, the errors were mainly concentrated around zero. Values ranging from -0.1 to 0 were greater in number than those over the range 0-0.1 (Figures 2(a) and 2(b)). The median lines of the boxplots for the various models are notably around or close to zero. These impressions can suggest that the models exhibited very high performance. However, by realizing that the actual observed (target) data against which each model was being calibrated ranged from 0.01 to 0.48 mg/l, it means that the impressive performance was due to the order of magnitude of the values being considered (Figure 2(b)). The difference between two small values is even smaller. Nevertheless, it is noticeable that the whiskers of the boxplots of the models go beyond the absolute value of 0.2 mg/l. This suggest that large observed values were mainly either overestimated or underestimated. Based on the results for graphical error analysis, the best performing models were FFB, RF, and GPR (Figure 2(b)). This, however, could not be conclusive without further analysis of the model performances as done next.

Figure 3 shows comparison of observations with model outputs. Each of the charts (Figures 3(a)–3(o)) contains the 1 : 1 or $y = x$ diagonal line through the scatter points. For an effective model, all the scatter points would fall along the bisector (or diagonal line). For deep learning methods (Figures 3(a)–3(f)) and three of the machine learning techniques including RF, SVM, and GPR (Figures 3(g)–3(h), (o)), the scatter points especially for RCC values lower than

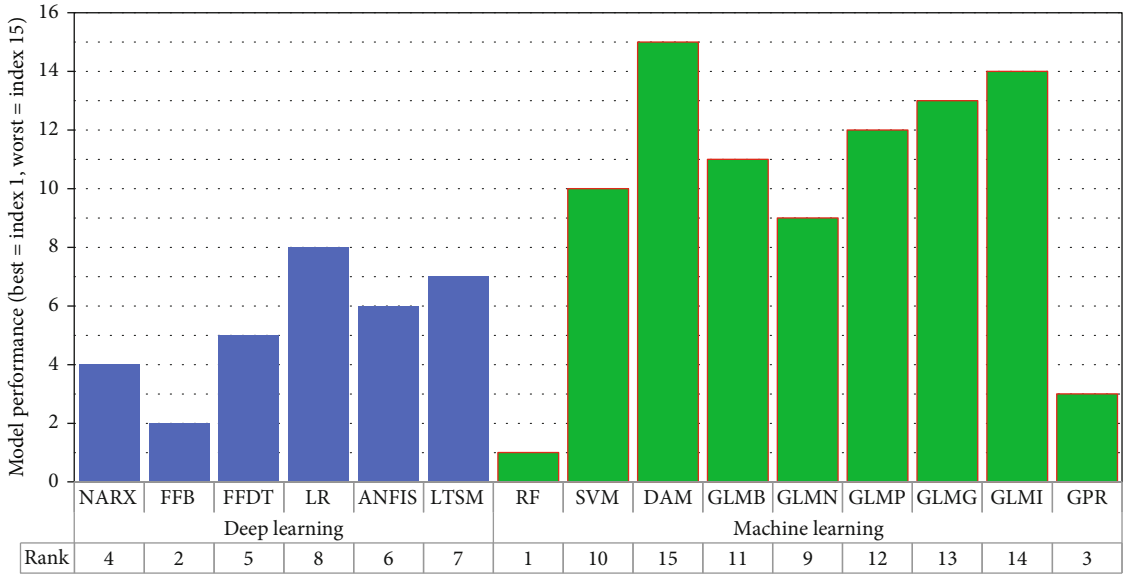


Figure 5: Overall performance of models.

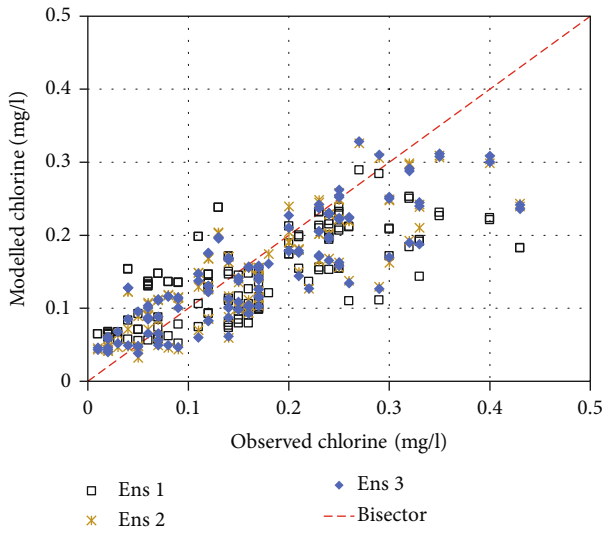


Figure 6: Observed versus modelled RCC for ensemble model.

0.3 mg/l were distributed above and below the bisector. The scatter points from DAM were more widely distributed than those of other models. For the remaining machine learning techniques (Figures 3(j)–3(n)), the scatter points tended to be distributed horizontally between the lines $y = 0$ and $y = 0.2$ mg/l. The scatter points especially, for RCC values greater than 0.3 mg/l, were characterized by large spread around the bisector. One cause of this would be heteroscedascticity or the existence of increasing differences among observations. However, a close look at the scatter points shows that the large values were instead systematically underestimated. Thus, the large spread of the large values were indicative of the difficulty of the models to reproduce observations especially as the RCC increased in magnitude.

Table 1: Performance of the model ensembles.

Metric	Ens1	Ens2	Ens3
R^2	0.507	0.736	0.731
Rr	0.170	0.360	0.356
E	0.081	0.223	0.218
RMSE	0.073	0.054	0.055
PBIAS	13.237	10.280	10.416
NSE	0.443	0.691	0.685
SMAPE	20.834	16.286	16.402
TSS1	0.658	0.844	0.841
IOA1	0.763	0.893	0.891
r_d	0.509	0.708	0.703

Figure 4 shows the heat map for the model performance evaluation. On average, the deep learning methods generally performed better that the machine learning techniques. For instance, the ranks obtained for the GLMs based on beta (GLMB), normal (GLMN), poisson (GLMP), gamma (GLMG), and inverse Gaussian (GLMI) distributions were large. This is consistent with the poor performance of these models as already demonstrated in Figures 1–3.

Evidently, there was no any model which scored the same rank considering all the metrics used to measure model quality. This indicates that the judgement of a model depends on the selected "goodness-of-fit" metric.

Figure 5 shows the overall summary of the model performance. The two best performing machine learning methods included RF and GPR (Figure 4(a)). On the other hand, NARX and FFB yielded the best performance compared with the other deep learning models (Figure 4(b)). When all the models were considered, the ranking process showed that the first, second, third, and fourth best models were RF, FFB, GPR, and NARX, respectively (Figure 4(c)).

Generally, the best performing GLM was that based on the normal distribution. Nevertheless, the GLMs did not generally perform well compared with other models. In summary, DAM, GLMI, GLMP, and GLMP were the four worst performing models, and they were all based on the machine learning techniques.

Figure 6 shows comparison of ensembles with observed RCC. Observed values from 0.3 mg/l and above were mainly underestimated by each of the model ensembles. This showed lack of capacity of the models to capture large RCCs. Results of the statistical measures of the mismatches between observed and ensemble RCC can be seen in Table 1. As already shown in Figure 4, RF was the best model. However, each set of the model ensembles performed better than RF (Table 1). This demonstrated the need to prefer model ensembles to results of individual models. Ens2 (or arithmetic mean of results from the top or best 20% of the 15 models) exhibited the best performance. This was followed by Ens2 (or weighted mean from all the 15 models).

4. Discussion

Application of statistical models is on the increase in a number of areas such as prediction of precipitation ([12, 38–42], river flow ([43–47]), and temperature ([48, 49], and [50]). Some recent studies on modelling water quality using statistical methods include Wadkar and Kote [51], Li et al. [52], García-Ávila et al. [29], and De Santi et al. [53]). For predicting RCC in drinking water, most of the studies (about 90%) available in literature as shown by Onyutha and Kwio-Tamale [2] applied artificial neural networks. Examples of such studies include Wadkar and Kote [51], and García-Ávila et al. [29]. As demonstrated in this study, both machine learning and deep learning methods (which are actually subsets of artificial intelligence) can be applied for making predictions of RCC. Machine learning techniques can fall under four main categories including (i) semisupervised learning, (ii) unsupervised learning, (iii) supervised learning, and (iv) reinforcement learning. The key issues with the machine learning is that it requires selection of features which must carefully be done before the process of training can be performed. This follows from the selectivity invariance issues to which machine learning techniques are susceptible. On the other hand, deep learning eliminates the requirement of feature selection by making use of various abstraction layers to automatically learn the details of the data through application of nonlinear techniques to solve complex problems.

A reliable insight on the uncertainty on modelling results due to the choice of a particular model can be obtained when many models are applied. It is known that each of the various models could be characterized by varying degree of structural complexity. Thus, combining outputs from various models to obtain one set (or an ensemble) of modelled results is important to take into account the uncertainties due to the differences among models. The concept of multimodel ensemble has not been applied in predictions of RCC in drinking water. However, as demonstrated in this study, a multimodel ensemble can perform better than outputs from individual models thereby boosting the credibility and confidence in the predictions of RCC in drinking water.

While modelling chlorine residuals, most modelers tend to make use of the R^2 or R-squared [2]. The use of R^2 is premised on the assumption that the relationship between the predictor and predictand is linear. Thus, the value of R^2 becomes wrong when we have nonlinear relationships. Other commonly used "goodness-of-fit" metrics include RMSE and mean squared error (MSE). By squaring the model residuals, large weights are assigned to big values and this makes RMSE and MSE sensitive to outliers. As shown in this study, the choice of a particular "goodness-of-fit" metric influences the selection of the best performing model. Thus, it is important to conduct comparative analysis of models while taking into consideration the influence of the choice of a particular objective function or "goodness-of-fit" metric in judging the quality of each model. Several objective functions can be considered including, among others, RRS [32], NSE [33], OE [32], PBIAS, SMAPE, IOA [34], TSS [35], and r_d [36]. Results of a careful intercomparison of models conducted to select the best performing models can crucially influence the suitability of a multimodel ensemble.

5. Conclusions

Following the limited exploration of the applicability of the statistical models for modelling chlorine decay, this study investigated the performance of deep learning and machine learning methods in predicting chlorine residuals in drinking water. Suitability of arithmetic mean of all the models (Ens1), average of the best three performing models (Ens2), and weighted mean of outputs from all the 15 models applied in this study was investigated. Generally, on average, results of deep learning methods were better than those of the machine learning techniques. Considering only the deep learning algorithms, the first and second best methods were NARX and FFB, respectively. If we consider only the machine learning algorithms, the first and second best methods included RF and GPR, respectively. The worst deep learning method and machine learning techniques included the LR neural network and DAM, respectively. By combining all the machine learning and deep learning algorithms, the first, second, and third best methods included RF, FFB, and GPR, respectively.

The total variance explained by the individual models ranged from 13% to 66%. However, multimodel ensembles explained total variance in the range 51-74%. Ens2 explained 74% of the total variance in observed residual chlorine. Thus, the performance of Ens2 was better than that for each of the individual models.

This study corroborated the acceptability of multimodel ensemble prediction of chlorine residual concentrations in drinking water. It is important that intercomparison of models should carefully be conducted to select the best model which can be applied. Comparison of the models should be based on a number of model efficiency criteria.

This is because, the use of a particular "goodness-of-fit" metric influences the judgement of model quality.

Abbreviations

ANFIS: Adaptive-network-based fuzzy inference system
DAM: Discriminant analysis model
FFN: Feedforward neural network
GLM: Generalized linear model
GLMB: GLM based on beta distribution
GLMG: GLM based on gamma distribution
GLMI: GLM based on inverse Gaussian distribution
GLMN: GLM based on normal distribution
GLMP: GLM based on poisson distribution
GPR: Gaussian process regression
IOA: Index of agreement
LGFS: Lirima gravity flow scheme
LSTM: Long short-term memory
NARX: Nonlinear autoregressive model with exogenous input
NSE: Nash-Sutcliffe efficiency
NWSC: National water and sewerage corporation
OE: Onyutha efficiency
PBIAS: Percentage bias
RCC: Residual chlorine concentration
RF: Randomforest
RMSE: Root mean square error
RNNs: Recurrent neural networks
RRS: Revised R-squared
SMAPE: Symmetric mean absolute percentage error
SVM: Support vector machine
TSS: Taylor skill score.

Data Availability

Data used in this study can be obtained upon request from the corresponding author.

Consent

This research did not involve personal information for which consent could have been sought.

Conflicts of Interest

The author declares no conflict of interest and no competing financial interests.

Acknowledgments

The author acknowledges that the datasets used in this study were based on water sampled from the Lirima Gravity Flow Scheme (LGFS) in Uganda, East Africa upon permision granted by the Research and Development Department from the Head Office of the National Water and Sewerage Corporation (NWSC) in Kampala. The author is grateful to Dr. Irene Nansubuga and Mr. Christopher Kanyesigye from the Head Office of the NWSC for the support regarding permission to access LGFS and laboratory for testing the samples. It is acknowledged that the datasets used in this study were also partially used by Mr. Julius Caesar Kwio-Tamale (JCKT) in his MSc Thesis titled "Comparison of physical and statistical models in predicting space-time decay of residual chlorine in drinking water distribution system" following a research conducted under the supervision of Dr. Charles Onyutha and submitted by JCKT in 2022 for the award of Master of Science in Water and Sanitation Engineering Degree of Kyambogo University, Uganda.

References

[1] World Health Organization, *Water safety in distribution systems*, WHO Library Cataloguing-in-Publication Data, Geneva-Switzerland, 2014, Available: (accessed: 7th March 2022).

[2] C. Onyutha and J. C. Kwio-Tamale, "Modelling chlorine residuals in drinking water: a review," *International journal of Environmental Science and Technology*, 2022.

[3] World Health Organization, *Principles and practices of drinking water chlorination: a guide to strengthening chlorination practices in small-to medium sized water supplies*, World Health Organization, Regional Office for South-East Asia, New Delhi, 2017.

[4] M. S. Ishaq, Z. Afsheen, A. Khan, and A. Amjad Khan, "Disinfection Methods," in *Photocatalysts-applications and attributes*, S. B. Khan and K. Akhtar, Eds., pp. 3–19, Intech Open, 2018.

[5] L. Vuta and G. E. Dumitran, *Some aspects regarding chlorine decay in water distribution networks*, Cluj University Press, 2011, Available via http://aerapa.conference.ubbcluj.ro/2011/PDF/Vuta_Dumitran.pdf (accessed 12th September 2022).

[6] M. Dettori, A. Piana, P. Castiglia, E. Loria, and A. Azara, "Qualitative and quantitative aspects of drinking water supply in Sardinia, Italy. A descriptive analysis of the ordinances and public notices issued during the years 2010-2015," *Annali di Igiene*, vol. 28, no. 4, pp. 296–303, 2016.

[7] L. A. Rossman, *EPANET 2.0 user manual. Water supply and water resources division, national risk management laboratory*, Cincinnati, USEPA, 2000.

[8] A. Kumar, R. Singh, P. P. Jena, C. Chatterjee, and A. Mishra, "Identification of the best multi-model combination for simulating river discharge," *Journal of Hydrology*, vol. 525, pp. 313–325, 2015.

[9] C. Onyutha, C. J. Amollo, J. Nyende, and A. Nakagiri, "Suitability of averaged outputs from multiple rainfall-runoff models for hydrological extremes: a case of river Kafu catchment in East Africa," *Int J Energ Water Res*, vol. 5, no. 1, pp. 43–56, 2021.

[10] R. Rojas, L. Feyen, and A. Dassargues, "Conceptual model uncertainty in groundwater modeling: combining generalized likelihood uncertainty estimation and Bayesian model averaging," *Water Resources Research*, vol. 44, no. 12, pp. 1–16, 2008.

[11] T. Lin, B. G. Horne, P. Tino, and C. L. Giles, "Learning long-term dependencies in NARX recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1329–1338, 1996.

[12] S. Yang, D. Yang, J. Chen, and B. Zhao, "Real-time reservoir operation using recurrent neural networks and inflow forecast from a distributed hydrological model," *Journal of Hydrology*, vol. 579, no. 124229, 2019.

[13] I. J. Leontaritis and S. A. Billings, "Input–output parametric models for non-linear systems Part I: deterministic non-

linear systems," *International Journal of Control*, vol. 41, no. 2, pp. 303–328, 1985.

[14] M. Norgaard, O. Ravn, N. K. Poulsen, and L. K. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems*, Springer, Berlin, 2000.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470), Vol. 2*, pp. 850–855, MIT Press, 1999.

[17] W. Xie, A. Noble, and A. Zisserman, "Layer recurrent neural networks," 2016, Retrieved via https://openreview.net/pdf?id=rJJRDvcex (accessed 25th February 2022).

[18] F. Visin, K. Kastner, C. Kyunghyun, M. Matteo, C. Aaron, and Y. Yoshua, "ReNet: a recurrent neural network based alternative to convolutional networks," 2015, https://arxiv.org/abs/1505.00393, 2015, vol 1505.00393.

[19] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[20] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[21] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees, Statistics/Probability Series*, Wadsworth Publishing Company, Belmont, California, USA, 1984.

[22] V. Vapnik, *The nature of statistical learning theory. 2nd ed. statistics for engineering information science*, Springer Verlag, New York, USA, 1995.

[23] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 955–974, 1998.

[24] X. Chen, Y. Li, R. Harrison, and Y.-Q. Zhang, "Type-2 fuzzy logic-based classifier fusion for support vector machines," *Applied Soft Computing*, vol. 8, no. 3, pp. 1222–1231, 2008.

[25] L. Zhang, W. Zhou, and L. Jiao, "Wavelet support vector machine," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 34, no. 1, pp. 34–39, 2004.

[26] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[27] J. Mercer, "XVI. Functions of positive and negative type, and their connection the theory of integral equations," *Philosophical Transactions. Royal Society of London*, vol. 209, no. 441-458, pp. 415–446, 1909.

[28] R. M. Clark, "Chlorine fate and transport in drinking water distribution systems: results from experimental and modeling studies," *Frontiers in Earth Science*, vol. 5, no. 4, pp. 334–340, 2011.

[29] F. García-Ávila, A. Avilés-Añazco, J. Ordoñez-Jara et al., "Modeling of residual chlorine in a drinking water network in times of pandemic of the SARS-CoV-2 (COVID-19)," *Sustain Environ Res*, vol. 31, no. 12, 2021.

[30] A. Y. Karikari and J. A. Ampofo, "Chlorine treatment effectiveness and physico-chemical and bacteriological characteristics of treated water supplies in distribution networks of Accra-Tema metropolis, Ghana," *Applied Water Science*, vol. 3, no. 2, pp. 535–543, 2013.

[31] Q. Mao, J. Feng, W. Wang, Q. Wang, Z. Hu, and S. Yuan, "Chlorination of parabens: reaction kinetics and transformation product identification," *Environmental Science and Pollution Research*, vol. 23, no. 22, pp. 23081–23091, 2016.

[32] C. Onyutha, "A hydrological model skill score and revised R-squared," *Hydrology Research*, vol. 53, no. 1, pp. 51–64, 2022.

[33] J. E. Nash and J. V. Sutcliffe, "River flow forecasting through conceptual models part I – a discussion of principles," *Journal of Hydrology*, vol. 10, no. 3, pp. 282–290, 1970.

[34] C. J. Willmott, "On the validation of models," *Physical Geography*, vol. 2, no. 2, pp. 184–194, 1981.

[35] K. E. Taylor, "Summarizing multiple aspects of model performance in a single diagram," *Journal of Geophysical Research*, vol. 106, no. D7, pp. 7183–7192, 2001.

[36] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *Annals of Statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.

[37] P. Krause, D. P. Boyle, and F. Bäse, "Comparison of different efficiency criteria for hydrological model assessment," *Advances in Geosciences*, vol. 5, pp. 89–97, 2005.

[38] J. A. Anochi, V. A. de Almeida, and H. F. de Campos Velho, "Machine learning for climate precipitation prediction modeling over South America," *Remote Sensing*, vol. 13, no. 13, p. 2468, 2021.

[39] S. Narejo, M. M. Jawaid, S. Talpur, R. Baloch, and E. Pasero, "Multi-step rainfall forecasting using deep learning approach," *Peer J Computer sci*, vol. 7, p. e514, 2021.

[40] W. M. Ridwan, M. Sapitang, A. Aziz, K. F. Kushiar, A. N. Ahmed, and A. El-Shafie, "Rainfall forecasting model using machine learning methods: case study Terengganu, Malaysia," *Malaysia. Ain Shams Eng J*, vol. 12, no. 2, pp. 1651–1663, 2021.

[41] D. Sun, J. Wu, H. Huang, R. Wang, F. Liang, and H. Xinhua, "Prediction of short-time rainfall based on deep learning," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6664413, 8 pages, 2021.

[42] L. Xiao, M. Zhong, and D. Zha, "Runoff forecasting using machine-learning methods: case study in the middle reaches of Xijiang River," *Front Big Data*, vol. 4, no. 752406, 2022.

[43] S. Ha, D. Liu, and L. Mu, "Prediction of Yangtze River streamflow based on deep learning neural network with El Niño–Southern Oscillation," *Scientific Reports*, vol. 11, no. 1, p. 11738, 2021.

[44] D. Hussain and A. A. Khan, "Machine learning techniques for monthly river flow forecasting of Hunza River, Pakistan," *Earth Science Informatics*, vol. 13, no. 3, pp. 939–949, 2020.

[45] S. Kabir, S. Patidar, and G. Pender, "Investigating capabilities of machine learning techniques in forecasting stream flow," *Proc Inst Civ Eng-Water Manage*, vol. 173, no. 2, pp. 69–86, 2020.

[46] D. Liu, W. Jiang, L. Mu, and S. Wang, "Streamflow prediction using deep learning neural network: case study of Yangtze River," *IEEE Access*, vol. 8, pp. 90069–90086, 2020.

[47] S. Zhu, E. K. Nyarko, M. Hadzima-Nyarko, S. Heddam, and S. Wu, "Assessing the performance of a suite of machine learning models for daily river water temperature prediction," *Peer J*, vol. 7, no. e, p. e7065, 2019.

[48] M. S. Hanoon, A. N. Ahmed, N. Zaini et al., "Developing machine learning algorithms for meteorological temperature and humidity forecasting at Terengganu state in Malaysia," *Scientific Reports*, vol. 11, pp. 1–19, 2021.

[49] J. Zhao, X. Li, S. Liu, K. Wang, Q. Lyu, and E. Liu, "Prediction of hot metal temperature based on data mining," *High*

Temperature Materials and Processes, vol. 40, no. 1, pp. 87–98, 2021.

[50] S. Alawadi, D. Mera, M. Fernández-Delgado, F. Alkhabbas, C. M. Olsson, and P. Davidsson, "A comparison of machine learning algorithms for forecasting indoor temperature in smart buildings," *Energy Systems*, vol. 13, no. 3, pp. 689–705, 2022.

[51] D. Wadkar and A. Kote, "Prediction of residual chlorine in a water treatment plant using generalized regression neural network," *Int J Civ Eng Technol*, vol. 8, no. 8, pp. 1264–1270, 2017.

[52] Z. Li, S. Huang, and J. Chen, "A novel method for total chlorine detection using machine learning with electrode arrays," *RSC Advances*, vol. 9, no. 59, pp. 34196–34206, 2019.

[53] M. De Santi, U. T. Khan, M. Arnold, J.-F. Fesselet, and S. I. Ali, "Forecasting point-of-consumption chlorine residual in refugee settlements using ensembles of artificial neural networks," *npj Clean Water*, vol. 4, no. 35, 2021.