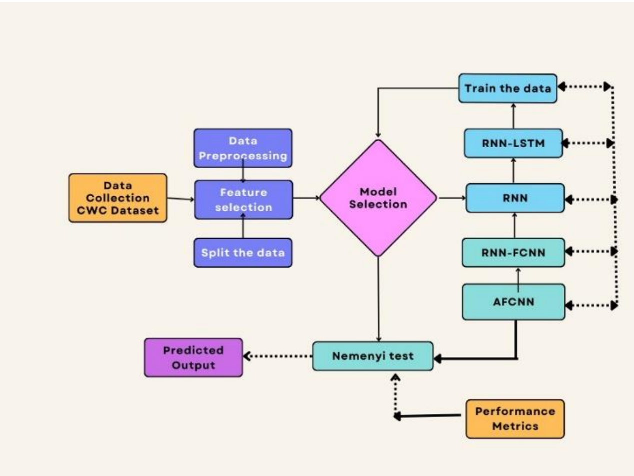
Global NEST Journal, Vol 26, No X, pp 1-14

Sustainable development in reservoir sedimentation load estimation from statistical analysis of datasets to deep learning algorithms

Umarajeswari Subramaniyan[1], Ilangovan Rengasamy[2] and Ramasamy Sankar Ram Chellappa[2]*
[1]MIET Engineering College, India
[2]Anna University, India
Received: 27/03/2024, Accepted: 09/11/2024, Available online: 20/11/2024
*to whom all correspondence should be addressed: e-mail: csankarraam@gmail.com
https://doi.org/10.30955/gnj.005971

Graphical abstract



Abstract

The process of sedimentation in reservoirs is an important issue that must be addressed for effective water resource management. Traditional methods for predicting sedimentation are complex and require extensive data inputs. In this study, we innovate by integrating four distinct neural network architectures into a cohesive approach for prediction. Specifically, Recurrent Neural Networks (RNN), RNN combined with Long Short-Term Memory (RNN-LSTM), RNN combined with Fully Connected Neural Networks (RNN-FCNN), and Autoencoder integrated with Fully Connected Neural Networks (A-FCNN) were utilized to estimate sediment volume based on their performance. This approach is faster, more accurate, and can handle multiple inputs. Additionally, Nemenyi and Principal Component Analysis (PCA) tests were conducted to build more robust and reliable models for estimating sediment volume. These computational methods were tested on real-world data, and the models demonstrated strong predictive performance for a specific reservoir located in Theni, Tamil Nadu. The inputs selected for the model included the following factors such as: ·Water level storage capacity ·Temperature ·Precipitation ·Wind speed ·Solar radiation ·Vapour pressure ·Inflow ·Outflow and ·Runoff. For this study, a dataset spanning twenty years was utilized, focusing on the Vaigai Reservoir located on the Vaigai River in Theni. A classification system based on performance indices, such as Root Mean Square Error (RMSE), Mean Square Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), was proposed and applied to identify the best-performing model. The A-FCNN combination achieved the highest accuracy across the datasets, with a predictive accuracy of 99.78% for the test data. Predicting sediment volume helps to improve water resource management, contributing to sustainable development and environmental protection.

Keywords: Sedimentation, Reservoir, Deep Learning, Prediction, Nemenyi, Statistical Analysis

1. Introduction

Reservoirs play a critical role in water resource management, providing a reliable source of water for irrigation, power generation, and other essential functions. However, sedimentation can significantly impact the functionality and operation of a reservoir, leading to reduced storage capacity, decreased water quality, and other environmental issues. Traditional techniques for estimating sedimentation volume can be complex and require extensive data inputs, which can be challenging to obtain. Thus, the use of advanced statistical analysis and deep learning algorithms can offer a more accurate, efficient, and effective approach for predicting sedimentation volume in reservoirs. The application of deep learning algorithms in water resource management has been gaining momentum in recent years driven by their capacity to analyze vast datasets and forecast outcomes with precision.

Conventional approaches to reservoir sediment volume prediction sometimes demand large amounts of data collecting and are labor-intensive. Using a comparative analysis of four deep learning architectures—Recurrent

Neural Networks (RNNs), RNNs with Long Short-Term Memory (RNN-LSTMs), RNNs combined with Fully Connected Neural Networks (RNN-FCNNs), and Autoencoders integrated with Fully Connected Neural Networks (A-FCNNs)—this work presents a unique approach for sediment volume estimate). This method makes use of deep learning's capacity to manage intricate interactions between sediment volume and several environmental variables (water level, temperature, precipitation, etc.). Using a mix of error measurements (RMSE, MSE, MAE, MAPE) and statistical methods (Nemenyi test, PCA), we evaluate these models on real-world data from the Vaigai Reservoir in Theni, Tamil Nadu. This work not only shows how well deep learning performs for reservoir management but also names A-FCNNs as the most appropriate model for precise sediment prediction, hence advancing sustainable reservoir management methods.

In this review of related works, various studies on sediment prediction and transport at different locations are presented. Several studies specifically focus on the application of neural networks for sediment estimation. For instance, Hassan *et al.* (2022) evaluated a neural network model to estimate sediment deposition levels at the Tarbela reservoir, while Cigizoglu *et al.* (2002) focused on estimating and forecasting suspended sediment using artificial neural networks. Dibike *et al.* (1999) encapsulated numerical-hydraulic models using artificial neural networks, and Feyzolahpour *et al.* (2012) also employed neural networks to estimate suspended sediment concentration. Chen *et al.* (2013) applied a neural network approach to model rainfall-runoff caused by typhoons, and Goh *et al.* (1995) modeled complex systems using backpropagation-induced neural networks. Furthermore, Jothiprakash *et al.* (2008) combined conventional techniques with neural network models to estimate reservoir trapping efficiency, while Babanezhad *et al.* (2021) employed an adaptive neuro-fuzzy inference system (ANFIS) to simulate suspended sediment load in a river system. Similarly, Ehteram *et al.* (2019) explored the potential of combining various artificial intelligence models for improved results in sediment prediction.

In contrast, a different set of studies emphasize sediment transport mechanisms and the environmental factors affecting sedimentation. Sayah *et al.* (2019) investigated the effects of man-made ponds on soil erosion and sediment transport, and Arfan *et al.* (2019) analyzed the temporal and spatial variability of flow in the Indus River. Chen *et al.* (2006) explored the temporal variations in fine suspended sediment concentration in the Changjiang River estuary and adjacent coastal waters, while Vente *et al.* (2005) focused on forecasting sediment yield and soil erosion at the basin level. Di Francesco *et al.* (2016) characterized flood events through sediment analysis, and Licznar *et al.* (2003) employed artificial neural networks to predict soil erosion and runoff at the plot level.

Other studies concentrate on specific case applications and methodologies in sediment prediction. For example, Lee *et al.* (2006) provided a numerical assessment of sediment accumulation in reservoirs following three typhoon events, using a neural network model to estimate sedimentation. Shangle *et al.* (1991), Yoon *et al.* (1992), and Yitian *et al.* (2003) studied sedimentation status in reservoirs and the transport of sediment in river systems, proposing artificial neural network-based methods for sediment removal. Sharma *et al.* (1991) analyzed water and sediment yields from the High Himalayas into the Satluj River, while Srinivasulu and Jain (2006) compared different training methods for artificial neural network-based rainfall-runoff models. Goh *et al.* (1994) also evaluated the potential for seismic liquefaction using neural networks.

Together, these studies illustrate the versatility and growing importance of neural networks and artificial intelligence in sedimentation prediction and reservoir management. This body of research not only contributes to advancements in sediment modeling techniques but also highlights the potential of combining machine learning with traditional sediment transport studies to enhance water resource management.

Here by using four standalone deep learning algorithms can potentially improve prediction accuracy and robustness in the dataset. Accordingly, the models have been explored as alternatives, providing a range of options for model selection depending on data availability and computational resources. The data analysis techniques like Principal Component Analysis (PCA) helps to reduce dimensionality and address multicollinearity have been used to prepare data for modeling. The models are trained on the prepared data. To evaluate the performance of each model on a validation set using appropriate metrics the Nemenyi test is applied to the validation set results. This test is to statistically compare the performance of these models. Also, this helps to identify which models perform significantly better or worse than others.

These statistical analyses can provide valuable insights into the performance of deep learning models and help identify areas for improvement. This article will discuss the importance of statistical analysis and deep learning algorithms in sedimentation prediction and explore their applications in reservoir management. Specifically, we included the data description, experimental design, materials and methods, deep learning algorithms used for sedimentation prediction, results and discussions, and conclude with insights and recommendations for future research.

The overall objectives of the study are: (1) To develop the four neural network models namely RNN, RNN-LSTM, RNN-FCNN, and A-FCNN using 20-year datasets. (2) To compare the prediction accuracy of four models by using metrics like RMSE, MAE, MSE, and MAPE; (3) To identify which models perform significantly better or than others by using the Nemenyi test.

2. Data description

2.1. Value of the Data

This dataset provides valuable insights into the long-term behavior of a reservoir, making it an essential resource for studying trends and developing predictive models for sediment load and other critical parameters. Collected over a 20-year period (2000-2020), the data allows researchers to analyze the interactions between various environmental and hydrological factors, contributing to more informed decision-making in water resource management. The dataset includes key features such as water level, storage capacity, maximum and minimum temperature, precipitation, wind speed, solar radiation, vapor pressure, inflow, outflow, sediment load, and runoff, all of which are critical in understanding sediment dynamics and reservoir performance. The detailed specifications of the dataset are provided in Table 1. Incorporating this dataset into predictive models can help forecast sediment load and other important factors, aiding in the management and sustainability of reservoirs. A visual representation or map of the reservoir's location, combined with this data, would further enhance comprehension and contextualize the significance of these features.

Table 1. Specifications of Data

Area	Sedimentation modeling
Type of Data	Integer and Floating-point numbers
Description	Features for prediction
Features	Water Level (m), Storage (BCM), tmax(degC), tmin(degC), ppt(mm), ws(mps), srad(W/m^2), vap(kPa), Inflow (Mm^3), Outflow (Mm^3), Sediment Load (MT) and Runoff (Mm^3)
Target variable	Sediment Load (MT)
Data source location	Vaigai Reservoir
Coordinates	10.0128° N, 78.1583° E

The Vaigai reservoir is located in the Madurai district of Tamil Nadu, India. Its coordinates are approximately 10.0128°N, 78.1583° E. The reservoir intercepts a catchment area of 2,253 square kilometers and consists of a 3,243-meter-long earthen dam and a 232-meter-long masonry dam with a maximum height of 33.83 meters. It has a gross storage capacity of 194.78 million cubic meters and can irrigate an area of 9,650 hectares.

The Vaigai River basin is characterized by hard crystalline rock masses of Archaean age on its western side, while the eastern side is composed of sedimentary rocks of upper Gondwana, Tertiary, and Quaternary age. The basin has a tropical climate with an average rainfall of about 850 mm, which varies from region to region. The temperature varies with the region and ranges from around 25° C in January to 27.5° C to 35° C in May. The accurate prediction of sedimentation volume in the Vaigai reservoir is crucial for proper reservoir management and sustainable water resource use (Guerrero *et al.* 2016; Shangle, A. K. 1991). **Figure 1** illustrates the study area map of the Vaigai reservoir.

3. Experimental design, materials and methods

This section includes the overall flow of the proposed work. The overall flow depicted in **Figure 2** compares two approaches for predicting sediment volume: RNN-based and Autoencoder-based. RNNs (including LSTMs) capture temporal patterns in sediment data, while FCNNs learn complex relationships. The Autoencoder pre-trains on the data to extract informative features fed to an FCNN for prediction. Both approaches aimed to improve sediment volume prediction accuracy.

3.1. Statistical analysis of datasets

Effective statistical analysis is essential for extracting valuable insights from the dataset and forms the backbone of this study's approach to understanding sediment dynamics in reservoirs. By conducting comprehensive statistical analysis, researchers can identify key patterns, relationships, and trends within the data, which are essential for developing accurate sedimentation models and informed management strategies. Descriptive statistics are a fundamental tool in this process, providing a clear overview of the data's main characteristics and guiding subsequent analyses. In this study, thorough statistical analysis is crucial for improving our understanding of sediment behavior and its impact on reservoir performance. Several statistical analyses must be performed on the dataset before it is used in deep learning algorithms to ensure data quality and integrity. Data preprocessing techniques, such as cleaning the data and addressing missing values, are applied to prepare the dataset for accurate model training and prediction.

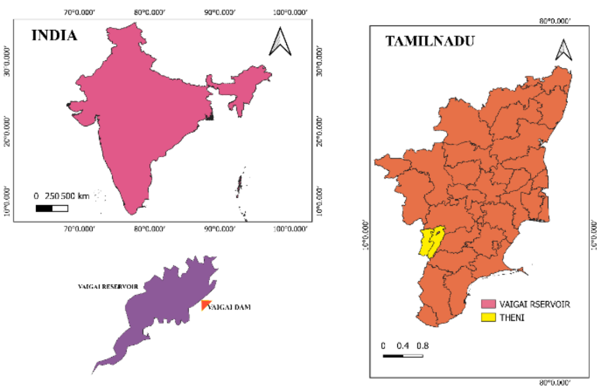


Figure 1. Study area map of Vaigai reservoir

3.2. Data distribution

The importance of data distribution lies in the fact that it allows to understand the underlying patterns and characteristics of the data. By examining the distribution of each feature, we can identify any outliers or skewed distributions that may need to be addressed. Outliers may indicate data entry errors or extreme values that can affect the accuracy of the analysis, and skewed distributions of reservoir data can affect the validity of certain statistical tests (Vente *et al.* 2005; Francesco *et al.* 2016). Measures like histograms, density plots, and quantiles help us visualize the shape and distribution of the data.

3.3. Correlation analysis

Correlation analysis is a non-parametric test based on Spearman's Rank. It can be used with ordinal data where the difference between ranks is more important than the actual values. This is one of the valuable techniques for identifying the relationships between pairs of features in dataset. By analyzing the correlation matrix, we can gain insights into which features are strongly correlated and which are weakly correlated which can inform the selection of appropriate deep learning algorithms and feature selection techniques. For instance, characteristics exhibiting significant correlation might be redundant and thus could be eliminated to enhance both the efficacy and comprehensibility of the model. (Hassan *et al.* 2022; Srinivasulu *et al.* 2006; Sharma *et al.* 1991). The heat map is used to visualize the correlations between multiple variables using color intensity.

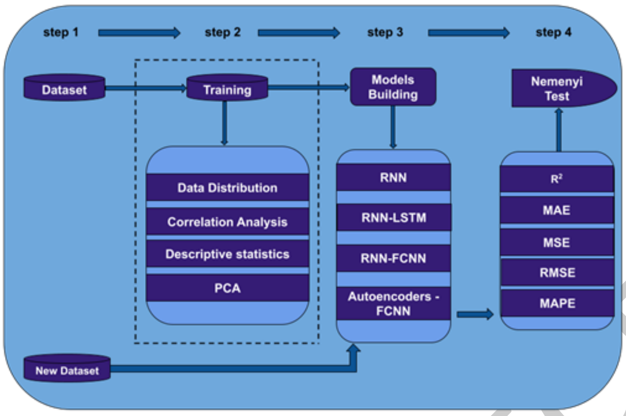


Figure 2. Deep Learning Architectures for Reservoir Sedimentation Prediction

3.4. Descriptive statistics:

Descriptive statistics are a fundamental tool in analyzing and summarizing large datasets. They are especially important in hydrology and environmental science, where vast amounts of data are collected from various sources, including reservoirs. Descriptive statistics allow us to evaluate central tendency measures (mean, median, mode) and variability (range, standard deviation, coefficient of variation, variance, interquartile range, standard error, skewness, and kurtosis) within the data. Additionally, they help to identify and handle missing data by implementing strategies such as imputing with the mean or median. This information is essential for developing accurate models and making informed decisions related to water resource management, sedimentation control, and other environmental issues. In this context, descriptive statistics are used to summarize and analyze reservoir sediment datasets, providing valuable insights into the dataset's characteristics. These insights play a critical role in informing strategies to mitigate the impacts of reservoir sedimentation.

3.5. Principal Component test:

The Principal Component Analysis (PCA) helps identify and address issues with multicollinearity in the dataset. This occurs when input features are highly correlated with each other, which can negatively impact model performance. This application should perform before training the models. It can help to reduce dimensionality by identifying a smaller set of uncorrelated features that capture most of the relevant information from the original data. This can improve model training efficiency and potentially reduce overfitting.

4. Deep learning algorithms for sedimentation prediction

Deep learning algorithms, with their capacity for handling complex, non-linear relationships in large datasets, offer a powerful solution for predicting sediment volume in reservoirs. The challenge in sedimentation prediction lies in capturing the intricate interactions between environmental variables and sediment accumulation. Deep learning models excel in addressing this complexity by learning from vast amounts of data, continuously refining their predictions over time. This makes them highly valuable for long-term sedimentation forecasting and reservoir management. However, the use of deep learning algorithms presents specific challenges. These models require large quantities of high-quality data to perform optimally, and expertise in data preprocessing and model selection is critical to their success. Moreover, model interpretability can be particularly demanding. The black-box nature of deep learning often makes it difficult to understand how predictions are made, complicating the task of explaining the model's reasoning. To overcome these challenges—such as data scarcity and the need for transparent predictions—researchers must focus on curating robust datasets and employing techniques that enhance model interpretability. Despite these hurdles, deep learning remains a promising approach for advancing sedimentation prediction efforts (Hassan *et al.* 2022; Olden *et al.* 2004).

To overcome these challenges, researchers are exploring new approaches to deep learning, such as transfer learning and explainable Artificial Intelligence (AI). Transfer learning refers to the practice of utilizing pre-trained models for a similar problem and fine-tuning them for sedimentation prediction, thus minimizing the need for extensive amounts of data. This aims to improve the interpretability of deep learning models by providing explanations for the model's predictions. Another area of research is the integration of statistical analysis and deep learning algorithms to improve sedimentation prediction. By combining these two approaches, researchers can leverage the strengths of both techniques, such as identifying critical variables through statistical analysis and using deep learning algorithms to model complex relationships and make accurate predictions. Moreover, deep learning algorithms can also help in predicting sediment transport patterns, which can aid in predicting sedimentation volumes. These algorithms can analyze complex sediment transport models and identify patterns in sediment transport, which can be used to predict sedimentation volumes accurately (Raghuwanshi *et al.* 2006; Sarangi *et al.* 2005). Finally, while deep learning algorithms have shown great potential for sedimentation prediction, more research is needed to fully explore their capabilities and limitations and develop reliable and interpretable models for reservoir management. The four standalone algorithms RNN, RNN-LSTM, RNN-FCNN and A-FCNN were explained below

4.1. Recurrent Neural Networks (RNN):

RNN offer a great approach for analyzing and modeling water resources data that involves temporal sequences. However, their suitability depends on the specific task, data availability, and computational resources. RNN can capture temporal relationships within these sequential data. This allows the model to perform past events might influence future water resources conditions, like prediction.

4.2. Long Short-Term Memory Networks (LSTM):

LSTM (Long Short-Term Memory) is one of the RNN variants. It can effectively learn long-term dependencies. These models were predicting the long-term reservoir sedimentation based on historical sediment loads. This leads modeling the impact of climate change on water availability patterns over extended periods. LSTMs are concentrated for handling time series data with complex patterns and dependencies (Lee *et al.* 2019).

4.3. Autoencoders Networks

Autoencoders networks are a type of neural network that is trained to reconstruct its input data, and has become a widely used method in various fields of research including processing sediment data. It can be used for dimensionality reduction, anomaly detection, and even time series prediction. In the case of time series data, the autoencoders would be trained to predict the next time step in the sequence, which can then be used as a basis for predicting sediment load. The suitability of each algorithm depends on the specific characteristics of the reservoir data and the problem we are trying to solve. The experiments in this study need to experiment with different models and hyper parameters to find the best approach.

4.4. Fully Connected Neural Network (FCNN):

In deep learning, FCNN are stacked together with multiple hidden layers to create deep neural networks capable of handling complex problems. It can handle various input and output data types, making them adaptable to different prediction tasks. They can discover intricate relationships between various input features and the target variable. This allows them to predict sediment volume in an efficient way.

4.5. Integration of standalone algorithms:

This integrated architecture of RNN-LSTM leverages the strengths of both RNNs and LSTMs. It can capture sequential information and handle long-term dependencies within the sequence data. Integration of RNN for capturing temporal dependencies and FCNN (RNN-FCNN) for learning spatial features offers a comprehensive approach for sediment load prediction, leveraging both temporal and spatial information. Utilizing Autoencoders for feature extraction and FCNN (A-FCNN) for prediction enables an end-to-end learning framework, facilitating efficient representation learning and accurate sediment load estimation.

4.6. Integrated Approaches:

These novel combinations enhance sediment load prediction models by harnessing the strengths of different architectures, leading to improved accuracy and robustness in reservoir sedimentation management.

4.6.1. RNN-LSTM

The combined RNN-LSTM followed by an FCNN for predicting the output can be represented in **Figure 3**.

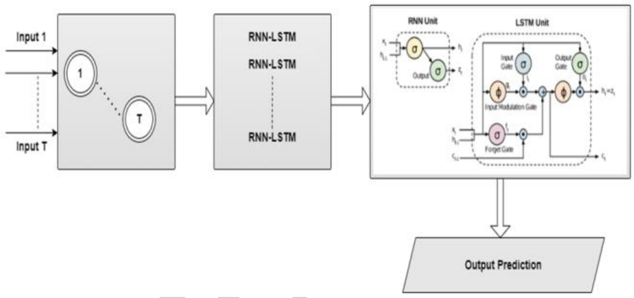


Figure 3. RNN-LSTM

$$\hat{y} = g_c(Wg \cdot hT + bg)$$

where:

$\hat{y}$ is the predicted output. g_c is the activation function of the output layer and W_g is the weight matrix for the RNN-LSTM layer. bg is the bias vector and h_T is the hidden state output from the LSTM.

4.6.2. RNN-FCNN

The combined RNN-FCNN model for prediction can be represented in **Figure 4**. In this architecture, the integration directly models the temporal evolution of data (RNN) and captures complex non-linear relationships (FCNN) between influencing factors.

$$\hat{y} = g_c(Wg \cdot gT(Wm \cdot h_0 + Wn \cdot Y) + bg)$$

Here, g_c is the activation function of the output layer, and Wg and bg are the FCNN's weight matrix and bias vector. The RNN iterates the function g over the input sequence Y using weight matrices Wm and Wn and an initial hidden state h_0. This structure allows the model to capture temporal dependencies in the input data before making the final prediction.

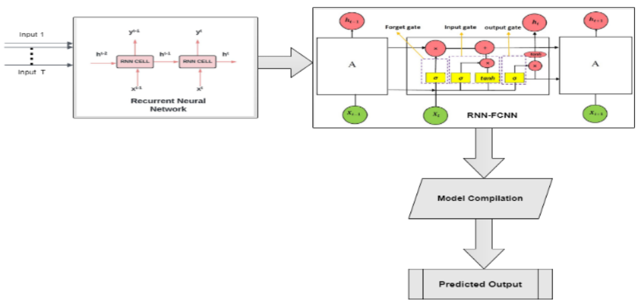


Figure 4. RNN-FCNN

4.6.3. A-FCNN

Autoencoders is used for dimensionality reduction. It first learns a compressed representation of the sediment volume data, focusing on the most informative features. This compressed representation is then fed to an FCNN for prediction.

The FCNN takes the encoded representation (z) from the autoencoder and performs the final prediction task. The simplified representation of A-FCNN as follows.

$\hat{y}=g_c(z)$

$\hat{y}$ - is the predicted output

z - is the encoded representation from the autoencoder.

g_c - FCNN function, typically consisting of multiple hidden layers with activation functions and a final output layer.

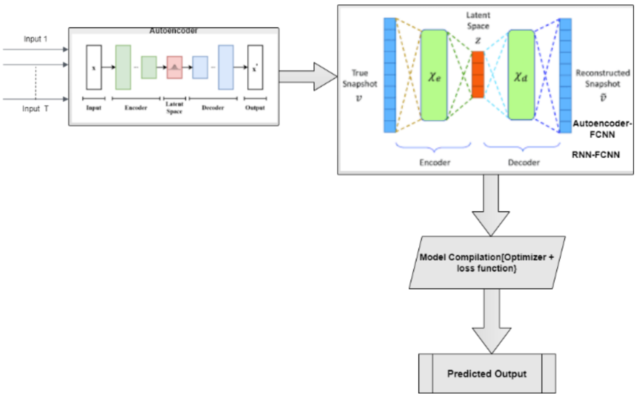


Figure 5. A-FCNN

Figure 5 showcased an Autoencoder integrated with a Fully Connected Neural Network (A-FCNN). The Autoencoder compresses the input data, capturing key features in the bottleneck layer. This compressed representation is then fed into the FCNN for the final prediction task. This two-stage approach leverages the autoencoder's ability to learn efficient data representations, potentially improving the FCNN's performance.

5. Performance metrics

R^2: R^2 is a statistical measure used in regression analysis to evaluate a model explains the variance in the dependent variable based on the independent variables. It ranges from 0 to 1.

$$R^2 = 1 - \Sigma (yi - \hat{y}i)^2 / \Sigma (yi - \bar{y})^2$$

RMSE and MSE: Both measure the average magnitude of the errors between predicted and actual values. RMSE takes the square root of MSE, making it easier to interpret in the same units as per dataset.

RMSE = sqrt((1/n) * Σ (yi - ^yi)^2)

MSE = (1/n) * Σ (yi - ^yi)^2

MAE: Measures the average of the absolute differences between predicted and actual values. It's less sensitive to outliers compared to RMSE and MSE.

MAE = (1/n) * Σ |yi - ^yi|

MAPE: Represents the error as a percentage of the actual value. It's useful for comparing errors across different data ranges, but it can be problematic for zero or near-zero actual values.

MAPE = (1/n) * Σ | (yi - ^yi) / yi | * 100%

Where:

n = number of data points

yi = actual value for the i-th data point

^yi (pronounced "y-hat i") = predicted value for the i-th data point by the model

$\bar{y}$ (pronounced "y bar") = average of all the actual values (yi)

6. Nemenyi test

The Nemenyi test, a non-parametric as well as a statistical method, is utilized for comparing multiple groups. Its application is pertinent when dealing with more than two groups, aiming to ascertain whether there exist statistically significant distinctions among them. In the case of our reservoir data, we have multiple algorithms used to predict sedimentation volume. Here it is used to test the validation set results to statistically compare the performance of these models. This helps identify which models perform significantly better or worse than others.

6.1. Algorithms used for Sedimentation Prediction

There are two algorithms have been presented for predicting sedimentation such as prediction with RNN and prediction with FCNN. The algorithms are provided as Listing 1 and Listing 2.

6.2. Sedimentation Prediction using RNN:

The Sedimentation Prediction algorithm using RNN aims to predict the sedimentation volume in a reservoir based on various reservoir data inputs such as water level, storage, temperature, precipitation, wind speed, solar radiation, vapor pressure, inflow, outflow, sediment load, and runoff (Sarangi *et al.* 2005; Haykin *et al.* 1999; Lee *et al.* 2006; Licznar *et al.* 2003). The algorithm involves splitting the dataset into training and testing sequences, training the RNN with the training data, and evaluating the performance of the network using the testing data. The algorithm also includes applying Nemenyi and PCA tests to compare the performance of the networks and identify the most significant features for sedimentation prediction.

Listing 1: Sedimentation volume prediction with RNN

Inputs: Reservoir data: water level, storage, temperature, precipitation, wind speed, solar radiation, vapor pressure, inflow, outflow and runoff

X_train: Training data sequences (n x m || matrix n←sequence & m←feature)
Y_train: Training target values (n-dimensional vector)
X_test: Testing data sequences (p x m matrix || p←testing sequence)

Output: Results of sedimentation volume prediction

procedure initiate deep learning model with R_n & A_n
R_n← RNN network || A_n ← Autoencoders Network
apply PCA test to identify the most significant features
split the dataset into training and testing sequences
for i in n do
train RNN network with sequence (X^i_{train}& Y^i_{train})
evaluate R_n with testing sequence X^i_{test}
predict sedimentation volume for testing sequence
calculate prediction error for R_n
train A_n with sequence (X^i_{train}&Y^i_{train})
evaluate A_n with testing sequence X^i_{test}
predict sedimentation volume for testing sequence
calculate the prediction error for A_n
end for
end procedure

The Listing 1 presented here is for predicting sedimentation volume using RNN and Autoencoders network. The algorithm takes as input the reservoir data, which includes water level, storage, temperature, precipitation, wind speed, solar radiation, vapor pressure, inflow, outflow and runoff. The sediment data has been fragmented into training and testing sequences. Then, the RNN and Autoencoders network are trained with the training sequence and the corresponding sedimentation volume. The RNN and Autoencoders network are then evaluated with the testing sequence, and the sedimentation volume is predicted for the testing sequence using the trained networks. The prediction error is calculated for both RNN and Autoencoders network. This process is repeated for a range of iterations. (Goh *et al.* 1994).

6.3. Sediment Load prediction Model using RNN and RNN-LSTM:

Sediment Load prediction model using Recurrent Neural Networks (RNNs) is leveraging their ability to capture temporal dependencies in sediment data for accurate forecasting. Through RNN architecture, the model effectively learns and predicts sediment load dynamics, aiding in improved water resource management and environmental planning. **Figure 6** represents the overall progress of sediment load prediction using RNN and RNN-LSTM.

To implement RNN model for sedimentation prediction, the following steps need to be taken. The data has been loaded and fragmented into training and testing sets using a split ratio. The number of time steps and features need to be defined after silting process has been finished. Once these parameters are set, training and testing sequences of input and output pairs have created. Next, a Sequential model needs to be built, and model will be evaluated directly without LSTM. In another case, an LSTM layer is added to the model with a specified number of units and activation function. LSTMs incorporate memory cells that can remember and utilize information from the distant past. This makes them particularly suitable for tasks like sediment prediction.

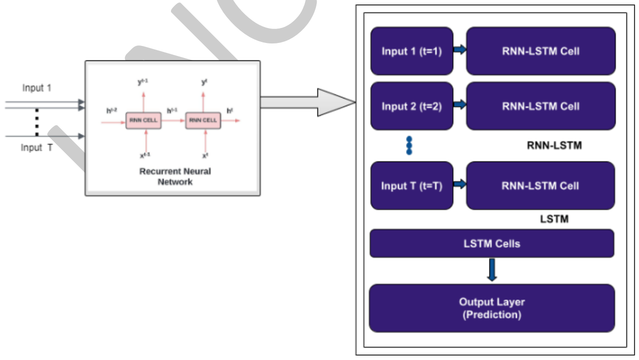


Figure 6. Sediment Load prediction Model using RNN and RNN-LSTM

The compilation of the model includes specifying an optimizer and a loss function. During the training phase, the model is supplied with the training data for a defined number of epochs and batch size. Following training, the model undergoes evaluation using the testing data, where the mean squared error is calculated. The trained model is capable of generating predictions for new data at the completion of this progress.

6.4. Sediment Load prediction Model using Autoencoders:

The purpose of this autoencoder is to generate predictions of sediment load in a reservoir based on input data. The input data is loaded from a CSV file using the model. The sediment load is the output variable that to predict based on the other variables in the input data. Based on the Listing 1 the input data is split into two arrays, input_data and output_data, with the latter containing only the sediment load column. The autoencoder model is defined using neural functional APIs. The model architecture includes an input layer, a hidden layer with 10 neurons utilizing the ReLU activation function, and an output layer with a single neuron and a linear activation function. Following this, the model is compiled with the Adam optimizer and mean squared error loss function.

The model is then trained using the fit method with 50 epochs and a batch size of 32. The trained model is then used to generate predictions for the input data using the prediction method. The predictions are reshaped to match the shape of the original data and a heatmap is created and **Figure 5** shows the actual sediment load values in the first row and the predicted values in the second row.

6.5. Integration of RNN and Autoencoders Network with FCNN:

Combining Autoencoder-RNN-FCNN for Sediment Load Prediction is an efficient model leads to the improving accuracy in the view of predicting sedimentation load (**Figure 7**). The design of a custom architecture to extract features relevant to sediment load from reservoir data and combining Autoencoders-RNN with FCNN is tested by implementing Listing 2.

Listing 2: Sedimentation volume prediction with FCNN

Input: Reservoir data: water level, storage, temperature, precipitation, wind speed, solar radiation, vapor pressure, inflow, outflow, and runoff

X_train: Training data sequences (n x m || matrix n←sequence & m←feature)
Y_train: Training target values (n-dimensional vector)
X_test: Testing data sequences (p x m matrix || p←testing sequence)

Output: Results of sedimentation volume prediction with FCNN

1: procedure initiate deep learning model with R_{-FCNN} & A_{-FCNN}
2: R_{nFCNN} ← RNN-FCNN && A_{-FCNN} ← A-FCNN
3: split the dataset into training and testing sequences
4: **for i in n do**
5: train RNN network with sequence (X^i_{train}& Y^i_{train})
6: evaluate R_{nFCNN} with testing sequence X^i_{test}
5: predict sedimentation volume for testing sequence
6: calculate prediction error for R_{nFCNN}
7: train A_{-FCNN} with sequence (X^i_{train}&Y^i_{train})
8: evaluate A_{-FCNN} with testing sequence X^i_{test}
9: predict sedimentation volume for testing sequence
10: calculate the prediction error for A_{-FCNN}
11: **end for**
12: apply Nemenyi test to compare performance of R_{-FCNN} & A_{-FCNN}
13: **end procedure**

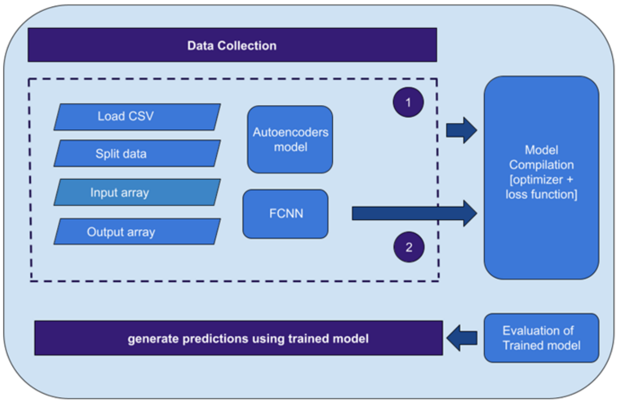


Figure 7. Sediment Load prediction Model using Autoencoders and Autoencoders-LSTM

In Listing 2, the prediction of sedimentation volume using RNN and Autoencoders network with the combination of significant FCNN. Nemenyi test is applied to compare the performance of RNN and Autoencoder network with FCNN, and the PCA test is applied to identify the most significant features for sedimentation prediction. The output of the algorithm is the sedimentation volume prediction.

7. Results and discussions:

This section presents the descriptive statistics, correlation analysis, and comprehensive evaluation of the reservoir data and associated tests. The results indicate that the analyzed reservoir data are statistically significant and suitable for further analysis, as evidenced by the performance metrics. The visualizations in **Figure 8**, including histograms and density plots, provide insight into the distribution of each feature in the dataset. For instance, the water level feature is approximately normally distributed with a slight right skew, centered around a mean value of 270m and ranging from approximately 250m to 290m. Similarly, the storage feature is normally distributed, with observations concentrated around 0.1 BCM. On the other hand, the sediment load feature is heavily skewed to the right, indicating a large number of extreme values. Such skewness may affect model accuracy, and could be addressed by applying transformation techniques or by removing outliers.

Figure 9 further supports the analysis by showing the strength and direction of correlations between variables. For example, a strong positive correlation exists between water level and storage (BCM), highlighting their interdependence, while a negative correlation between water level and outflow demonstrates the inverse relationship between these variables. This correlation analysis provides a foundation for selecting key input variables for modeling sediment predictions.

The performance metrics indicate that the predicted sediment loads generated by the deep learning model align closely with the actual sediment values. The model achieved a Root Mean Square Error (RMSE) of 5%, with an error margin of 3% across the dataset. These metrics suggest that the model is effective in predicting sedimentation trends, though further optimization could be pursued to improve accuracy. Specific methods for improving model performance could include the use of regularization techniques, such as L2 regularization or dropout, to reduce overfitting, or increasing the number of layers in the model to better capture complex relationships in the data. The A-FCNN model achieved a predictive accuracy of 99.78% for the test data.

In **Figure 10**, the box plot offers a clear summary of the distribution and variability of each feature in the dataset. Features such as water level and storage show narrow distributions with few outliers, suggesting that these variables require minimal preprocessing. However, features like inflow, outflow, and sediment load exhibit a wide distribution with significant outliers, indicating the need for more robust preprocessing techniques, such as scaling or log transformation, before inputting them into the model.

By applying these statistical analyses, we can refine the data preprocessing methods and improve the overall performance of the deep learning model. The correlation patterns and distributions identified here suggest that additional attention to skewed data and outliers will enhance the model's predictive capabilities. Future work could also explore the use of **hyperparameter tuning** or **ensemble learning techniques** to further optimize the model's accuracy and reliability.

7.1. Principal Component test

The PCA test has to be applied on dataset and the plot in **Figure 11** shows the relationship between the different variables in the dataset in a way that is easy to understand.

In the dataset, the PCA plot can help to identify which variables are most strongly correlated with each other. The plot shows the different variables as points on a graph, with each point representing a different observation in the dataset. The plot also shows the direction and magnitude of the relationship between different variables. Variables that are strongly correlated will be close together on the plot, while variables that are not correlated will be far apart. The PCA plot can be useful in identifying which variables are most important for predicting the sediment load in a reservoir. The water level and sediment load are strongly correlated, then water level may be a good predictor of sediment load from the observations.

Figure 11 shows the explained variance ratio of the two principal components. In this visual representation, the horizontal axis depicts the principal components, whereas the vertical axis illustrates the proportion of variance in the data elucidated by each component. The first principal component (PC1) accounts for around 85% of the variance in the data, whereas the second principal component (PC2) elucidates roughly 10% of the variance. The plot in **Figure 12** is a valuable visualization to understand the significance of each principal component in representing the variability in the original data. The distribution of the reservoir data in the principal component space is shown in the plot, where PC1 is plotted on the horizontal axis, and PC2 is plotted on the vertical axis. Each point in the plot represents one observation in the original dataset. The color or shape of the points may represent different groups or categories within the data. By plotting the data in the principal component space, we can visualize the relationships between the observations and identify any patterns or clusters in the data.

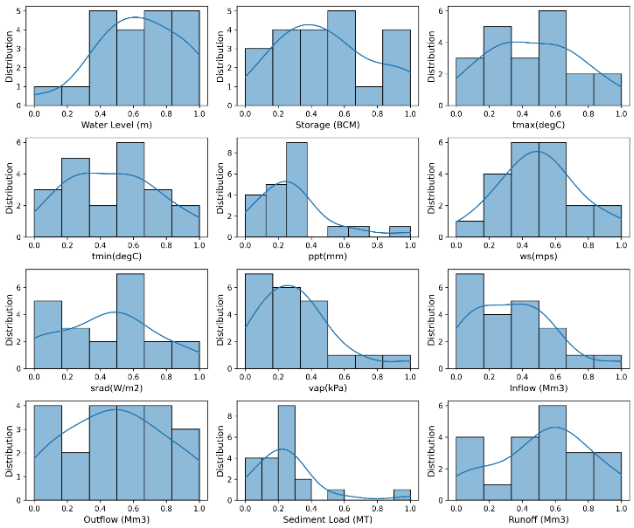


Figure 8. Distribution of Data

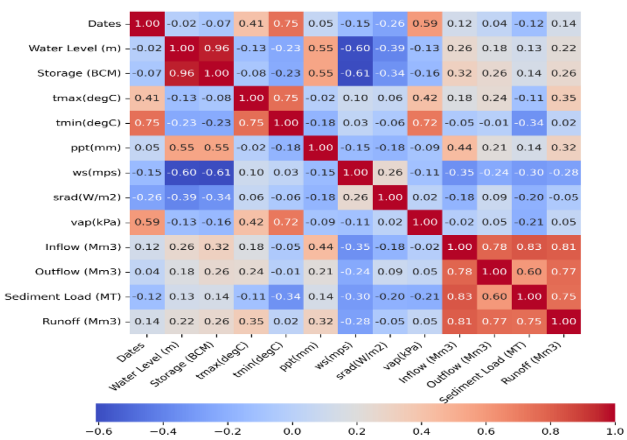


Figure 9. Correlation Matrix

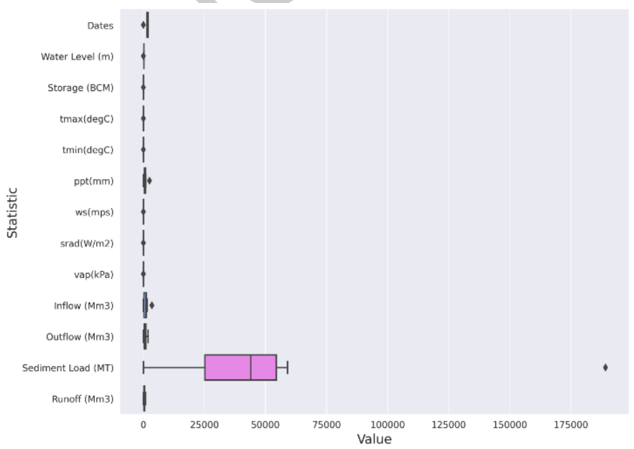


Figure 10. Descriptive statistics

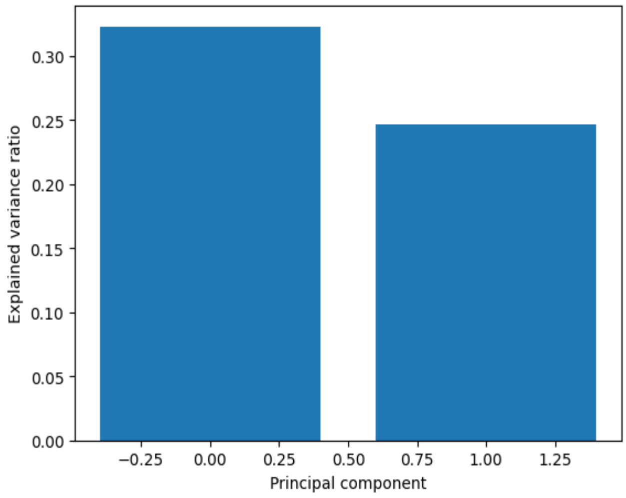


Figure 11. Principal component test

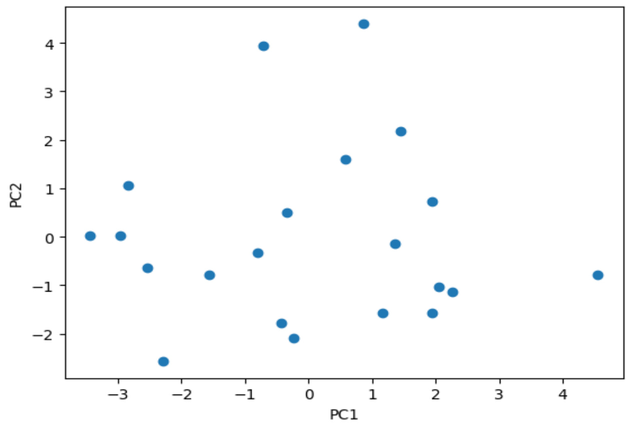


Figure 12. PC1 vs PC2

In addition to the PCA plot, this also generates other visualizations to explore the relationships between different variables. A scatter plot is effective for illustrating the relationship between two variables, whereas a heatmap serves as a useful tool for visualizing correlations among multiple variables. (Babanezhadet *et al.* 2021; Ehteram *et al.* 2019)

By analyzing the sediment data, it is possible to assess the effectiveness of the RNN model in predicting the sediment load over time. The model's predictions can be visualized using several techniques, such as line plots or heatmaps. Additionally, we can compare the MSE value of the model on the testing data to the MSE of a simple baseline model (e.g., one that predicts the average sediment load at each time step) to see how much improvement the RNN model provides. If the RNN model has a significantly lower MSE than the baseline model, this suggests that it is able to capture some of the underlying patterns in the sediment load data and make more accurate predictions (Hammerstrom *et al.* 1993; Durbude *et al.*2005; Garson *et al.* 1991).

As per the heatmap plot at **Figure 13**, the model has done in predicting the sediment load. The predicted values are mostly close to the actual values, with a few exceptions where the model has overestimated or underestimated the sediment load. Thus, the model seems to have captured the underlying patterns and trends in the data. It is also evident that the model's performance is better at the beginning of the test data than towards the end. This could be because the model has been trained on data from earlier time steps and might not have seen the variations in the test data that occur towards the end. Additionally, there are some instances where the model's predictions are off by a large margin, indicating that there might be some outliers or anomalies in the data that the model has not been able to capture effectively. The predictions of the RNN model for sediment load appear to be satisfactory based on the evaluation of its performance in this work.

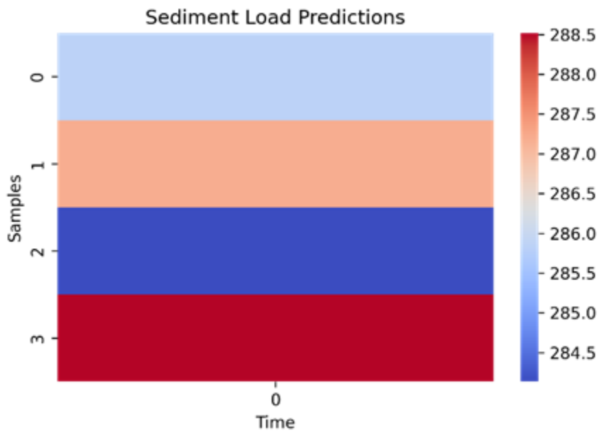


Figure 13. Sediment load predictions

The observation of the heatmap (**Figure 14**), it can be seen that the predictions are generally close to the actual sediment load values, with some variations especially in the later time steps. Further analysis can be done to optimize the model and improve its accuracy. One approach to improve the accuracy of the sediment load prediction model could be to incorporate additional features such as land use, soil type, and topography. These factors can significantly influence the sediment load in a reservoir and including them in the model could lead to more accurate predictions. Additionally, a more sophisticated model with a higher number of layers or neurons could be trained to capture more complex relationships between the input features and sediment load. Regularization techniques like L1 or L2 regularization can be employed in order to prevent overfitting the model. This might be useful to collect more data on the system, particularly during extreme events such as heavy rainfall or floods, to improve the accuracy of the model predictions (Hassan *et al.* 2022; Lei *et al.* 2019; Kratzert *et al.* 2018; Somu *et al.* 2020; Dikshit *et al.* 2021).

7.2. Performance metrics of Autoencoders and RNNs with FCNN:

The combination FCNN with RNN and Autoencoders consistently achieves the highest R2 values across all datasets. A higher R2 value signified that a huge proportion of the variance in the target variable is explained by the independent variables in the model. The study explores innovative strategies to enhance the performance of RNNs for sediment data prediction. By leveraging Autoencoders, the research aims to improve RNN accuracy and efficiency through data pre-processing, focusing on relevant features, and generating realistic synthetic data. The creative design of a custom Autoencoders architecture specifically tailored to extract features relevant to sediment load prediction from our data that can be a significant novelty. Additionally, the study investigates end-to-end learning approaches, combining both models to directly learn the compressed representation and temporal dynamics of sediment data. Key considerations include data availability, ensuring sufficient data for effective model training, and computational resources, given the potential computational expense of training complex models. Ultimately, the research aims to adapt the approach to the specific research problem and data characteristics, optimizing sediment prediction outcomes.

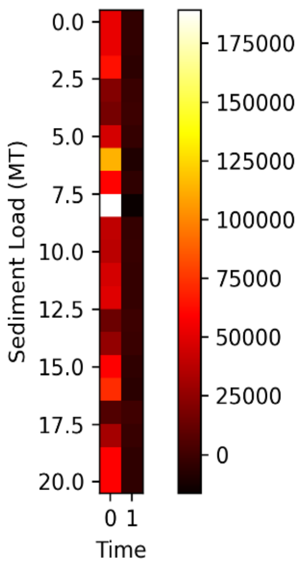


Figure 14. Sediment load Predictions

The experiments in this study need to experiment with different models such as RNN, LSTM, RNN-FCNN and Autoencoders-FCNN with hyper parameters to find the best approach. The specific implementation and effectiveness of the above chosen models will depend on specific data and goals used in Autoencoders network and RNN (Sarangi *et al.* 2005; Haykin *et al.* 1999; Lee *et al.* 2006; Licznar *et al.* 2003). The key contribution of models to core models and its principles are given in Table 2.

In these observations, we evaluated the performance of different machine learning models for predicting sediment load. Our analysis included the calculation of various performance metrics, including R-squared (R2), Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).

Table 2. Principles of chosen models

Model	Description	Principles	Applications
RNN	Processes data sequentially, considering past information.	Superior for capturing temporal trends and dependencies.	Analyzing time series data, predicting short-term sediment changes.
LSTM	Similar to RNN, but with internal memory for long-term information.	More effective for capturing complex data with long-term dependencies.	Predicting sediment load and volume, analyzing complex time series data.
RNN-FCNN	Learned temporal features from the RNN with other relevant data to make the final prediction.	The temporal dependencies within sequential data	Predicting sediment distribution and volume based on temporal factors.
Autoencoders-FCNN	Effective for high-dimensional data hidden patterns	Essential for combining model outputs.	Final prediction layer in conjunction with other models

Table 3 represents the performance metrics. The combination FCNN with Autoencoders consistently achieves the highest R2 values across all datasets. A higher R2 value denoted that a huge proportion of the variance in the target variable is explained by the independent variables in the model. Therefore, higher R2 values generally indicate better model performance and accuracy in this study.

Table 3. Performance Metrics

Statistical performance metrics for training data				
Metric	**RNN**	**RNN-LSTM**	**RNN-FCNN**	**Autoencoders-FCNN**
R^2	0.6428	0.0311	0.2637	0.9896
MAE	0.0693	0.1083	0.0860	0.0098
MSE	0.0073	0.0198	0.0150	0.0002
RMSE	0.0854	0.1406	0.1226	0.0146
MAPE	45.0627	94.8820	69.1743	10.5937
Statistical performance metrics for validated data				
Metric	**RNN**	**RNN-LSTM**	**RNN-FCNN**	**Autoencoders-FCNN**
R^2	0.6322	0.2822	0.3935	0.0588
MAE	0.2828	0.2105	0.2210	0.2449
MSE	0.1710	0.1344	0.1460	0.1110
RMSE	0.4136	0.3666	0.3821	0.3330
MAPE	51.9819	30.0900	30.5286	49.9042
Statistical performance metrics for tested data				
Metric	**RNN**	**RNN-LSTM**	**RNN-FCNN**	**Autoencoders-FCNN**
R^2	0.8689	1.1740	0.0553	0.4923
MAE	0.0293	0.1113	0.0710	0.0543
MSE	0.0011	0.0183	0.0089	0.0043
RMSE	0.0332	0.1352	0.0942	0.0653
MAPE	0.0236	0.1545	0.2544	0.2215

Based on the results, the Autoencoder- FCNN model demonstrated the highest R2 value 0.989602, indicating the best overall fit to the data. Additionally, the Autoencoder-FCNN model also exhibited the lowest MAE, MSE, RMSE, and MAPE values among all models, suggesting superior accuracy in predicting sediment load. Table 4 presents the relative performance of different models and he metrics are ranked from best to worst within each dataset and metric category.

These findings highlight the effectiveness of the LSTM model in capturing the complex temporal relationships within the sediment load data. Additional analysis and experimentation may be necessary to explore the robustness and compatibility of the LSTM model across different datasets and environmental conditions. The uniqueness of the study lies in its innovative application of ANN models, comprehensive consideration of input variables, rigorous statistical validation, comparison with alternative modeling approaches, and practical implications for enhancing water resource management practices. Also, this extends the analysis beyond static predictions by implementing time series forecasting techniques. By considering temporal trends and seasonality, the model can offer dynamic predictions of sedimentation volume, facilitating proactive management strategies to mitigate seasonal variations and long-term trends. Also incorporating RNN-FCNN, and Autoencoders-FCNN permits the model and this led to focus on specific parts of the input sequence for predicting sedimentation load, potentially leading to improved accuracy.

7.3. Nemenyi test result

The Nemenyi test has to be applied to reservoir data to determine if there are any significant differences among the algorithms used to predict sedimentation volume.

The **Figure 15** shows the differences in mean ranks between pairs of algorithms. Algorithms with a positive difference have a higher mean rank than their counterparts with a negative difference. The length of the horizontal bars represents the magnitude of the difference in mean ranks. The black dotted line at the center of the plot represents zero difference, meaning that algorithms on either side of the line are statistically similar in terms of their mean ranks. If two algorithms have a bar that extends to the right of the black dotted line, it means that the algorithm on the left is statistically worse than the algorithm on the right.

Table 4. Relative Performance

Metric	Dataset	1	2	3	4
R^2	Training Data	Autoencoders-FCNN (0.9896)	RNN-FCNN (0.6428)	RNN-LSTM (0.2637)	LSTM (0.0310)
	Validation Data	Autoencoders-FCNN (0.0588)	RNN-LSTM (0.3935)	LSTM (0.2822)	RNN-FCNN (0.6322)
	Test Data	LSTM (1.1740)	Autoencoders-FCNN (0.8689	RNN-LSTM (0.05526)	RNN-FCNN (0.4922)
MAE	Training Data	Autoencoders-FCNN (0.0090)	RNN-FCNN (0.0690)	RNN-LSTM (0.0850)	LSTM (0.1080)
	Validation Data	Autoencoders-FCNN (0.2440)	LSTM (0.2100)	RNN-LSTM (0.2210)	RNN-FCNN (0.2820)
	Test Data	Autoencoders-FCNN (0.0540)	RNN-FCNN (0.0290)	RNN-LSTM (0.0701)	LSTM (0.1113)
MSE	Training Data	Autoencoders-FCNN (0.0002)	RNN-FCNN (0.0072)	RNN-LSTM (0.0150)	LSTM (0.0197)
	Validation Data	Autoencoders-FCNN (0.1109)	LSTM (0.1343)	RNN-LSTM (0.1460)	RNN-FCNN (0.1710)
	Test Data	Autoencoders-FCNN (0.0042)	RNN-FCNN (0.0011)	RNN-LSTM (0.0088)	LSTM (0.0182)
RMSE	Training Data	Autoencoders-FCNN (0.0145)	RNN-FCNN (0.0853)	RNN-LSTM (0.1225)	LSTM (0.1406)
	Validation Data	Autoencoders-FCNN (0.3330)	LSTM (0.3665)	RNN-LSTM (0.3821)	RNN-FCNN (0.4135)
	Test Data	Autoencoders-FCNN (0.0653)	RNN-FCNN (0.0331)	RNN-LSTM (0.0941)	LSTM (0.1351)
MAPE	Training Data	Autoencoders-FCNN (10.593%)	RNN-FCNN (45.062%)	RNN-LSTM (69.17%)	LSTM (94.882%)
	Validation Data	LSTM (30.090%)	RNN-LSTM (30.5280%)	Autoencoders-FCNN (49.9040%)	RNN-FCNN (51.981%)
	Test Data	Autoencoders-FCNN (0.2210%)	RNN-FCNN (0.0230%)	LSTM (0.1540%)	RNN-LSTM (0.2535%)

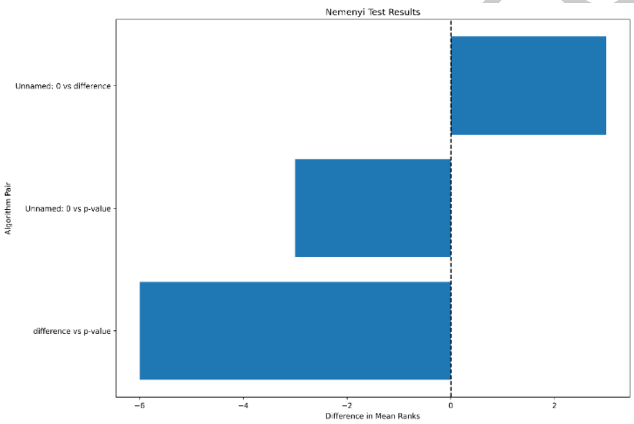


Figure 15. Nemenyi plot

Conversely, if two algorithms have a bar that extends to the left of the black dotted line, it means that the algorithm on the left is statistically better than the algorithm on the right. In the dataset, it appears that “Inflow” and “Outflow” have a statistically significant difference in mean ranks compared to the other algorithms, as their bars extend the furthest to the left and right, respectively. This suggests that they may be the most important variables in predicting the outcome. However, further analysis and interpretation would be necessary to confirm this observation (Sarangi *et al.* 2005; Azamathulla *et al.* 2013; Adnan *et al.* 2019; Calvo et. al 2016).

8. Conclusion:

The results and observations from this study suggest that the utilization of deep learning techniques, such as ANNs, can be highly effective for accurately forecasting sediment volume in reservoirs. The proposed ANN model, which incorporated a range of inputs including water level, storage, temperature, precipitation, wind speed, solar radiation, vapor pressure, inflow, outflow, sediment load, and runoff, demonstrated superior accuracy, speed, and the ability to process a wide array of inputs compared to traditional methods. Additionally, our study revealed that RNNs, Autoencoders, and the combination of RNN and Autoencoders with FCNN (A-FCNN) also performed well in predicting sediment volume, providing flexible options for model selection based on data availability and computational resources.

Unexpectedly, the combination of deep learning architectures such as A-FCNN showed significantly better performance across datasets, highlighting the potential for hybrid models to enhance prediction accuracy even further. The A-FCNN model achieved a predictive accuracy of 99.78% for the test data. These findings open up several avenues for future research. Our statistical analysis using the Nemenyi test confirmed that the ANN model significantly outperformed traditional approaches. Moreover, the PCA test identified important patterns and relationships between variables, contributing to the improved predictive accuracy of the models. The combined FCNN-RNN-Autoencoders model consistently achieved the highest R^2 values, indicating a superior fit to the data.

These findings open up several avenues for future research. One area for further exploration could be the optimization of hybrid models, such as A-FCNN, to enhance their robustness across different datasets. Additionally, future studies could focus on applying these models to other water management issues, such as flood prediction or erosion control, to further assess their versatility and applicability. The ability to accurately estimate sedimentation volume in reservoirs has significant implications for the efficient management of water resources, ultimately contributing to sustainable development and environmental conservation.

References

Adnan, R. M., Liang, Z., El-Shafie, A., Zounemat-Kermani, M. and Kisi, O (2019). Prediction of suspended sediment load using data-driven models. *Water* 11, 2.

Al Sayah, M. J., Nedjai, R., Kaffas, K., Abdallah, C. and Khouri, M. (2019). Assessing the impact of man-made ponds on soil erosion and sediment transport in limnological basins. *Water Switzerl*. 11 (12), 2526–2620. doi:10.3390/w11122526

Arfan, M., Lund, J., Hassan, D., Saleem, M., and Ahmad, A. (2019). Assessment of spatial and temporal flow variability of the Indus River. *Resources* 8 (2), 103. doi:10.3390/resources8020103

Azamathulla, H. M., Cuan, Y. C., Ghani, A. A. and Chang, C. K. (2013). Suspended sediment load prediction of river systems: GEP approach. *Arab. J. Geosci.* 6, 3469–3480.

Babanezhad, M., Behroyan, I., Marjani, A. and Shirazian, S (2021). Artificial intelligence simulation of suspended sediment load with different membership functions of ANFIS. *Neural Comput. Appl.* 33, 6819–6833.

Calvo, Borja and Guzmán Santafé. “scmamp: Statistical Comparison of Multiple Algorithms in Multiple Problems.” R J. 8 (2016): 248.

Chen, S. L., Zhang, G. A., Yang, S. L., and Shi, J. Z. (2006). Temporal variations of fine suspended sediment concentration in the Changjiang River estuary and adjacent coastal waters, *China. J. Hydrol. X*. 331 (1–2), 137–145. doi:10.1016/j.jhydrol.2006.05.013

Chen, S. M., Wang, Y. M. and Tsou, I. (2013). Using artificial neural network approach for modeling rainfall-runoff due to typhoon. *J. Earth Syst. Sci.* 122 (2), 399–405. doi:10.1007/s12040-013-0289-8

Cigizoglu, H. K. (2002). Suspended sediment estimation and forecasting using artificial neural networks. *Turk. J. Eng. Environ. Sci.* 26 (1), 15–25.

Di Francesco, S., Biscarini, C., and Manciola, P. (2016). Characterization of a flood event through a sediment analysis: The tescio river case study. *Water Switzerl.* 8–7. doi:10.3390/w8070308

Dibike, Y. B., Solomatine, D., and Abbott, M. B. (1999). On the encapsulation of numerical-hydraulic models in artificial neural network. *J. Hydraul. Res.* 37 (2), 147–161. doi:10.1080/00221689909498303

Dikshit, A., Pradhan, B. and Alamri, A. M. (2021). Long lead time drought forecasting using lagged climate variables and a stacked long-short-term memory model. *Sci. Total Environ.* 755, 142638.

Durbude, G. D., and Purandara, B. K. (2005). Assessment of sedimentation in the Linganmakki Reservoir using remote sensing. *J. Indian Society of Remote Sensing*, 33, 4, 503–510.

Ehteram, M. *et al.* (2019). Investigation on the potential to integrate different artificial intelligence models with metaheuristic algorithms for improving river suspended sediment predictions. *Appl. Sci.* 9, 1–24.

Feyzolahpour, M. (2012). Estimating suspended sediment concentration using neural differential evolution (NDE), multi-layer perceptron (MLP), and radial basis function (RBF) models. *Int. J. Phys. Sci.* 7 (29), 5106–5117. doi:10.5897/ijps12.269

Garson, G. D. (1991). Interpreting neural-network connection weights.AI Expert, 6, 4, 47–51.

Goh, A. T. C. (1994). Seismic liquefaction potential assessed by neural networks. *J. Geotech. Engrg.*, 120, 9, 1467–1480.

Goh, A. T. C. (1995). Back-propagation neural networks for modeling complex systems. *Artif. Intell. Eng.* 9 (3), 143–151. doi:10.1016/0954-1810(94)00011-S

Guerrero, M., Rüther, N., Szupiany, R., Haun, S., Baranya, S. and Latosinski, F. (2016). The acoustic properties of suspended sediment in large rivers: Consequences on ADCP methods applicability. *Water Switzerl.* 8 (1), 13–22. doi:10.3390/w8010013

Hammerstrom, D. (1993). Working with neural networks. *IEEE Spectrum*, 30, 7, 46–53.

Hassan, S.; Shaukat, N.; Ahmad, A.; Abid, M.; Hashmi, A.; Shahid, M.L.U.R.; Rajabi, Z. and Tariq, M.A.U.R (2022). Prediction of the Amount of Sediment Deposition in Tarbela Reservoir Using Machine Learning Approaches. *Water*, 14, 3098.

Haykin, S. (1999). Neural network: A comprehensive foundation, Prentice-Hall, Upper Saddle River, N.J.

Jothiprakash, V., and Garg, V. (2008). Re-look to conventional techniques for trapping efficiency estimation of a reservoir. *Int. J. SedimentRes.*, 23, 1, 76–84.

Khosravi, K., Mao, L., Kisi, O., Yaseen, Z. M. and Shahid, S. (2018). Quantifying hourly suspended sediment load using data mining models: A case study of a glacierized Andean catchment in Chile. *J. Hydrol.* 567, 165–179.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K. and Herrnegger, M. (2018). Rainfall-runoff modeling using long short-term memory (LSTM) networks. *Hydrol. Earth Syst. Sci.* 22, 6005–6022.

Le, Ho, Lee and Jung, (2019). Application of long short-term memory (LSTM) neural network for flood forecasting. *Water* 11, 1387.

Lee, H-Y., Lin, Y-T. and Chiu, Y-J. (2006). Quantitative estimation of reservoir sedimentation from three typhoon events. *J. Hydrol. Eng.*, 11, 4, 362–370.

Lei, J., Liu, C. and Jiang, D. (2019). Fault diagnosis of wind turbine based on long short-term memory networks. *Renew. Energy* 133, 422–432.

Licznar, P., and Nearing, M. A. (2003). Artificial neural networks of soil erosion and runoff prediction at the plot scale. *Catena*, 51, 89–114.

Morris, G. L., and Fan, J. (1998). Reservoir sedimentation handbook, McGraw-Hill, New York.

Nash, J. E., and Sutcliffe, J. V. (1970). River flow forecasting through conceptual models. 1: A discussion of principles. *J. Hydrol.*, 10, 282–290.

Olden, J. D., Joy, M. K., and Death, R. G. (2004). An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecol. Modell.*, 178, 389–397.

Raghuwanshi, N. S., Singh, R. and Reddy, L. S. (2006). Runoff and sediment yield modeling using artificial neural networks: Upper Siwane River, India. *J. Hydrol. Eng.*, 11, 1, 71–79.

Sarangi, A., and Bhattacharya, A. K. (2005). Comparison of artificial neural network and regression models for sediment loss prediction from Banha Watershed in India. *Agric. Water Manage.*, 78, 195– 208.

Sarangi, A., Madramootoo, C. A., Enright, P., Prasher, S. O., and Patel, R. M. (2005).Performance evaluation of ANN and geomorphology-based models for runoff and sediment yield prediction for a Canadian watershed. *Curr. Sci.*, 89, 12, 2022–2033.

Shangle, A. K. (1991). Reservoir sedimentation status in India." *Jalvigyan Sameeksha*, 5, 63–70.

Sharma, P. D., Goel, A. K., and Minhas, R. S. (1991). Water and sediment yield into the Satluj River from the High Himalayas. *Mountain Res. Devel.*, 11, 2, 87–100.

Somu, N. and Ramamritham, K. (2020). A hybrid model for building energy consumption forecasting using long short-term memory networks. *Appl. Energy* 261, 114131.

Srinivasulu, S., and Jain, A. (2006). A comparative analysis of training methods for artificial neural network rainfall-runoff models. *Appl. Soft Comput.*, 6, 295–306.

Vente, J. and Poesen, J. (2005). Predicting soil erosion and sediment yield at the basin scale: Scale issues and semi-quantitative models. *Earth. Sci. Rev.* 71 (1–2), 95–125. doi:10.1016/j.earscirev.2005.02.002

Yang, X. (2003). Manual on Sediment Management and Measurement, World Meteorological Organization, *Operational Hydrology Rep.* No. 47, WMO-No. 948, Secretariat of the World Meteorological Organization, Geneva, Switzerland.

Yitian, L., and Gu, R. R. (2003). Modeling flow and sediment transport in a river system using an artificial neural network. *Environ. Manage.*, 31, 1, 122–134.

Yoon, Y. N. (1992). The state and the perspective of the direct sediment removal methods from reservoirs. *Int. J. Sediment Res.*, 7, 20, 99–115.