Comparative Study of Suspended Sediment Load Prediction Models Based on Artificial Intelligence Methods

Cynthia Borkai Boye¹, Paul Boye²* and Yao Yevenyo Ziggah¹

¹ Department of Geomatic Engineering, University of Mines and Technology, Ghana
² Department of Mathematical Sciences, University of Mines and Technology, Ghana

*Corresponding author: Paul Boye, Department of Mathematical Sciences, University of Mines and Technology, Ghana. Email: pboye@umat.edu.gh

Abstract: Quantification of suspended load sediment is crucial for maintaining the ecosystem and quality of water/river bodies that serve as the habitat for many living organisms. Because the influencing factors are nonlinearly related to the suspended load sediment, it is a challenge to apply linear statistical models to predict accurately. To address such a problem, this study applied artificial intelligence methods to simulate and predict suspended load sediment. The artificial intelligence methods are robust and can handle adequately issues related to nonlinearity in modeling. In the present study, four artificial intelligence methods were developed to predict suspended sediment load distribution. The methods include a backpropagation neural network, group method of data handling, least squares support vector machine, and generalised regression neural network. In developing the respective models, drainage areas, river slopes, and length of rivers served as predictor variables while suspended sediment load was the response variable. The models were evaluated using the metrics of root mean square error, percentage root mean square error, uncertainty at 95%, root mean square error observations standard deviation ratio, and Legates and McCabe index. According to the results, the generalised regression neural network model achieved higher prediction accuracy than the other competing methods. The performance of the generalised regression neural network model can be attributed to its ability to calibrate and generalise appropriately to the training and testing data set. Hence, in practice, the generalised regression neural network model is proposed for suspended sediment load prediction for the study area which can be useful to policymakers and managers of water resources.

Keywords: generalised regression neural network; suspended sediment load; water quality

1. Introduction

Globally, river quality and its management have become increasingly complex. With higher demands for clean water for domestic and industrial use, accurate prediction of suspended sediment load (SSL) could help to improve river quality, water monitoring design, river transportation research and management, water-related decision-making, and regulatory formulation.

Increasing soil erosion as a result of the indiscriminate clearing of vegetation around river catchments together with higher stormwater runoff has impacted greatly SSL due to climate change. Consequently, the majority of sediment load is carried in suspension or dissolved in solution, with the minority moved near the bed level (Alexandrov et al., 2009; Schenk and Bragg, 2014). Although sediment plays an important role in the physical and aquatic environments, it can carry microorganisms, pollutants, and nutrients along the stream; and when in excess may pose challenges to water resource management industries, especially for low-income countries. However, the processes involved in causing these changes and challenges are presently
poorly understood. As a result, the prediction of SSL has become a great challenge, especially in developing countries in Africa where a limited number of monitoring stations are available (Chapman et al., 2016).

It is therefore a challenge to understand with certainty the spatial variability of the actual quantity of sediment load produced in Africa (Walling, 1977; Vannaeumecke, 2014). In addition, the lack of continent-wide compilation of sediment load data has also resulted in little understanding of the spatial variability of sediment load in Africa (Walling and Webb, 1985; Walling and Webb, 1996; Milliman and Farnsworth, 2013). Moreover, a gathering of regional or country-wide sediment load data could serve as a substitute to supplement the existing small amount of data (Dunne, 1979).

To accurately predict SSL, various conventional methods (e.g. multiple linear regression and principal component regression) have been used in literature, but due to the nonlinear nature of SSL data coupled with model development computational complexities, model prediction accuracy has been poor (Alp and Cigizoglu, 2007; Lafdani et al., 2013). For decades, artificial intelligence (AI) methodologies have been utilised in the prediction of SSL as they have successfully shown their capability to find optimal mappings between input and output variables to give good accuracy in many prediction problems (Lee et al., 2020; Nourani and Andalib, 2015; Babanezhad et al., 2021).

For example, Melesse et al. (2011) used an AI modeling approach with an error backpropagation algorithm to develop models to predict the SSL of river systems. Results showed that BPNN is the best model to predict SSL as compared to the conventional methods. Rezaei et al. (2021) also developed AI models for SSL prediction. Comparative results revealed that LSSVM was superior to the other models in terms of prediction. Mehr et al. (2021) developed the GMDH model to predict SSL. Results showed that the GMDH model predicted the sediment load accurately. Wang et al. (2009) on the other hand developed feed-forward BP and GRNN models in comparison with the classical regression models to predict SSL. Statistical results showed that the GRNN model was superior to the other models in terms of prediction accuracy.

In literature, although these models (BPNN, GMDH, LSSVM, and GRNN) have been employed in the prediction of SSL, from the no-free-lunch theorem, a developed model cannot be used to solve all real-world problems. Thus, SSL prediction is a site-specific phenomenon (Mohamed and Shah, 2018). Moreover, accurate SSL prediction usually depends on some key factors such as data characteristics availability, geographical location of the catchment area, and data quality; but does not only depend on the method applied. Therefore, a developed model’s prediction of a particular problem can vary from country to country (Adam et al., 2019; Wolpert, 2002). Consequently, this study aims to develop AI models to predict SSL and compare the models to determine the best one suitable for prediction in the study area. The models developed were the group method of data handling (GMDH), least squares support vector machine (LSSVM), backpropagation neural network (BPNN), and generalised regression neural network (GRNN). The reason is that these methods (BPNN, GMDH, LSSVM, and GRNN) have been applied successfully and evaluated in many fields, which also include SSL prediction with promising results (Hazarika et al., 2020; Kisi, 2012; Cigizoglu and Alp, 2006). Yet, the implementation of the applied AI methods (BPNN, GMDH, LSSVM, and GRNN) in the Ghanaian setup is to be thoroughly explored.

With the world gearing towards digital technology, AI has become one of the main focal technologies to achieve this. However, in most developing countries like Ghana, the consciousness of using such technologies to foster quick decision-making and increase productivity is still at the infant stage. Undoubtedly, the forthcoming AI has practically impacted substantially all aspects of our life, society, employment, and firms (Makridakis, 2017). Therefore, this study fills a research gap by way of creating awareness and communicating the significance of using AI as a computational tool to solve problems in SSL prediction. This study which is worth a scientific investigation in the Ghanaian setting can also provide reliable guidance to researchers and policymakers. Therefore, the main contributions of this study are to:

- Develop SSL AI prediction models: BPNN, GMDH, LSSVM, and GRNN; and
- Compare and evaluate the developed models to determine the best model that suits SSL prediction for the study area. Thus, the presented study has provided a comprehensive assessment of the developed AI models for improving SSL prediction accuracy.

2. Materials

Using the study area topographic map, the coastal river drainage areas were delineated and digitised in a Geographical Information System (GIS) environment. A digital planimeter was then used to compute the coverage areas enclosed by the catchments of the rivers for validation purposes. Relying on the river flow existing data of the main rivers Butre, Ankobra, and Pra in the study area (southwestern Ghana), the discharge values were estimated for each river (Boy et al., 2019). Applying the logical method (Thompson, 2006; Kim et al., 2003; Booth et al., 2002), peak discharge was estimated from runoff rates of rainfall at maximum stormwater (Packman and Kidd, 1980). The logical method was employed due to its easiness in estimating the discharge values for small drainage basins, wide usage for computing peak discharge, and its efficiency in working with limited rainfall and drainage data (Montalto et al., 2007; Cleveland et al., 2011). Equation (1) shows the discharge of the peak flood.

\[ Q_{w} = 277 \times 10^{-3} IAC \] (1)

where \( Q_{w} \) is instantaneous water discharge (m³/s), \( I \) is the rainfall intensity (mm/h), \( A \) is the catchment area (km²), and \( C \)(constant) is the region’s runoff coefficient. By considering the duration of a whole storm in the area of study, a calculation of volumes of
runoff was made. The independent variables used to develop the model were obtained from a topographic map of the study area. Thus, the independent variables comprise catchment areas, length of longest rivers, and river slopes. From Equation (2), the coefficient of sediment rating (α) and index (m) were deduced from the analyses of data collected from 21 sediment loads monitored at various stations (Akrasi, 2011). These constants were substituted into the sediment rating regression relation (Equation (2)) (Walling, 1977; Nittrouer and Viparelli, 2014) to estimate the SSL (Q) in the catchment area. The SSL served as the dependent variable to develop the model.

\[
Q = \alpha Q_w^m
\]  

(2)

2.1 Study Area

The coast of the Western Region of Ghana was used for this study (Figure 1). The study area is located from longitudes 3°07′ to 1°40′ West and latitudes 4°40′ to 5°10′ North. It covers a land area of 23,921 km² constituting 10% of the size of the country (Boye, 2015).

The region’s coastline measures about 192 km out of the total of 540 km length of Ghana’s coastline. It is characterised by a broad continental shelf with a maximum width of about 80 km around Cape Three Points (Boye et al., 2018). The eastern part of the area is bounded by rocky coast while the western section contains uninterrupted soft beaches that extend to about 100 km. The western section comprises sandy beaches which are seldomly crossed by lagoons and other wetlands. River Pra, Tano, Bia, and Ankobra rivers are the four main rivers that drain through the region. Fluvial sediment from these rivers nourishes the shores, thus stabilising the beach from the sturdy west-east alongshore drift (Boateng, 2012). Most parts of the region are underlain by Pre-Cambrian rocks that is the Birimian and Tarkwaian series (Keates, 2021). These rocks contain precious minerals such as manganese, diamonds, and gold, which are mined in the country. In Ghana, the western region records the maximum rainfall with an average value ranging from 1250 to 2000 mm/year which nourishes the land to support the production of several crops. The western region is the largest producer of rubber, coconut, and cocoa.

![Figure 1](image_url)

**Figure 1**

Ghana Western Coast (Source: Ghana Lands Commission)

3. Model Development

3.1 Backpropagation Neural Network

The BPNN is a proposed method of ANN by (Rumelhart et al., 1986); and it has been used in many fields to solve problems. The BPNN architecture comprises input, hidden, and output layers arranged in a feedforward manner (Figure 2). In Figure 2,
Figure 2. BPNN Architecture

Input layer | Hidden layer | Output layer
---|---|---
X<sub>1</sub> | | Y<sub>i</sub>
X<sub>2</sub> | | Y<sub>k</sub>
X<sub>3</sub> | | 

Consider the BPNN architecture to have \( m \) nodes in the input, \( g \) nodes in the hidden, and \( n \) nodes in the output layers. Let \( Y_k \) and \( X_i \) denote the expected output and input data of units \( k \) and \( i \) respectively. \( w_{ij} \) and \( b_j \) are the weights and thresholds from the input layer to the hidden layer respectively, while \( w_{jk} \) and \( b_k \) are those from the hidden layer to the output layer respectively. The learning rate and the incentive function are \( \eta \) and \( f(x) \) respectively.

The BPNN algorithm is described as follows. Firstly, the data received from the external environment by the input layer enter the network through the hidden layer nodes. The data are then multiplied by their respective weights and adds them all together with a constant bias \( b \). The computation process in the hidden layer node \( j \) is as shown in Equation (3) (Gupta et al., 2011; Zhang et al., 2021).

In this study, the Gaussian transfer function (Equation (4)) (Gundogdu et al., 2016) which was used in the hidden layer screens the added signals received from the neurons with a bias constant input value of one.

\[
f(x) = \exp \left[ -\frac{(x-\mu)^2}{2\sigma^2} \right]
\]

where \( \mu \) and \( \sigma^2 \) are center and dispersion parameters respectively. The output results of the hidden layer unit \( j \) neurons are given as in Equation (5).

\[
net_j = f\left( \sum_{i=1}^{m} (w_{ij}x_i + b_j) \right) \text{ for } j = 1, 2, ..., g
\]

The output results of the output layer unit \( k \) neurons are given as in Equation (6).

\[
out_k = \sum_{j=1}^{g} (w_{jk} net_j + b_k) \text{ for } k = 1, 2, ..., n
\]

The mean square deviation (MSE) between the network and the expected outputs is known as the loss function (Equation (7)).
The forecasted results from the network are compared with the true observed target value to estimate the error by using Equation (7). Errors that do not meet the minimum error threshold, is reverted to the network for the connection weights and biases updation of the individual neurons using the generalised delta learning rule. The process is repeated till a minimum threshold of the error is obtained based on the loss function criterion. (Toker and Kaçıranlar, 2013).

For each neuron in the output layer, Equations (8) and (9) are used to update the network weights based on the delta rule (Zurada, 1994).

\[
E = \frac{1}{2} \sum_{k=1}^{n} (y_k - \text{out}_k)^2 \tag{7}
\]

\[
w_{jk} = w_{jk} + \nabla w_{jk} \tag{8}
\]

\[
w_{ij} = w_{ij} + \nabla w_{ij} \tag{9}
\]

The chain derivation rule (Equations (10 - 13) are used to update the values of each parameter as shown as follows:

\[
\nabla w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} = -\eta \left( \frac{\partial E}{\partial \text{out}_k} \right) \left( \frac{\partial \text{out}_k}{\partial w_{jk}} \right) = \eta e_k \text{net}_j \tag{10}
\]

\[
\nabla w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta \left( \frac{\partial E}{\partial \text{out}_k} \right) \left( \frac{\partial \text{out}_k}{\partial w_{ij}} \right) \left( \frac{\partial \text{out}_j}{\partial w_{ij}} \right)
\]

\[
= \eta \sum_{k=1}^{n} e_k w_{jk} \frac{\partial}{\partial x} \left( \sum_{i=1}^{m} (w_{ij} x_i + b_j) \right) x_i \tag{11}
\]

\[
w_{jk} = w_{jk} + \eta e_k \text{net}_j \tag{12}
\]

\[
w_{ij} = w_{ij} + \eta \sum_{k=1}^{n} e_k w_{jk} \frac{\partial}{\partial x} \left( \sum_{i=1}^{m} (w_{ij} x_i + b_j) \right) x_i \tag{13}
\]

where \( \frac{\partial}{\partial x} (.) \) is the derivative of the excitation function into the output layer which emanates from the input layer. Equations (14) and (15) are used to update the network thresholds in units \( k \) and \( j \):

\[
b_k = b_k + \nabla b_k \tag{14}
\]

\[
b_j = b_j + \nabla b_j \tag{15}
\]

where \( \nabla \) is the gradient vector. The chain derivative rule is used to compute the updated values of each threshold parameter as follows:

\[
\nabla b_k = -\eta \frac{\partial E}{\partial b_k} = -\eta \left( \frac{\partial E}{\partial \text{out}_k} \right) \left( \frac{\partial \text{out}_k}{\partial b_k} \right) = \eta e_k \tag{16}
\]
\[ \nabla b_j = -\eta \frac{\partial E}{\partial b_j} = -\eta \left( \frac{\partial E}{\partial \text{net}_j} \left( \frac{\partial b_k}{\partial b_j} \right) \right) \]

\[ = \eta \sum_{k=1}^{n} (e_k w_{jk}) \frac{\partial}{\partial x} \left( \sum_{i=1}^{m} (w_{ij} x_i + b_j) \right) \]

\[ b_k = b_k + \eta e_k \] (17)

\[ b_j = b_j + \eta \sum_{k=1}^{n} (e_k w_{jk}) \frac{\partial}{\partial x} \left( \sum_{i=1}^{m} (w_{ij} x_i + b_j) \right) \] (18)

Studies have shown that the scaled conjugate gradient algorithm was the preferred algorithm for training the BPNN. The reason being that the automatic algorithm does not require fine tuning of the parameters, and it is faster as well (Moller, 1993).

### 3.2 Group Method of Data Handling

The GMDH is a feed-forward network formed to search for the optimum solution of a complex nonlinear problems based on the fundamentals of heuristic self-organisation systems. GMDH is an algorithm to find a linear parameter complex polynomial function. By external criteria, the algorithm performs selection of the best solution. GMDH can be described as subset components formulation of the function in Equation (20) (Nguyen et al., 2019; Stefenon et al., 2020).

\[ \hat{h}(A) = b_0 + \sum_{u=1}^{m}(b_u f_u) \] (20)

where \( A = (a_1, a_2, \ldots, a_n) \) is the input vector, \( B = (b_0, b_1, \ldots, b_m) \) is the coefficients vector (weights), \( \hat{h} \) is each iteration prediction, \( f_u \) are the elementary functions that depends on diverse subsets as inputs, and \( m \) is the number of function components.

For the best solution to be obtained, GMDH uses many subsets of the base function (Equation (20)) components. The least squares method was used to estimate the coefficients of the models. As a regression technique, the use of the least squares was to approximate the system solution by minimising the residual sum of squares produced during the process. Gradually, the GMDH upsurges the partial components number and looks for a perfect complexity structure shown by a minimum value of an external criterion.

Prior to the development of the model, the external information was partitioned into training and testing sets according to a definite percentage. The data was trained to estimate the model coefficients and the test set was used to check the model soundness. The neuron was assessed and verified by an external criterion and those with the worst prediction were rejected. In the next layer, reorganisation, training, testing, and selection processes were performed again until the prediction error of the neuron stops decreasing. Figure 3 shows the GMDH structure where the asterisk neurons were removed because of poor prediction (Rayegani and Onwubolu, 2014; Armaghani et al., 2020).

Consider the use of a sediment load dataset. The relationships between the lags are learned by the algorithm and the path to follow is selected automatically by GMDH. In the GMDH neural network, the input and output variables mapping depict a nonlinear function as shown in Equation (21). Consider the input pair variables \( a_u \) and \( a_v \), Equation (22) represents the regression method employed to solve for the vector of coefficients.

\[ \hat{h}(A) = b_0 + \sum_{u=1}^{n}(b_u a_u) + \sum_{u=1}^{n} \sum_{v=1}^{n}(b_{uv} a_u a_v) + \sum_{u=1}^{n} \sum_{v=1}^{n} \sum_{t=1}^{n}(b_{uvt} a_u a_v a_t) + \ldots \] (21)

\[ G(a_u, a_v) = b_0 + b_1 a_u + b_2 a_v + b_3 a_u^2 + b_4 a_v^2 + b_5 a_u a_v \] (22)
Equation (23) represents the regularity external criterion, where $\hat{h}_u$ is the actual target values. The least $W_t$ of the layer is noted, and when $W_t$ is not reducing anymore as compared with the preceding layer’s value, the network forecast error is said to be nondecreasing and the result of the preceding layer is produced.

$$W_t = \frac{\sum_{u=1}^{n}(\hat{h} - h_u)^2}{\sum_{u=1}^{n}h_u^2}$$  \hspace{1cm} (23)$$

where $p$ is the number of testing data set. The vector of coefficients is estimated by the least squares error method as shown in Equation (24).

$$\text{least square error} = \left\{ \begin{array}{ll} \hat{h}(A) = G(a_u, a_v) \\ \text{error} = \sum_{u=1}^{n}(h_u - \hat{h})^2 \\ \frac{d}{dh_i} \text{error} = 0, \ t = 1, 2, 3, 4, 5. \end{array} \right. \hspace{1cm} (24)$$

To facilitate the analysis, the computation of the vector of coefficients was done using matrix form as shown in Equation (25).

$$\hat{h} = (A^T A)^{-1} A^T h$$  \hspace{1cm} (25)
where Equation (26) depicts the input dataset.

\[
A = \begin{bmatrix}
1 & a_{u1} & a_{u1}^2 & a_{u2} & a_{u2}^2 \\
1 & a_{u2} & a_{u2}^2 & a_{u3} & a_{u3}^2 \\
1 & a_{u3} & a_{u3}^2 & a_{u4} & a_{u4}^2 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
1 & a_{un} & a_{un}^2 & a_{un} & a_{un}^2
\end{bmatrix}
\]  

(26)

### 3.3 Least Squares Support Vector Machine

LSSVM is an iteration of the improved standard SVM. It is one of the algorithms that gives important results of statistical learning theory. Using the least squares loss function (LSLF), LSSVM generates optimisation problem and it is based on equality constraints. The LSLF entails linear equation solution and it is considered to be simple compared to that in the \( \epsilon \)-intensive loss function of the novel SVMs (Suykens et al., 2002). In general, LSSVM is employed for classification, regression problems, and optimal control (Gestel et al., 2004; Kaytez et al., 2015).

This section is introducing the least squares support vector regression (LSSVR) briefly. The LSSVR method is used to approximate an incomprehensible function by relying on a training dataset \([x_i, y_i]_{i=1}^{l}\). The regression is formulated as feature space representation shown in Equation (27).

\[
y = f(x) = w^T \phi(x) + b
\]  

(27)

where \( x \in R^n \) for a positive integer \( n \), \( y \in R \), \( w \) is the weight vector of the same dimension as the feature space, \( b \) is a bias and \( \phi(\cdot) : R^n \rightarrow R^n \) is a nonlinear mapping to the high dimensional feature space. The minimisation of the error together with the regularisation is given as in Equations (28) and (29).

\[
\text{Minimise } J(w, b, \epsilon) = \frac{1}{2} w^T w + \frac{\delta}{2} \sum_{i=1}^{l} \epsilon_i^2
\]  

(28)

Subject to \( y_i = w^T \phi(x_i) + b + \epsilon_i; \ i = 1, 2, ..., l \)  

(29)

where \( \epsilon = (\epsilon_1, \epsilon_2, ..., \epsilon_l) \in R^l \), \( \delta \) is the regularised parameter balancing the trade-off between the margin and the error.

The Lagrangian function \( (L_\epsilon) \) of the optimisation problem for Equations (28) and (29) is given as in Equation (30) (Kruif and De Vries, 2003; Yang et al., 2014).

\[
L_g(w, b, \epsilon; \beta) = J(w, b, \epsilon) + \sum_{i=1}^{l} \beta_i \left( y_i - w^T \phi(x_i) - b - \epsilon_i \right)
\]  

(30)

where the Lagrange multipliers which can either be negative or positive in the formulation of LSSVM are \( \beta = (\beta_1, \beta_2, ..., \beta_l) \in R \). The Lagrangian optimality conditions are shown in Equations (31) to (34).

\[
\frac{\partial L_g}{\partial w} = 0 \ \Rightarrow \ w = \sum_{i=1}^{l} \beta_i \phi(x_i)
\]  

(31)
\[
\frac{\partial L_g}{\partial b} = 0 \quad \Rightarrow \quad - \sum_{i=1}^{l} \beta_i = 0 \quad (32)
\]

\[
\frac{\partial L_g}{\partial e_i} = 0 \quad \Rightarrow \quad \beta_i = \delta e_i \quad (33)
\]

\[
\frac{\partial L_g}{\partial \beta_i} = 0 \quad \Rightarrow \quad w^T \phi(x_i) + b + e_i - y_i = 0 \quad \text{for} \quad i = 1, 2, \ldots, l \quad (34)
\]

The linear system (Equation (35)) represents the Lagrangian optimal conditions.

\[
\begin{pmatrix}
0 & e^T \\
e & \Omega + \frac{I}{\delta}
\end{pmatrix}
\begin{pmatrix}
b \\
\beta
\end{pmatrix}
= \begin{pmatrix}
0 \\
Y
\end{pmatrix}
(35)
\]

where \( I \in \mathbb{R}^{r \times l} \) is an identity matrix. \( Y = \begin{pmatrix} y_1 & y_2 & \cdots & y_l \end{pmatrix}^T \), \( \beta = (\beta_1 \beta_2 \cdots \beta_l)^T \) and \( e = (1 1 \cdots 1)^T \).

\( \Omega = (\Omega_{ij}) = k(x_i, x_j) k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \).

The output (Equations (36) and (37)) of the approximator is computed for new input values of \( x \) with \( \beta \) and \( b \) as

\[
\hat{y}(x) = \langle w, \phi(x) \rangle + b = \sum_{i=1}^{l} \beta_i \phi(x_i, \phi(x)) + b = \sum_{i=1}^{l} \langle \phi(x_i), \phi(x) \rangle + b = \sum_{i=1}^{l} \beta_i K(x_i, x) + b
(37)
\]

### 3.4 Generalised Regression Neural Network

GRNN is a variant of radial basis function network (RBFN) which uses a kernel regression network and is used to solve linear or nonlinear approximation problems. The network computes the most probable output that minimises the mean squared error value. The GRNN does not require an iterative training procedure as in backpropagation method, but each layer is passed through forward computation. The method has excellent performance in learning speed and robust function approximation ability. The reason is the method provides rapid convergence to the optimum regression surface by using a probability distribution. (Wang and Peng, 2018; Kissi et al., 2006; He et al., 2021). GRNN comprise input, hidden, summation, and output layers (Figure 4). External information is passed directly to the hidden layer without weighting through the input layer. The output of the \( i \)th neuron in the hidden layer is computed as in Equation (38) (Hou et al., 2022).

\[
h_i(x) = \exp \left( \frac{-||x - c_i||}{2r^2} \right) \quad \text{for} \quad 1 \leq i \leq p
(38)
\]

where \( X = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix} \) is the independent variable of the actual sediment load dataset, \( r \) is the radius of the RBF and it determines the generalisation capability of the GRNN by controlling the degree of smoothness, \( C_i \) is the training input vector, the number of training set is \( p \).
The summation layer comprises simple summation, $S_s$, and weighted summation, $S_w$. The $S_s$ computes the arithmetic sum of the hidden layer output as given in Equation (39).

**Figure 4. GRNN Architecture**

$$S_s = \sum_{i=1}^{p} h_i(x)$$  \hspace{1cm} (39)

The $j$th weighted summation (Equation (40)) calculates the hidden layer outputs weighted sum.

$$S_{wj} = \sum_{i=1}^{p} y_i h_j(x)$$  \hspace{1cm} (40)

where $y_i$ is the interconnection weight in the $i$th desired response. Finally, the output of the $j$th output neuron for GRNN model (Equation (41)) is computed as a weighted average of the desired response.

$$\hat{y}_j = \frac{S_{wj}}{S_s}$$  \hspace{1cm} (41)

### 3.5 Model Building Framework

This section presents a flowchart showing the model development, evaluation, and implementation to predict the SSL. The input and output variables were first selected and data partitioning was performed. To have homogeneity, the data was scaled into a specific interval using the normalisation process. The network was then trained and the optimum model was obtained based on a termination criterion. The optimum trained model was however tested and the respective performance metrics were computed based on the observed and predicted SSL. A detailed presentation of the computational process presented in Figure 5 is given in the subsequent sections.
3.5.1 Data Specification and Variables Selection

In developing the various AI models, 47 sample data set was used. This data was obtained with the help of a sediment rating curve by extracting drainage areas, river slopes and length of rivers from the topographic maps provided by the Survey and Mapping Division of Lands Commission, Ghana. The extracted features served as the independent variables while SSL served as the dependent variable. From the 47-sample data set, 33 (approximately 70%) of the training set was used to develop the AI methods.
prediction models and the remaining 14 data points (approximately 30%) served as the unseen data for the validation of the trained models. The partitioning of data was carried out using the widely used hold-out cross-validation approach.

3.5.2 Normalisation of Data

Data normalisation was performed to minimise the impact of larger input values on the smaller ones during the model process. This helps to put the training set into a common range which improves the convergence speed of the AI during training. Equation (42) was used to carry out the normalisation (Mueller and Hemonnd, 2013).

\[
d_{N} = d_{mn} + \frac{(d_{mx} - d_{mn})(u_{o} - u_{mn})}{(u_{mx} - u_{mn})}
\]

where \(d_{N}\) denotes the normalised data, \(u_{o}\) is the observed SSL, \(u_{mn}\) and \(u_{mx}\) represent minimum and maximum values respectively of the observed SSL with \(d_{mx}\) and \(d_{mn}\) ranged from -1 to 1 respectively.

3.5.3 Network Training

In modeling using ANN, datasets are trained to produce the desired output for a given input. Similarly, in this study, ANN was trained to find the functional relationship between the independent variables (drainage areas, river slopes, and length of rivers) serving as the input and the dependent variable (SSL) as the output. To perform the network training, the 47 datasets were put into a specific range and partitioned thereafter into 33 training and 14 testing subsets. The training set was used as parameterisation (adjustment of weight) to minimise the error function, while the testing set was used for model validation. In the network training, the Levenberg-Marquardt backpropagation algorithm was employed to train the BPNN. During this phase, the network was allowed to train until no additional effective enhancement happened. After training the network, the unseen testing data was used to give a general independent assessment of the performance of the BPNN. In determining the optimum BPNN model, the mean squared error (MSE) of the model was monitored at each stage of training and testing. In addition, the correlation coefficient (R) and coefficient of determination (R²) were used to judge the performance of the ANN model. After several trials, the model with the highest R and R² values with the lowest MSE was selected as the best model (Nandy et al., 2012).

The GMDH is a feed-forward multilayer network of quadratic neurons that are used to map the input-output variables' functional relationship. Thus, the key idea of the GMDH is to find a mapping \(\hat{G}\) as an approximation of the actual function \(G\) for the difference between the actual output \(y\) and the predicted output \(\hat{y}\) to be small as possible. The technique has a faster learning speed, converges to the optimal nonlinear or linear regression surface, and has good approximation capability. This phenomenon is achieved because the GMDH relies on optimisation technique that determines the optimal structure automatically by a layer-by-layer pruning process based on the mean square error criterion (MSE). The GMDH network automatically stops adding layers the moment the MSE of the proceeding layer exceeds the preceding layer. In this case, the network selects the lowest MSE component in the highest layer as its final model outcome (Srinivasan, 2008).

The LSSVM which is a variant of SVM adopts equality constraints. The LSSVM technique is to fit a functional model \(y(\mathbf{x})\) on the training data sets such that the function could be used to infer the target \(y\) for a new input data point \(\mathbf{x}\) later.

The training process involves two linear systems with an identical positive definite coefficient matrix, followed by the application of the conjugate gradient method. Thus, the underlying optimisation problem follows a system of a linear equation and this improves the training efficiency for large-scale learning tasks (Liu et al., 2013; Xia, 2018).

The GRNN is based on a standard statistical technique known as kernel regression. Considering the training, the output for the input is computed in two steps. Firstly, the hidden layer produces a set of weights associated with the closeness of the input vector to the training patterns. Here, the weighted sum is one and it represents the contribution of every training pattern to the final result. Secondly, the output layer computes the output as the sum of the product of the weights and the targets. The GRNN technique approximates any arbitrary function between the input and output vectors, drawing the function estimate directly from the training data. That is, as the training set increases in number, the estimated error approaches zero with only mild restrictions on the function (Cigizoglu and Alp, 2006).

3.5.4 Model Performance Evaluation

This study used root mean square error (RMSE), Percentage Root-Mean Square Error (PRMSE), Uncertainty at 95%, root mean square error observations standard deviation ratio (RSR), and Legates and McCabe’s (Elm) (Willmott and Matsuura, 2005;
Legates and McCabe, 2013; Tian et al., 2016) statistical indices to determine the efficiency of the developed models for sediments load prediction.

**Root Mean Square Error**

The RMSE (Equation (43)) is a dimensioned measure of average model metric. The metric expresses average model-prediction error in the units of the variable of interest.

\[
RMSE = \sqrt{\frac{\sum_{j=1}^{\tau} (O_j - P_j)^2}{\tau}}
\]  

(43)

where \( \tau \) is the test observations size, \( O \) and \( P \) are the test observation and prediction values respectively.

**Percentage Root-Mean Square Error**

The PRMSE (Equation (44)) which is capable of evaluating the precision of a models’ predictive performance is a scale-independent measure.

\[
PRMSE = \frac{RMSE}{x_m} \times 100
\]  

(44)

**Uncertainty at 95%**

\( U_{95} \) (Equation (45)), the uncertainty at 95% confidence level, is a statistical analysis indicator that reveals more information about the developed model prediction deviations from the actual observations.

\[
U_{95} = 1.96\sqrt{S_d^2 + \left(\frac{RMSE}{x_m}\right)^2}
\]  

(45)

where \( S_d \) is the standard deviation between the prediction and the observation values, 1.96 is the coverage factor corresponding to 95% confidence level, RMSE is the Root-Mean Square Error.

**Root Mean Square Error Observations Standard Deviation Ratio**

The RSR (Equation (46)) which depends largely on the RMSE varies from an optimal value of 0 to a large positive value. The smaller the RMSE value, the least RSR value becomes. Hence, the better the predictive power of the developed model.

\[
RSR = \frac{RMSE}{S_d} = \frac{\sum_{i=j}^{\tau} (O_j - P_j)^2}{\sqrt{\sum_{i=j}^{\tau} (O_j - O_m)^2}}
\]  

(46)

where \( S_d \) is the standard deviation of the test observations, \( \tau \) is the test observations size, \( O \) and \( P \) are test observation and prediction values respectively. \( O_m \) is the mean of the observation values.

*Legates and McCabe*
Artificial Intelligence and Applications  Vol. XX Iss. XX yyyy

The $E_{LM}$ (Equation (47)) reveals the correlation between the predictions and the observed values. Closer the $E_{LM}$ value to 1 is an indication of the developed model having strong predictive power.

$$E_{LM} = 1 - \frac{\sum_{i=1}^{r} Abs(p_j - o_j)}{\sum_{i=1}^{r} Abs(o_j - o_m)}$$ (47)

where $O$ and $P$ are test observation and prediction values respectively, $O_m$ is the mean of the observation values, $τ$ is the test observations size.

5. Numerical Application

5.1 Data Used

Topographic maps from Survey and Mapping Division of Lands Commission, Ghana, West Africa and sediment rating curve were used to extract the data for the analysis (Figures 6). Drainage areas, river slopes, and length of rivers served as independent variables whiles suspended sediment load served as the dependent variable. Table 1 shows the statistical summary results of the data used to develop the suspended sediment load prediction models.

![Figure 6. Topographic Map of the Study Area](image)

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
</table>

1 cm = 25 km
5.2 Optimum Model Developed

The BPNN optimum trained model had a model structure comprising three inputs, eight hidden neurons and one output, that is [3-8-1]. For the GRNN, the optimum model that produced the best performance had a smoothing parameter of 2.6 with three inputs, and one output. In training the LSSVM, the best values for regularization parameter (gamma) and the width of the kernel function (sigma) were 34177922.666 and 30.2519838598, respectively. The optimum GMDH model consisted of two layers and single neuron. In the GMDH training process, two input variables (drainage areas and river slopes) were automatically selected and seen to be very relevant in SSL prediction. This showcases the feature extraction capability of the GMDH model. The final GMDH model for SSL prediction is shown in Equation (48).

\[
X_4 = -1.73313143729 - 0.00170794657055 X_2 + 1.1237213778 X_1 \\
- 3.49331727402 \times 10^{-05} X_1 X_2 + 1.76142505781 \times 10^{-07} X_2^2 \\
+ 0.00102950304895 X_1^2
\] (48)

Final GMDH Model

\[
= 1.02012760202 + 1.00280017581 X_4 - 0.154325304254 X_1 \\
- 0.00558111412937 X_1 X_4 + 0.00102456552567 X_4^2 \\
+ 0.00644401995933 X_1^2
\]

5.3 Developed Model Efficiency Assessment

In developing the models (BPNN, GMDH, LSSVM, and GRNN) the dataset acquired was partitioned into 70% training set (32 data points) and the remaining 30% as testing set (13 data points). The training set was used to fit the models and the testing set served as an independent data to authenticate the forecasting strength of the developed models. To predict the suspended sediment load, model assessment was performed for the competing methods BPNN, GMDH, LSSVM, and GRNN to find not only how the best method fits the dataset well, but how it will work in future applications. This was accomplished by using the five test-case performance indices RMSE, PRMSE, U95, RSR, and ELM as shown in Table 2.

The ELM ideal value is 1 and it is an indication that the model is having a strong predictive strength. From Table 2, the GRNN model had the best ELM value of 0.9637. This means that 96.37% of the total variability in the suspended sediment load predicted was explained by the independent variables (drainage areas, length of rivers, river slopes) used to develop the model. In other words, the GRNN model ELM value of 96.37% shows clearly that the GRNN model predictions are closely related to the observed suspended sediment load. This close association can additionally be viewed in Figures 5 and 6. Consequently, the GRNN model does not only fit the observed data very well, but it has a strong predictive power as well.

The RMSE, PRMSE, U95, and RSR indices show the models’ bias in the prediction of the suspended sediment load. The smaller these indices values are, the better the acceptable accuracy of the developed model. From Table 2, RMSE, PRMSE, U95, and RSR values for the GRNN method are 2.6779, 9.3594, 7.1446, and 0.0327 respectively. This is an indication that the GRNN method fitted the data very well than the other competing methods based on their index’s values.

From Table 2, the intercomparison among the methods employed was also perceptible. The GRNN method had the best RMSE value of 2.6779 followed by LSSVM, BPNN, and GMDH with 3.1522, 4.5127, and 7.2736 values respectively. The performances of the methods in descending order for PRMSE are GRNN, LSSVM, BPNN, and GMDH with values 9.3594, 11.0171, 15.7721, and 25.4217 respectively. In the case of U95, GRNN method had the least value of 7.1446 followed by LSSVM, BPNN, and GMDH with values 8.8422, 12.3461, and 20.2807 respectively. For RSR, the best method in a descending order is GRNN, LSSVM, BPNN, and GMDH with values 0.0327, 0.0385, 0.0551, and 0.0888 respectively.

| Drainage areas (sq km) | 0.8860 | 873.0910 | 56.1632 | 151.8754 |
| River slopes (m/km)    | 4.0105 | 129.3718 | 29.4597 | 26.0539 |
| Length of rivers (m)   | 124.0000 | 3808.7000 | 929.4156 | 994.2973 |
| Suspended sediment load (mg/L) | 0.4368 | 12665.8140 | 327.5531 | 1885.3659 |

Table 1. Summary of dataset
The reason for the accurate forecasting ability of the GRNN model is based on the reliability of its feed forward additional summation layer in selecting the best output which is used for the weighted average computation for the desired response (Kang et al. 2019; Jian et al. 2019).

To determine further the accuracy of the four ANN methods (BPNN, GMDH, LSSVM, and GRNN) the line graph (Figure 7) was employed. The line graph is very useful when assessing the performance of many methods as it graphically summarises the methods into a single plot and allowing methods comparison with the observed dataset. The methods performances are expressed in terms of their standard deviations from the 45° diagonal line. Thus, the standard deviation is proportional to the method’s distance from the 45° diagonal line.

From Figure 7, clearly the simulation point of the GRNN model is closer to the observed 45° diagonal line than any of the other methods used. This implies that there are similarities in the GRNN model predictions and the observed dataset in terms of its obtained lowest standard deviation (least distance from the 45° diagonal line). This shows that the GRNN method had quality in simulating the test data well than any of the other methods studied.

<table>
<thead>
<tr>
<th>Model</th>
<th>RMSE</th>
<th>PRMSE</th>
<th>RSR</th>
<th>$E_{LM}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPNN</td>
<td>4.5127</td>
<td>15.7721</td>
<td>12.3461</td>
<td>0.0551</td>
</tr>
<tr>
<td>GMDH</td>
<td>7.2736</td>
<td>25.4217</td>
<td>20.2807</td>
<td>0.0888</td>
</tr>
<tr>
<td>LSSVM</td>
<td>3.1522</td>
<td>11.0171</td>
<td>8.8422</td>
<td>0.0385</td>
</tr>
<tr>
<td>GRNN</td>
<td>2.6779</td>
<td>9.3594</td>
<td>7.1446</td>
<td>0.0327</td>
</tr>
</tbody>
</table>

Figure 5. AI Developed Models Prediction
Figure 6. AI Models Prediction Errors

Figure 7. Models Performance Assessment
5.4 Validation of Results with Literature

It is of great importance to validate the present study’s results with the established research results in the literature. In that regard, a review of literature from 2019-2023 (5 years) has been made, with two papers reviewed each year (Table 3). As per the dataset range, the absolute error metrics might vary from one case study to another. Consequently, the coefficient of determination value ($R^2$) was selected for validation. For example, Latif et al. (2023) developed an AI model to predict sediment load. The maximum $R^2$ value realised from the developed models ranges from 0.79 to 0.91. In Keshtegar et al. (2023), the prediction of sediment yields using a data-driven model achieved an $R^2 = 0.72$, RMSE = 0.51, and MAPE = 11.99%. Furthermore, several AI models which include ANN-GA, SEA/Balance, ITD-EPR, WATEM/SEDEM, ElasticNet LR, MLP, EGB, LSTM, RS, SVM-RBF, SVM-NPK, RF, MM-ANNs, FNN-PSOGSA, FNN-PSO, FNN, ANFIS, WM5, WANN, LSTM and M5T have also been developed. Based on the revealed prediction accuracy from the literature and in comparison, with this study, the GRNN was able to obtain comparable predictability performance as evident in Table 2.

<table>
<thead>
<tr>
<th>Authors</th>
<th>Paper Title</th>
<th>Methods Applied</th>
<th>Validation Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latif et al. (2023)</td>
<td>Sediment load prediction in Johor River: deep learning versus machine learning models.</td>
<td>LSTM, ANN, SVM</td>
<td>$R^2$ values ranges from 0.79 to 0.91</td>
</tr>
<tr>
<td>Keshtegar et al.</td>
<td>Prediction of sediment yields using a data-driven</td>
<td>RM5T</td>
<td>RM5T ($R^2 = 0.72$), RM5T (RMSE = 0.51), RM5T (MAPE = 11.99%).</td>
</tr>
<tr>
<td>Yadav et al. (2022)</td>
<td>Optimised scenario for estimating suspended sediment yield using an artificial neural network coupled with a Genetic Algorithm</td>
<td>ANN-GA</td>
<td>ANN-GA ($R^2 = 0.8710$), ANN-GA (RMSE = 0.0088)</td>
</tr>
<tr>
<td>Maltsev et al. (2022)</td>
<td>Assessment of net erosion and suspended sediments yield within river basins of the agricultural belt of Russia.</td>
<td>SEA/Balance, WATEM/SEDEM</td>
<td>SEA/Balance ($R^2 = 0.78$), WATEM/SEDEM ($R^2 = 0.79$)</td>
</tr>
<tr>
<td>Zhao et al., 2021</td>
<td>A decomposition and multi-objective evolutionary optimisation model for suspended sediment load prediction in rivers.</td>
<td>ITD-EPR</td>
<td>ITD-EPR ($R^2 = 0.92$), ITD-EPR (WI = 0.93)</td>
</tr>
</tbody>
</table>
Artificial Intelligence and Applications  Vol. XX Iss. XX yyyy

<table>
<thead>
<tr>
<th>Authors</th>
<th>Methodology</th>
<th>Performance Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ElasticNet LR ($R^2 = 92.01$), MLP ($R^2 = 96.56$), EGB ($R^2 = 96.71$), LSTM ($R^2 = 0.9945$)</td>
</tr>
<tr>
<td>Nhu et al. (2020)</td>
<td>Monthly suspended sediment load prediction using artificial intelligence: testing of a new random subspace method.</td>
<td>RS, SVM-RBF, SVM-NPK, RF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RS (NSE = 0.83), SVM-RBF (NSE = 0.80), SVM-NPK (NSE = 0.78), RF (NSE = 0.68)</td>
</tr>
<tr>
<td>Meshram et al., (2020)</td>
<td>Application of artificial neural networks, support vector machine and multiple model-ANN to sediment yield prediction.</td>
<td>MM-ANNs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MM-ANNs ($R^2 = 0.921$), MM-ANNs (NSE = 0.74), MM-ANNs (RAE = 0.360)</td>
</tr>
<tr>
<td>Meshram et al. (2019)</td>
<td>New approach for sediment yield forecasting with a two-phase feedforward neuron network-particle swarm optimisation model integrated with the gravitational search algorithm.</td>
<td>FNN-PSOGSA, FNN-PSO, FNN, ANFIS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FNN-PSOGSA (NSE = 0.612), FNN-PSO (NSE = 0.500), FNN (NSE = 0.331), ANFIS (NSE = 0.244), FNN-PSOGSA (WI = 0.832), FNN-PSO (WI = 0.771), FNN (WI = 0.692), ANFIS (WI = 0.726)</td>
</tr>
<tr>
<td>Nourani et al. (2019)</td>
<td>A wavelet-based data mining technique for suspended sediment load modeling.</td>
<td>WM5, WANN, M5T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WM5 (NSE = 0.94), WANN (NSE = 0.89), M5T (NSE = 0.77)</td>
</tr>
</tbody>
</table>


5.5 Research Implications

Although rivers have natural sediment transports that vary with time, many significant threats to these rivers arise primarily from human activities such as climate change, pollution, landscape changes, and urbanisation; of which the excess of such phenomenon causes river channels to become unstable and flood capacity decreased due to infilling. Therefore, it is imperative to know the level of SSL because a higher level can cause flash floods in the advent of rains causing extensive damage to property, the social well-being of people, and human lives. As a result, developing a model to predetermine the level of SSL is vital. The GRNN model approach of predicting SSL can help governments and policymakers to formulate appropriate measures to prevent water/river pollution and this would reduce SSL levels. This would lead to substantial improvements in water quality and maintain the ecosystem that serves as the habitat for many living organisms. This is achievable because the accurate SSL prediction is very useful since it brings solutions to the devastating natural events (flood hazards) occurring around the world which are partly caused by SSL accumulation on river banks and cause these flash floods.

6. Conclusions

In this study, four AI models BPNN, GMDH, LSSVM, and GRNN have been developed for accurate SSL prediction based on case study data obtained from Ghana. The statistical analysis results showed that the developed AI models are good and can be used to predict SSL based on their performance indicators. The BPNN method had RMSE (4.5127), PRMSE (15.7721), $U_{05}$ (12.3461), RSR (0.0551), and $E_{LM}$ (0.9485). For the GMDH, 7.2736, 25.4217, 20.2807, 0.0888, and 0.9418 were produced...
correspondingly. On the contrary, the LSSVM had 3.1522, 11.0171, 8.8422, 0.0385, and 0.9476 for RMSE, PRMSE, U95, RSR and EL_M. The respective values obtained by the GRNN were 2.6779, 9.3594, 7.1446, 0.0327, and 0.9637. However, the GRNN has proven to be the best method suitable for predicting SSL based on its low RMSE, PRMSE and U95 achieved values, and high EL_M value when compared with the other contending methods.

The predictive power of the BPNN, GMDH, LSSVM, and GRNN models was also presented visually using a line graph. This revealed how closely the GRNN model prediction results were to the observed SSL and how its performance is comparable to the other models. Based on all the statistical results, it is clear that the GRNN method has confirmed good learning and generalisation power as compared with the other methods. Therefore, the developed GRNN method can provide reliable guidance to researchers and policymakers in Africa.

The study findings offer several avenues for future research to extend and strengthen prediction-oriented model assessment and comparison in SSL prediction. The possible challenges or restrictions that could impede the model’s performance are data quality and its associated characteristics. Hence, fine-tuning the hyperparameters of the AI model using metaheuristic optimisation algorithms must be considered for future analysis. This will help overcome the manual setting of the hyperparameters in the model development phase. In addition to that, environmental and climatic factors should also be considered in the future when developing SSL prediction models for different jurisdictions.

Conflicts of Interest

The authors have no conflicts of interest to declare.

References


