Combined Approach of Statistical, Deep learning and finer modeling methods for water quality forecasting

Water quality assessment forecasting, identifying quality parameter trends using statistical and deep learning modeling including hybrid approaches with autoencoders and other methods of fine tuning of the models.

Dr. Md. Jashim Uddin, Md Rakib Hasan, Fazla Zawadul Arabi, Arafat Rahman

Department of Soil, Water and Environment
University of Dhaka

Submitted to,
Prof Dr Md. Jashim Uddin
Department of SWE, University of Dhaka
On 28 January, 2024

Contents

1. *Introduction*

Maintaining clean and healthy water resources is crucial for the survival and well-being of all living organisms. However, water quality is constantly under threat from various anthropogenic and natural factors, leading to pollution and environmental degradation. To effectively manage water resources and prevent pollution outbreaks, accurate and timely prediction of water quality parameters is essential. Deep learning is emerging as a powerful tool for water quality forecasting, offering solutions to the limitations of traditional methods. This advanced technology can model the complex, nonlinear relationships between water parameters and influencing factors, capturing the dynamic nature of water bodies. Deep learning models like long-short term memory(LSTM), convolutional neural network(CNN) , recurrent neural network (RNN), gated recurrent unit (GRU) and hybrid models excel at learning long-term dependencies within time series data, allowing them to predict future water quality parameters with improved accuracy. Additionally, deep learning models can handle vast amounts of data from diverse sources, including sensor networks and historical records, leading to more robust and generalizable forecasts. By uncovering hidden patterns within the data using autoencoders, deep learning approaches can further refine their predictions and overcome challenges with data noise and limited availability. With its ability to adapt to dynamic environments and learn from extensive datasets, deep learning is poised to revolutionize water quality forecasting, the model can be feeded data using statistical approach of moving average, ultimately paving the way for better water resource management and pollution prevention strategies.

2. *Challenges in Water Quality Forecasting*

Predicting water quality is a complex task due to several factors:

- *Nonlinear relationships:* The interdependencies between various water quality parameters and influencing factors are often nonlinear and complex, making traditional statistical models less effective.
- *Dynamic nature of water bodies:* Water quality can be significantly impacted by seasonal changes, weather events, and human activities, requiring models to capture these dynamics.
- *Data heterogeneity:* Water quality data often originates from diverse sources with varying levels of accuracy and completeness, posing challenges for model training and generalization.

Proposed Approach:

This research proposal presents a novel hybrid approach for water quality forecasting that combines the strengths of statistical and deep learning techniques with an autoencoder.

- *Statistical models:* Initial data analysis and feature engineering will be conducted using statistical methods to identify significant correlations and trends in the data. This will provide insights for building robust models.
- *Deep learning:* Deep learning models will be employed to capture the complex temporal and non-linear relationships between water quality parameters and influencing factors.

3.1 Research Objectives

The primary objectives of this research are:

- Develop a hybrid forecasting model using statistical and deep learning techniques with an autoencoder.
- Evaluate the performance of the proposed model on real-world water quality data from a specific water body.
- Compare the accuracy and robustness of the proposed model with traditional statistical and standalone deep learning models.
- Identify key factors influencing water quality and provide insights for informed decision-making in water resource management.

Expected Outcomes:

- A novel and effective method for water quality forecasting with improved accuracy and generalizability.
- Insights into the complex dynamics of water quality and the key factors influencing it.
- Valuable tools and data-driven approaches for water resource management and pollution prevention.

3.2 Sampling and Water Quality Parameters

Data collection will be preliminary divided into four (4)/ three (3) sampling points putting into consideration the point of discharge close distance from industries. The distance from one

sample point to another will be 20/30/40 m apart. The sampling will be focused upstream to downstream of the river. The sampling will be done 2 days in a week. The water quality parameters will be considered in this study are:

i) Dissolved Oxygen (DO) ii) Temperature

iii) pH iv) Electrical Conductivity (EC)

v) Total Dissolved Solids (TDS) vi) Biological Oxygen Demand (BOD)

Water Bodies in concern:

i) Buriganga ii) Dhaleshwari

iii) Turag iv) Balu

v) Shitalakkhya vi) Hatirjheel

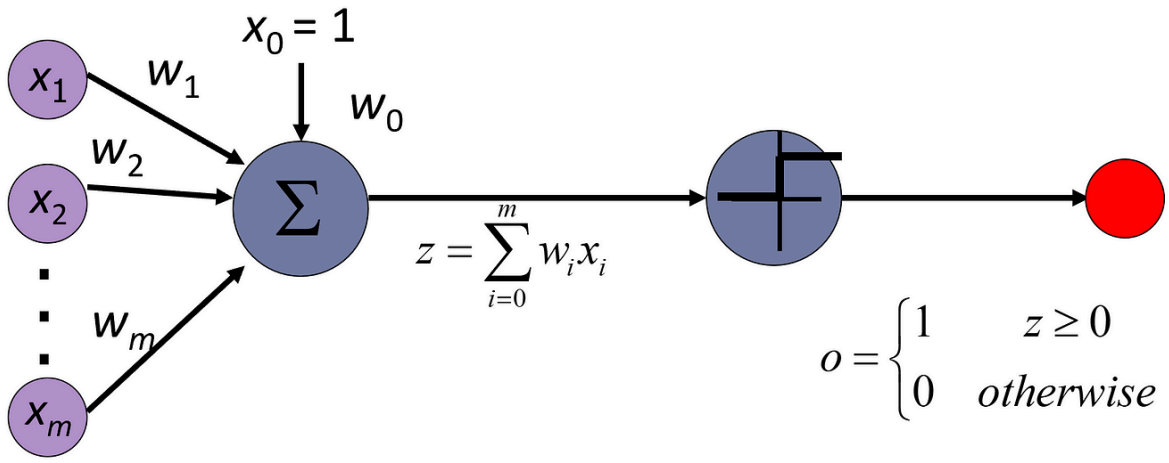
4. Statistical and Deep learning models

4.1. Perceptron

The perceptron, introduced by Frank Rosenblatt in 1958, is a foundational concept in machine learning. It represents a single-layer artificial neural network capable of performing binary classification tasks. Here's a breakdown of its core theoretical aspects:

Architecture:

- *Inputs:* The perceptron takes a set of real-valued features (numerical measurements) as inputs. These features represent characteristics of the data points being classified.
- *Weights:* Each input feature is associated with a weight, indicating its importance in influencing the classification decision.
- *Bias:* A constant term, the bias, adds flexibility to the decision boundary and accounts for overall offsets in the data.
- *Activation Function:* Finally, a thresholding function, typically the step function, transforms the weighted sum of inputs and bias into a binary output. If the sum exceeds the threshold, the output is 1 (positive class), otherwise, it's 0 (negative class).



Learning Algorithm:

The perceptron employs a supervised learning algorithm called the perceptron learning rule. This iterative process adjusts the weights based on the difference between the predicted and actual class of a data point:

- *Prediction error:* If the perceptron misclassifies a data point, the error is calculated as the difference between the desired class and the predicted class.
- *Weight update:* Each weight is adjusted proportionally to the prediction error and the corresponding input feature, pushing the decision boundary towards the correct classification.

Capabilities and Limitations:

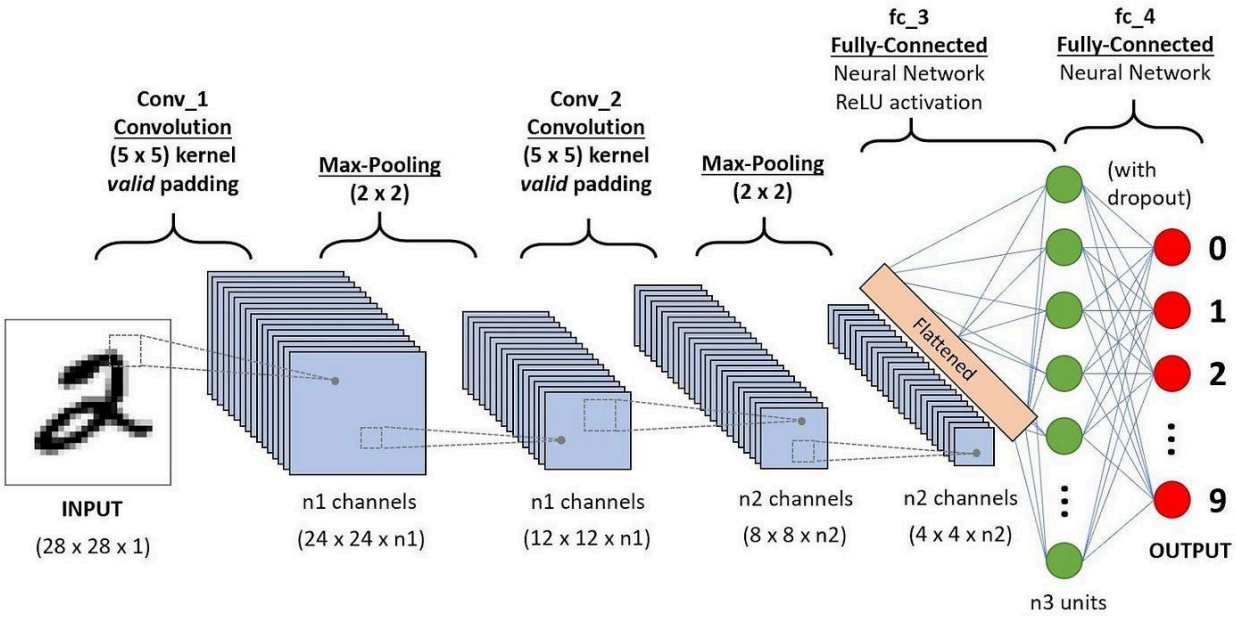
- *Linear Separation:* Perceptrons can only efficiently handle linearly separable data, where a straight line can perfectly divide the data points into two distinct classes.
- *Limited Complexity:* As single-layer models, perceptrons struggle with complex non-linear relationships between features and classes.
- *Convergence Concerns:* The learning rule may not always converge, potentially getting stuck in an oscillation loop without reaching an optimal solution.

Significance:

Despite its limitations, the perceptron laid the groundwork for more advanced neural network architectures. It introduced key concepts like weighted summation, activation functions, and learning algorithms, paving the way for the development of powerful deep learning models.

4.2 Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) are a powerful class of deep learning models particularly adept at processing image data. While convolutional neural networks (CNNs) are traditionally associated with image processing and computer vision tasks, their ability to extract spatial and temporal features has made them increasingly relevant in diverse fields, including forecasting. Here's an exploration of CNN model theory within the context of forecasting:



Capturing Spatial and Temporal Dependencies:

- Multidimensional data: Forecasting tasks often involve time series data where each time step corresponds to a measurement of the target variable (e.g., temperature, sales volume). CNNs can effectively handle this multidimensional data by leveraging 1D convolutions along the time axis and 2D convolutions for processing additional spatial information if available (e.g., grid-based weather data).

1D and 2D convolutions for forecasting

- Local relationships: Similar to image processing, CNNs in forecasting extract features by performing convolutions over local windows of the time series data. This allows them to capture short-term dependencies and identify patterns within specific timeframes.
- Long-term trends: Stacking multiple convolutional layers with increasing kernel sizes enables the model to capture progressively longer-term trends and relationships within the data.

Advantages of CNNs for Forecasting:

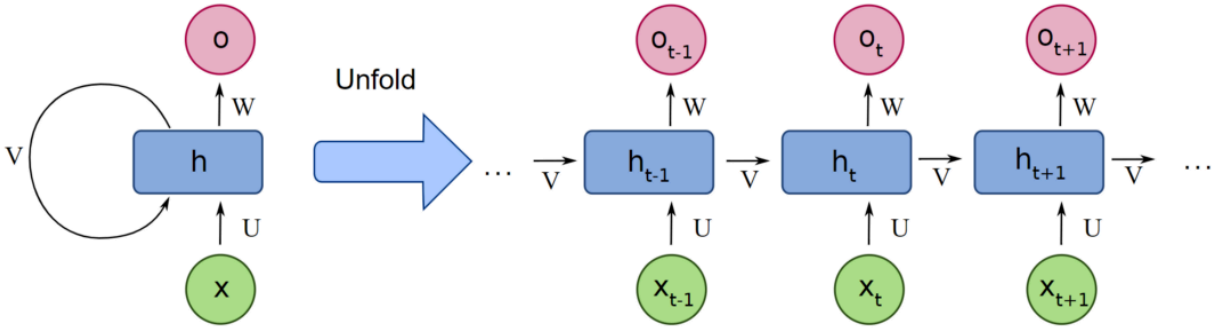
Automatic feature extraction: Unlike traditional statistical models that require manual feature engineering, CNNs automatically learn relevant features from the raw data using the convolution operation.

- Handling complex patterns: CNNs can adeptly handle non-linear and complex patterns within the data, often outperforming traditional models in capturing nuances and hidden relationships.
- Scalability: With efficient parallelization on GPUs, CNNs can handle large datasets effectively, making them suitable for real-world forecasting applications with vast amounts of data.
- Activation functions: Non-linear functions like ReLU are applied to the output of the convolution layer to introduce non-linearities and enhance feature representation.
- Pooling layers: These layers downsample the feature maps by summarizing information from a local neighborhood (e.g., max pooling takes the maximum value within a window). This reduces the dimensionality of the data and helps control overfitting.
- Multiple layers and channels: CNNs typically have multiple convolutional layers stacked together, with each layer extracting increasingly complex features from the previous layer. Additionally, each layer often has multiple channels, allowing the network to learn different types of features simultaneously.
- Fully-connected layers: In the final stages of a CNN, fully-connected layers are used for tasks like classification or regression. These layers take the flattened feature maps from the convolutional layers and map them to the desired output.

4.3 Recurrent Neural Network (RNN)

Recurrent neural networks (RNNs) have emerged as powerful tools for forecasting tasks, particularly when dealing with sequential data like time series. Their ability to capture temporal

dependencies within data sequences makes them well-suited for predicting future values based on past information.



Memory Mechanism:

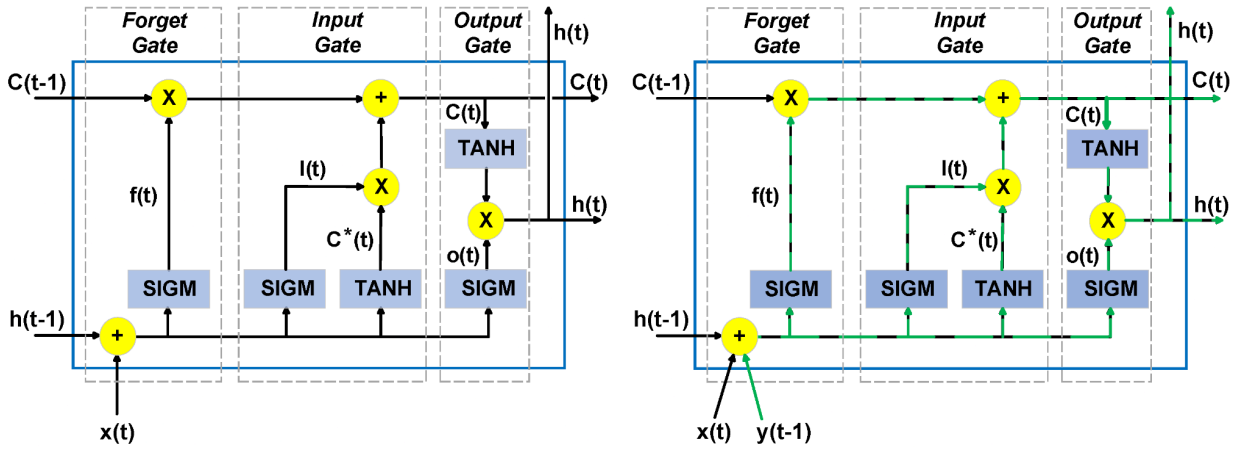
Unlike traditional feedforward neural networks, RNNs possess a unique memory mechanism that allows them to retain information from previous inputs. This is achieved through recurrent connections within the network that feed back processed information to itself at each time step.

Advantages of RNNs for Forecasting:

- Long-term dependency modeling: RNNs excel at capturing long-term temporal relationships within data, crucial for accurate forecasting of trends and patterns.
- Dynamic modeling: RNNs can adapt to changing patterns and incorporate new information over time, making them suitable for real-time forecasting applications.
- Handling complex sequences: RNNs can handle non-linear and complex relationships within data sequences, providing more accurate predictions than simpler models.

4.4 *Long Short Term Memory (LSTM)*

LSTMs are a special kind of recurrent neural network (RNN) designed specifically to address the limitations of traditional RNNs in handling long-term dependencies within sequential data.They excel at capturing long-range relationships and patterns within sequences, making them highly effective for tasks involving sequential data, such as natural language processing, time series prediction, and other applications where understanding context over extended time periods is crucial.



Key Components:

1. Cell State: The LSTM's "memory" unit, carrying information across time steps, selectively preserving or forgetting relevant details.
2. Forget Gate: Determines which information from the previous cell state to discard, preventing irrelevant details from cluttering memory.
3. Input Gate: Decides which new information to incorporate into the cell state, updating memory with current inputs.
4. Output Gate: Regulates the output of the cell, controlling how much of the cell state is exposed to the next time step and influencing predictions.

Advantages in Forecasting:

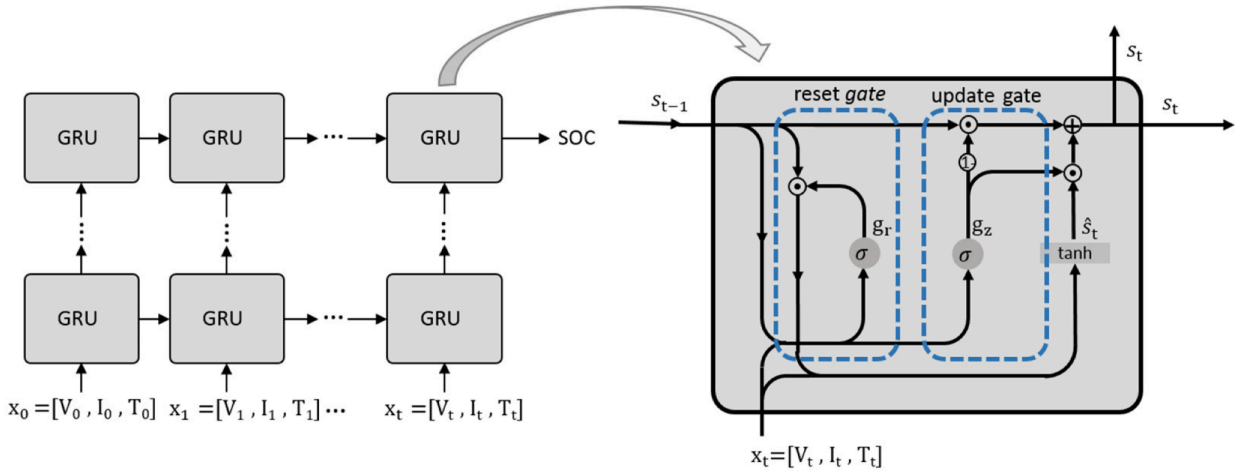
- Long-Term Dependency Handling: LSTMs excel at modeling long-range relationships within data, essential for tasks like predicting trends, seasonal patterns, and cyclical behaviors.
- Handling Varying Time Scales: LSTMs can adapt to different time scales within a sequence, effectively capturing both short-term fluctuations and long-term trends.
- Non-Linear Relationship Modeling: LSTMs can model complex, non-linear relationships within time series data, often outperforming linear models in accuracy.

4.5 Gated Recurrent Unit (GRU)

Gated Recurrent Units (GRUs) are another powerful tool in the forecasting toolbox, offering a compelling alternative to LSTMs. While they share the ability to handle long-term dependencies in sequential data, GRUs bring a distinct flavor with their simpler architecture and computational

efficiency. Unlike LSTMs with separate forget, input, and output gates, GRUs utilize a single "update gate" and a "reset gate" to control information flow. This streamlined approach:

- Reduces the number of parameters: Making GRUs faster to train and computationally cheaper to run.
- Maintains effectiveness: Despite the simpler design, GRUs can still capture long-term dependencies and achieve comparable accuracy to LSTMs in many forecasting tasks.



Key Components:

1. Hidden State: Similar to the LSTM cell state, the GRU's hidden state stores information across time steps.
2. Update Gate: Decides how much of the previous hidden state to retain and how much new information to incorporate.
3. Reset Gate: Determines how much to "reset" the hidden state, effectively forgetting irrelevant past information.

Forecasting with GRUs:

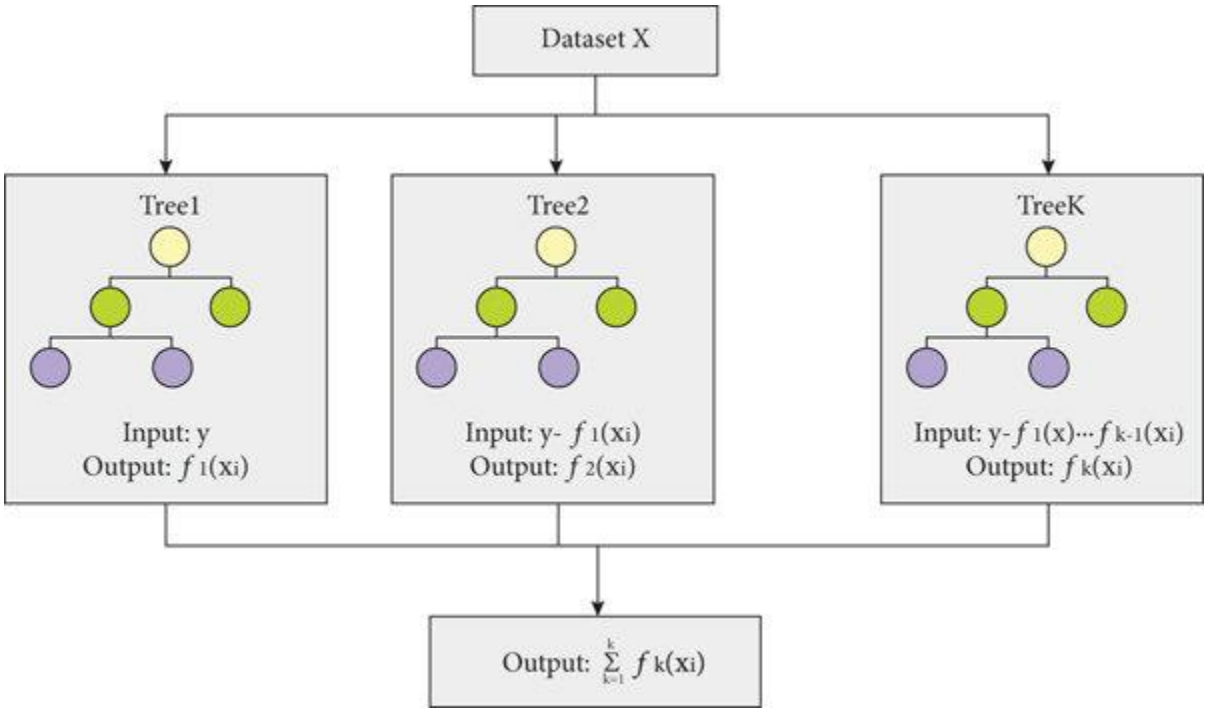
1. Sequential Processing: GRUs process time series data one step at a time, updating the hidden state based on the current input and previous hidden state.
2. Learning Temporal Relationships: The update and reset gates learn how past information influences future outcomes, capturing long-term dependencies within the data.
3. Prediction: Based on the updated hidden state, GRUs generate predictions for future time steps.

Advantages in Forecasting:

- Computational Efficiency: GRUs require fewer parameters and simpler computations compared to LSTMs, making them faster to train and run, especially on resource-constrained devices.
- Comparable Accuracy: Despite their simpler architecture, GRUs can often achieve similar or even better forecasting accuracy than LSTMs in certain tasks.
- Interpretability: GRUs have a more straightforward structure than LSTMs, making their internal workings and learned features easier to understand.

4.6 XGBoost

XGBoost, short for Extreme Gradient Boosting, has emerged as a leading tool for forecasting tasks thanks to its ability to handle complex relationships within data and produce highly accurate predictions.



Implementation for Forecasting:

- Regression Setting: XGBoost is typically used in a regression setting for forecasting tasks, aiming to predict the continuous values of the target variable at future time steps.

- Hyperparameter Tuning: Choosing the optimal configuration of parameters like learning rate, number of trees, and regularization strength is crucial for maximizing model performance.
- Validation and Evaluation: Robust validation techniques like cross-validation are essential to assess the model's generalizability and avoid overfitting.

Gradient Boosting:

- XGBoost builds on the concept of gradient boosting, an ensemble learning method that combines multiple weak learners (decision trees) to create a strong learner.
- Each tree focuses on correcting the errors made by the previous trees, progressively refining the prediction as more trees are added.
- This iterative process helps capture complex non-linear relationships and hidden patterns within the data.

Advantages for Forecasting:

- Accuracy and Efficiency: XGBoost often outperforms traditional forecasting models due to its ability to handle non-linearity and complex interactions between variables.
- Regularization: It utilizes techniques like L1 and L2 regularization to prevent overfitting, ensuring the model generalizes well to unseen data.
- Scalability: XGBoost can handle large datasets efficiently due to its parallelization capabilities and optimized memory usage.

4.7 ***Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN)***

CEEMDAN emerges as a powerful tool for enhancing time series forecasting by decomposing complex signals into simpler, interpretable components.:

Decomposition Power:

- Empirical Mode Decomposition (EMD): CEEMDAN builds upon EMD, which iteratively decomposes a signal into Intrinsic Mode Functions (IMFs) representing specific frequency components.
- Adaptive Noise Handling: Unlike traditional EMD, CEEMDAN tackles the issue of residual noise through ensembled sifting processes, leading to cleaner and more robust IMFs.

Implementation for Forecasting:

- Decomposition Step: Apply CEEMDAN to decompose the time series data into IMFs and a residue.
- Individual Model Application: Employ suitable forecasting models (e.g., RNN, LSTM) on each IMF and the residue separately.
- Reconstruction and Prediction: Reconstruct the predicted IMFs and residue to obtain the final forecast for the original signal.

Advantages for Forecasting:

- Identifying Hidden Patterns: Decomposing complex data into IMFs unveils hidden periodicities, trends, and non-linear relationships, enhancing understanding of the underlying dynamics.
- Improved Feature Extraction: Each IMF captures specific frequency components, providing targeted features for various forecasting models, often outperforming traditional feature engineering methods.
- Reduced Model Complexity: Forecasting individual IMFs can be simpler than directly modeling the original signal, potentially reducing computational cost and improving generalization.

4.8 Moving Average

Moving averages are a simple yet powerful tool for forecasting future values in a time series. Their core principle lies in smoothing out data fluctuations and revealing the underlying trend, making them particularly useful for short-term forecasts.

Calculating a Moving Average:

- Choose a window size (n), representing the number of data points to consider.
- For each time step t, calculate the average of the n data points surrounding it:

Moving Average (t) = 1/n * Σ (Data point i), where i ranges from t-n/2 to t+n/2

Types of Moving Averages:

- Simple Moving Average (SMA): The most basic type, averaging all data points within the window.

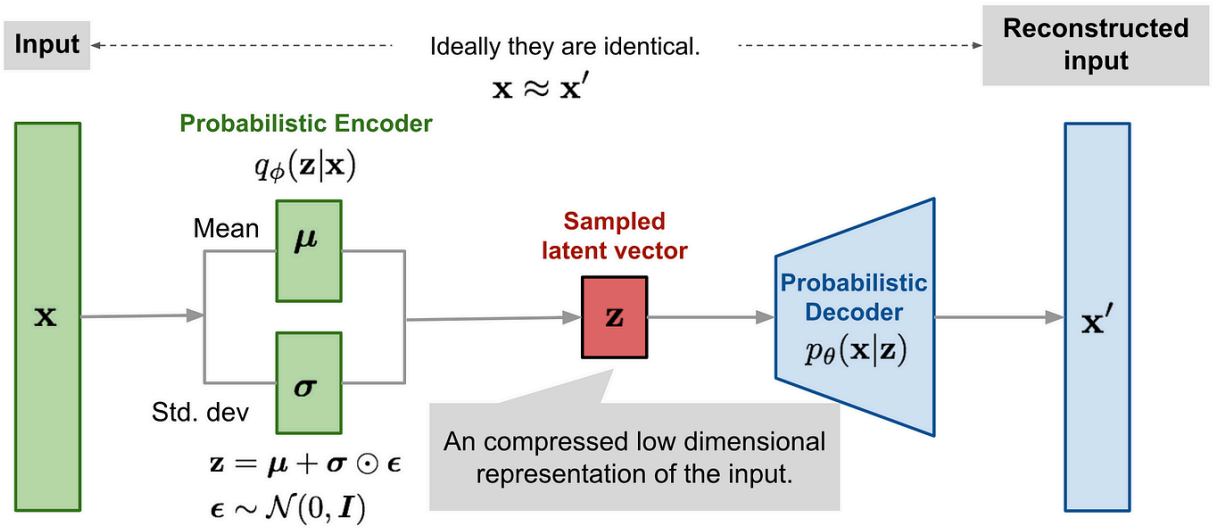
- Exponential Moving Average (EMA): Assigns higher weights to recent data points, giving them more influence on the average.
- Weighted Moving Average (WMA): Assigns custom weights to each data point within the window, allowing for flexible tailoring to specific trends.

Advantages for Forecasting:

- Simple and Easy to Implement: Moving averages are straightforward to calculate, requiring minimal computational resources.
- Effective Noise Reduction: They smooth out short-term fluctuations and random noise, revealing the underlying trend and seasonality.
- Suitable for Short-Term Predictions: Moving averages excel at predicting the next few values in a series, but their accuracy diminishes for longer-term forecasts.

4.9 Autoencoders

While autoencoders are traditionally known for their prowess in tasks like dimensionality reduction and anomaly detection, their ability to capture latent representations of data has sparked growing interest in their application to forecasting. Here's a deep dive into the theoretical underpinnings of autoencoders in forecasting:



Encoder-Decoder Architecture: Autoencoders consist of two neural networks:

    - Encoder: Compresses input data into a lower-dimensional latent representation, capturing essential features and patterns.
    - Decoder: Reconstructs the original input from the latent representation, ensuring the model has learned meaningful patterns.

Framework:

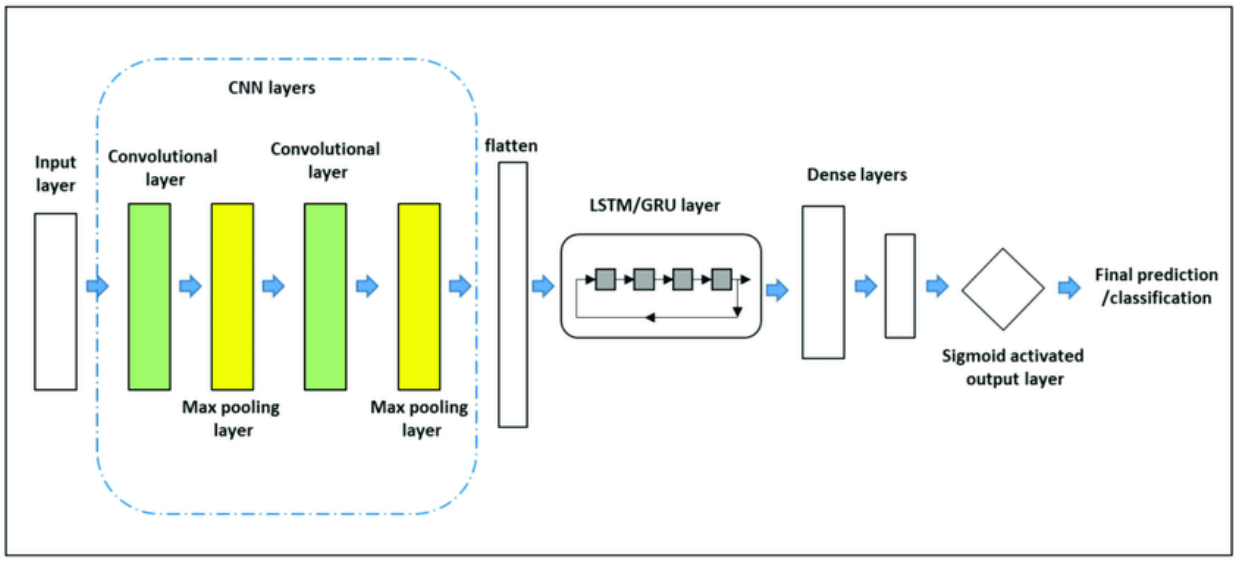
    - Train an autoencoder on historical time series data to learn latent representations.
    - Use the encoder to create latent representations for future time steps (e.g., using lagged values or external factors).
    - Feed these representations into a separate forecasting model (e.g., LSTM, statistical models) for prediction.

Advantages for Forecasting:

- Non-Linear Feature Learning: Autoencoders can capture complex, non-linear relationships within time series data, often outperforming traditional linear models.
- Noise Reduction: The compression-reconstruction process can filter out noise, enhancing the signal-to-noise ratio and improving forecast accuracy.
- Latent Representation Discovery: The latent space often reveals hidden patterns and underlying dynamics of the time series, providing insights into its behavior.
- Anomaly Detection: Deviations between original and reconstructed data can signal anomalies, useful for identifying unusual events or patterns.

4.10 Hybrid models

The realm of time series forecasting has witnessed a surge in hybrid models combining the strengths of diverse architectures like convolutional neural networks (CNNs), recurrent neural networks (RNNs), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRUs). This approach harnesses the unique capabilities of each architecture to create models that outperform individual techniques.



Individual Strengths:

- CNNs: Excel at extracting spatial features from images and grid-based data. In time series forecasting, they can capture local relationships and patterns within specific time windows.
- RNNs: Possess an internal memory mechanism that allows them to learn long-term dependencies within sequential data, making them suitable for capturing temporal relationships.
- LSTMs and GRUs: Specialized RNNs designed to overcome the vanishing/exploding gradient problem that limits traditional RNNs, leading to improved performance in capturing long-term dependencies.

Hybrid Model Strategies:

- Early Fusion: Combine raw time series data as input, leveraging CNNs for feature extraction and RNNs/LSTMs/GRUs for capturing temporal dependencies.
- Late Fusion: Extract features using individual models (e.g., CNNs for local patterns, RNNs for trends) and combine them before final prediction.
- Hierarchical Fusion: Build multiple layers with different architectures, with lower layers focusing on specific aspects (e.g., CNNs for seasonality) and higher layers integrating information for final prediction.

Advantages of Hybrid Models:

- Improved Feature Learning: Combining CNNs with RNNs/LSTMs/GRUs can lead to richer feature extraction, encompassing both spatial and temporal information.
- Enhanced Long-Term Dependency Modeling: Hybrid models can effectively capture both short-term fluctuations and long-term trends, providing more accurate forecasts.
- Flexibility and Adaptability: Different architectures can be combined and optimized to suit specific forecasting tasks and data characteristics.

5. Benefits of the Proposed Approach:

This hybrid approach offers several advantages over conventional methods:

- Improved accuracy: Combining statistical analysis with deep learning can lead to more accurate and reliable forecasts, especially for complex and non-linear relationships.
- Dynamic modeling: The proposed approach can capture the dynamic nature of water bodies by incorporating temporal dependencies and external factors into the model.
- Data efficiency: Autoencoders can help extract valuable features and reduce noise in the data, allowing the models to learn effectively even with limited data availability.

6. Equipment and Instrumentation list

i) DO meter ii) pH meter

iii) EC meter iv) TDS meter

v) Thermometer

7. Costing

Medium	Cost (Approx)
Transportation	
Equipment Cost	

8. Conclusion:

Accurate water quality forecasting is critical for ensuring the sustainability and health of our water resources. This research proposal presents a promising approach that combines the strengths of statistical and deep learning techniques to address this challenge. By developing and evaluating a hybrid model, this research aims to contribute significantly to the field of water quality monitoring and management.

Key References of literature review:

i) Wei-Bo Chen & Wen-Cheng Liu. *Artificial neural network modeling of dissolved oxygen*

*in reservoir (*https://link.springer.com/article/10.1007/s10661-013-3450-6)

ii) J. I. Ubah, L. C. Orakwe, K. N. *Ogbu. Forecasting water quality*

parameters using artificial neural network for irrigation purposes.
(https://www.researchgate.net/publication/357303697_Forecasting_water_quality_parameters_using_artificial_neural_network_for_irrigation_purposes)

iii) Jian Sha, Xue Li, Man Zhang. *Comparison of Forecasting Models for Real-Time Monitoring of*

Water Quality Parameters Based on Hybrid Deep Learning Neural Networks
(https://www.mdpi.com/2073-4441/13/11/1547)