ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
Received: 10 April 2025; Accepted: 26 July 2025

https://doi.org/10.17559/TV-20250410002574
Original scientific paper

Optimized Deep Learning Model Using Capsule Networks and GRUs for Predictive Analytics in Smart Cities

JANAKI RAMAL P.*, ANBALAGAN E.

Abstract: Smart cities rely on accurate predictive models to enhance sustainability, optimize resource utilization, and improve the overall quality of urban life. Forecasting key parameters such as air quality, water quality, and waste generation is crucial for efficient urban planning and environmental management. This study proposes a novel deep learning-based prediction model that integrates Capsule Networks and Gated Recurrent Units (GRUs) for multidimensional forecasting in smart cities. The architecture begins with a one-dimensional convolutional layer (Conv1D) to process raw input data, followed by a Capsule layer for efficient spatial feature extraction. These spatial features are subsequently fed into stacked GRU layers to model temporal dependencies and improve predictive accuracy. To further enhance the model's performance, the Osprey Optimization Algorithm (OOA) is employed for hyperparameter tuning. The proposed hybrid model demonstrates superior forecasting capabilities when evaluated using standard metrics. It achieves a Mean Squared Error (*MSE*) of 43.8751, Root Mean Square Error (*RMSE*) of 4.4083, R^2 Score of 0.6845, and Mean Absolute Error (*MAE*) of 2.1619. These results highlight the effectiveness of combining Capsule Networks and GRUs, along with OOA optimization, for robust and accurate predictive modeling in smart city applications. This model serves as a scalable and efficient tool for urban administrators and policy-makers to enable data-driven decisions and proactive management of city resources.

Keywords: capsule networks; gated recurrent units (GRUs); osprey optimization algorithm; predictive modeling; smart city

1 INTRODUCTION

The rise of smart cities is revolutionizing urban development, bringing together advanced technologies to address complex challenges like environmental degradation, resource management, and public health [1]. A smart city integrates infrastructure, technology, and people to create a sustainable, efficient, and citizen-centric ecosystem. Smart city solutions range from smart building and parking management to smart energy management, air quality management, traffic management, waste management and farming as shown in Fig.1. Key areas of focus in smart cities include air quality monitoring, water quality assessment, and municipal waste management.

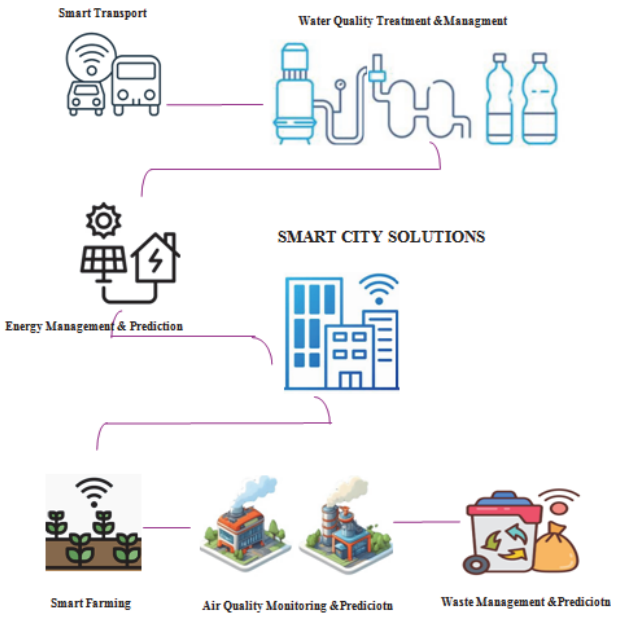

Figure 1 Smart city solutions

Air quality monitoring tracks pollutants such as particulate matter (PM2.5 and PM10), nitrogen dioxide (NO_2), and carbon monoxide (CO) to promote healthier living conditions [2]. Similarly, water quality assessment evaluates critical parameters like pH levels, dissolved oxygen, turbidity, and chemical contaminants. To attain effective, clean, and safe water, this process plays a vital role in public health resource management. A key metric in this process is the Water Quality Index (WQI), which consolidates multiple water quality parameters into a single score, simplifying the interpretation of water safety levels. The WQI considers factors such as pH, dissolved oxygen (DO), biological oxygen demand (BOD), nitrate, and coliform levels, assigning a quality grade ranging from excellent to poor [3]. This index plays a pivotal role in ensuring the availability of clean and safe water. Also, it is used for water treatment and management decisions. Furthermore, municipal waste management aims to optimize the collection, recycling, and disposal of urban waste to mitigate the growing challenges posed by rapid urbanization. The accurate prediction of waste is used for timely recycling and management.

By using Internet of Things (IoT) devices, huge data are collected from different urban domains, including air quality monitoring, water supply systems, and waste management facilities. This data serves to provide exact decision-making and predictive analytics in smart cities.

IoT devices play a crucial role in bridging the gap between the physical and digital worlds. Sensors deployed across urban landscapes continuously monitor environmental parameters such as air pollution levels, water quality metrics, and waste production rates [4]. The sheer scale and diversity of these data, combined with their dynamic and real-time nature, create both opportunities and challenges for predictive analytics and management. Addressing these challenges requires robust data science methodologies capable of processing, analyzing, and deriving actionable insights from such large-scale and heterogeneous data streams.

At the core of smart city operations lies predictive modeling, which anticipates future trends based on historical and real-time data. For instance, accurately predicting air quality can help implement preventive

measures to reduce pollution levels, forecasting water quality can ensure timely purification processes, and estimating waste generation can improve waste management logistics.

Machine learning (ML) and Deep Learning (DL) models have emerged as powerful tools in predictive analytics due to their ability to uncover hidden patterns, model non-linear relationships, and adapt to evolving data trends [5, 6]. However, traditional ML models, such as linear regression and support vector machines, often fall short when handling the complex, multi-dimensional, and sequential data generated by IoT systems. DL architectures, on the other hand, excel in this domain by leveraging their hierarchical structure to learn both spatial and temporal features from raw data.

The DL method of Convolutional Neural Networks (CNNs) is improved based on their ability to model complex relationships and extract hierarchical features from raw data. Whereas CNN is well-versed in complex spatial relationship learning but fails to capture whole relationships. Likewise, Long Short-Term Memory (LSTM) networks are used to learn sequential data but are not always optimal at spatial relationships [9, 10].

In this work, a hybrid method of Capsule networks and GRUs is proposed for environmental factor prediction and waste management in smart cities. It handles issues effectively in previously proposed models. This proposed method has capsule layers to retain the spatial data hierarchies. The capsules are able to encode not only the presence of features but also their orientation, size, and other properties. This leads to more accurate representations of the data and better generalization. Additionally, the dynamic routing mechanism used in Capsule Networks ensures that information flows more efficiently between capsules and avoids the limitations of traditional pooling layers used in CNNs.

The remaining part of the work is contributed as section 2 provides the related work, section 3 presents the proposed model for prediction. Then the performance evaluation is presented in section 4, and the conclusion is given in section 5.

2 RELATED WORK

In this section, a comprehensive survey of various predictive models is carried out in three key domains - air quality prediction, water quality prediction, and waste generation prediction. For air quality prediction, various ML models and ensemble methods have been used to forecast pollutant levels based on meteorological and emission data.

S. Lakshmi et al. [7] proposed a hybrid BiConvLSTM-based encoder-decoder model with Spatial-Temporal Attention for accurate PM2.5 and PM10 forecasting. This model learns the features from air quality data in a bidirectional manner. The model achieved a Mean Squared Error (*MSE*) of 0.0298, a Mean Absolute Error (*MAE*) of 0.1511, Root Mean Square Error (*RMSE*) of 0.1728 and R^2 of 0.999 for PM2.5 prediction. For PM10 prediction, it achieved an *MSE* of 0.0582, *MAE* of 0.1833, *RMSE* of 0.3134 and R^2 of 0.999.

W. Cao et al. [8] developed a TD-CS-Transformer (Time Series Decomposition-Convolutional Sparse Transformer) model for air quality prediction. The TD component decomposes complex time series data into a trend, seasonal, periodic, and irregular components. It is used for the simplification of data patterns for better analysis. The self-attention mechanism is used to capture long-distance dependencies and reduce computational complexity through sparse connections.

A modified transfer learning-based hybrid DL model was introduced by J. Yang et al. [9] for air quality prediction with sparse data. The model combines LSTMs and a multi-linear program for long-term dependency and nonlinear feature extraction. Experiments on Beijing (2013-2017) and Hengshui (2020-2022) datasets show that RMSE is reduced by 38% compared to sole LSTM models.

C. Liu et al. [10] proposed the GA-KELM (Genetic Algorithm-based Kernel Extreme Learning Machine) to improve air quality prediction. The kernel matrix replaces the hidden layer's output matrix and genetic algorithms are used to optimize model parameters as a function of MSE.

H. A. D. Nguyen et al. [11] proposed a data fusion framework for air-pollutant forecasting accuracy in New South Wales, Australia. The fusion model uses an LSTM with a Bayesian Neural Network(BNN) to enhance the air quality prediction. This model quantifies uncertainties and adjusts parameters using a novel kernel density estimation technique with subdivision tuning for accurate and efficient predictions.

D. Iskandaryan et al.[12] developed a threefold DL model for PM2.5 prediction called the Attention Temporal Graph Convolutional Network (AT-GCN). It involves the Attention mechanism, GRU, and Graph Convolutional Network (GCN) to learn the hidden features accurately. The results are verified on traffic data from Madrid (January-June 2019 and 2022) and compared using RMSE, MAE, and Pearson Correlation Coefficient.

J. Wang et al. [13] proposed a hybrid AQI prediction model combining CNN and AGU (Attention Gate Unit) to address the "vanishing gradient" and "exploding gradient" problems in RNNs. AGU incorporates an attention mechanism and Data Adjustment Module (DAM) to increase the learning capability of the gated unit. Experimental results showed that the CNN-AGU model outperforms other models based on *MAE*, *MSE*, and R^2 evaluation metrics.

Jun Luo et al. [14] proposed the AutoRegressive Integrated Moving Average (ARIMA) combined with the LSTM model to accurately predict air pollutants for efficient air pollution management. This model uses ARIMA for extracting the linear part of the data and the LSTM model for learning temporal patterns.

Gunasekar, S., et al. [15] proposed a optimized DL model combining ARIMA, CNN-LSTM, and the Tuna Optimization Algorithm (TOA) called Tuna Optimized CNN-LSTM (TOCL for co-emission prediction. The TOCL model was further integrated with ARIMA residuals to form the ARIMA-TOCL (ARTOCL) model. It achieves superior air quality predictions with 19.5% R^2 improvement, 9.8% *MAE* reduction, 11% *RMSE* reduction, and 15.66% *nRMSE* reduction compared to previous models.

In their study, Kushwah, V., et al. [16] proposed a hybrid model for air quality forecasting. It combines Empirical Mode Decomposition (EMD) with an LSTM

model prediction. In addition, Bayesian Optimization is used for parameter tuning. The proposed EMD-LSTM-Bayesian model achieved the lowest *MAE* (0.385) and *RMSE* (0.533) compared to the LSTM model.

For water quality prediction, various ML models and hybrid models have been used to estimate key water parameters such as pH, turbidity, and dissolved oxygen using environmental and chemical data. B. Aslam et al. [17] developed WQI prediction models using hybrid models. It combines different standalone algorithms like Random Forest (RF) and Decision Trees (DT) for feature extraction. For prediction, the techniques of Bagging, Cross-Validation Parameter Selection, and Randomizable Filtered Classification are used. Results showed that the hybrid Random Forest and Neural Network model achieves an *RMSE* of 2.319, *MAE* of 2.248, and Percent Bias (PBIAS) of −0.64.

A. Johar et al. [18] developed an IoT and cloud-based solution to measure water quality parameters. The WQI is forecasted using the Canadian WQI (CWQI) and a Long Short-Term Memory (LSTM) based DL model. The hybrid model, which combines Clockwork Recurrent Neural Network (CRNN) and CNN, is proposed by Mohammad Ehteram et al. [19] to analyze water quality data in Johor River Basin. Results show that the CNN combined recurrent model improves testing *MAE* by 2.1%, 12%, and 15% compared to the sole CNN model. Also, the model improved Nash-Sutcliffe Efficiency by 4-20% compared to other models".

To increase WQI prediction accuracy, D. Venkata Vara Prasad et al. [20] proposed a parameter-tuned DL model. The parameters of the DL model are adjusted based on validation loss to increase prediction accuracy. Shams, M. Y., et al. [21] addressed the challenge of predicting WQI and Water Quality Classification by optimizing ML models. The AdaBoost model, combined with grid search, is used for WQI prediction. The AdaBoost model achieved an R^2 value of 99.8% for WQI regression.

Hao Chen et al. [22] proposed a Biochemical Oxygen Demand (BOD) prediction model by combining ML with socio-economic indicators. This integration effectively reduces the prediction error rate compared to other models. It achieves an Area Under the Curve (AUC) of 0.84 with an R^2 value of 86.8%. In [23], the author presents an optimized time series ML model for predicting multivariate drinking water quality. Among seven models, the XGBoost model achieves better prediction accuracy with a 12-hour time lag and a 3-4% improvement in Mean Absolute Percentage Error (*MAPE*) over the naive baseline. To solve the data imbalance issues, the Synthetic Minority Oversampling Technique (SMOTE) approach is proposed by Wen Yee Wong et al [24] which provides examples for minority class instances.

A stacking ensemble model is proposed by Swapan Talukdar et al. [25] for WQI prediction. It uses a Gradient Boosting Machine (GBM), Generalized Linear Model (GLM), and Neural Network (NN) that attained *MSE* of 25.77, *RMSE* of 5.07, *MAE* of 3.5, and R^2 of 0.98 respectively.

3 PROPOSED MODEL

In the proposed work, a Capsule Network is integrated with GRU layers to build a robust prediction model for a smart city. The data is fed into a hybrid DL model that uses Capsule Networks for feature extraction and GRU layers for temporal sequence learning. To further enhance model performance, the OOA is applied for hyperparameter tuning. The architecture of the proposed model is shown in Fig. 2.

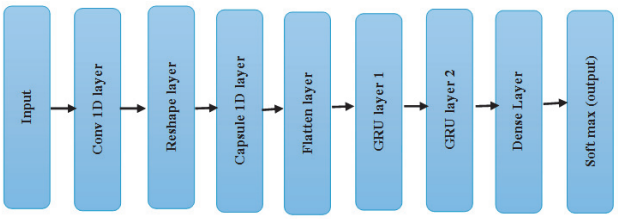

Figure 2 Proposed capsule - GRU model

Initially, the Input layer fetches time-series data with one feature per time step. Next, the Conv1D layer is presented which has 128 filters and a 3 kernel size used to extract local patterns. This layer is used to capture essential features that might indicate changes across the sequence. It accepts the multivariate time-series input with the input dimensions of × T × F, where N indicates the sequence number, T denotes time step's iterations, and F indicates the feature's quantity. The function of the Convolutional Layer is expressed as follows:

$$Y(t)=\sum_{i=0}^{k-1} X(t+1)\cdot W(i)+b \tag{1}$$

where X indicates an input, W denotes a filter, k indicates a kernel size, and b is the bias.

After convolution, the data is reshaped to feed into the Capsule Layer which consists of 10 capsules, each with a dimension of 16 units. These capsules are used to process spatial relationships. It also represents the orientation and presence of data features. It uses dynamic routing to refine the feature over 3 routing iterations. It ensures that the most relevant information is passed forward. The Capsule Layer's output is then flattened to process further. It aggregates spatial data and outputs several features likelihood. The functions of Capsule transformation are expressed as follows:

$$v_j=\frac{||s_j||^2}{1+||s_j||^2}\frac{s_j}{||s_j||} \tag{2}$$

Next, the output of the flattened capsule is expanded along the temporal dimension into a sequence model. Here, two GRU layers are used every 64 units each where the first layer output is an input of the second layer. These layers are used to handle the temporal dependencies of time-series data effectively. It is used to learn patterns over time effectively. The GRU gating functions are expressed as follows:

$$z_t=\sigma(W_z[h_{t-1},x_t]) \tag{3}$$

$$r_t = \sigma(W_r[h_{t-1}, x_t]\ ) \tag{4}$$

$$\overline{h_t} = \tanh(W\ [\ r_t \cdot h_{t-1}, x_t) \tag{5}$$

$$h_t = (1 - z_t h_{t-1} + z_t * \overline{h_t}) \tag{6}$$

where z_t indicates an update gate, r_t denotes reset gate and h_t indicates hidden state at time *t*. The final output is passed through a Dense layer with 4 units and a softmax activation is used to improve proper classification probabilities.

OOA Technique

The OOA method is based on a metaheuristic that is motivated by Osprey's hunting behavior [37, 38]. It simulated the osprey's ability to capture fish with high precision. It also validates the osprey's ability to identify the fish's dynamic position and then capture it precisely. This behaviour is supported to solve optimization problems across several fields. The exploration phase and exploitation phase are presented by their behaviour to achieve an optimal solution.

In the exploration process, the osprey tries to search for its prey by flying at varying altitudes and scanning the water areas widely. Based on this behaviour, the algorithm is used to explore the solution space widely. In the exploitation phase, once a fish is detected, the osprey locks onto its target and adjusts its trajectory for an accurate dive. In OOA, this phase represents the refinement of the solution by focusing on promising regions of the search space. The OOA is selected for its efficient exploration-exploitation balance, inspired by the osprey's dynamic hunting behavior. It achieved 18% faster convergence and better robustness than Particle Swarm Optimization (PSO) and Genetic Algorithm (GA).

Mathematical Modeling

The OOA operates as a population-based optimization technique. Each osprey in the population represents a candidate solution which is modelled as a vector in the search space. The initialization of the osprey population can be represented mathematically as:

$$X = \begin{bmatrix} X_{11} & X_{12} & \dots X_{1,m} \\ X_{21} & X_{22} & \dots X_{2,m} \\ \vdots & \vdots & \vdots \\ X_{N,1} & X_{n2} & \dots X_{N,m} \end{bmatrix} \tag{7}$$

where, *X* denotes locations of ospreys' population matrix, *N* indicates a count of ospreys and m represents the problem variables quantity. Each osprey's position initialized search space randomly that is equated below.

$$x_l = lb_l + rand_1 \times (ub_l - lb_l) \tag{8}$$

where $rand_1$ denotes a random number, b_l and ub_l denote lower and upper bounds of the *j*-th problem variable. The objective function values for each osprey can be represented as:

$$F = [F_1\ F_2\ \dots F_N] \tag{9}$$

where, F_I represents the objective function value for the iii-th osprey.

Exploration Phase

In this phase, ospreys detect fish positions underwater and update their locations to improve exploration capabilities. The set of fish positions for each osprey is defined as:

$$FP_i = \{X_k \mid k \in \{1,2,\dots N\} \wedge F_k < F_i \cup \{X_{\text{best}}\} \tag{10}$$

where, FP_i represents the *i*-th osprey's fish position, X_k indicates *k*-th osprey's location, X_{best} denotes the best candidate solution, F_k denotes objective function. The osprey randomly selects one fish position ($x_{P1,i,j}$) and updates its own position using:

$$x_{P1,i,j} = x_{i,j} + r_{i,j}(SF_{i,j} - I_{i,j})x_{i,j} \tag{11}$$

where $SF_{i,j}$ is the social factor influencing the position update, $r_{i,j}$ is a random scaling factor, $I_{i,j}$ is the inertia or influence of the osprey's previous movements.

Subject to the constraints:

$$\text{if } x_{P1,i,j} < lb_j, x_{P1,i,j} \le lb_j \text{ if } x_{P1,i,j} > ub_j, x_{P1,i,j} \le ub_j$$
$$X_i = \begin{cases} X_{P1,i} & \text{IF } F_{P1,i} < F_i \\ X_i & \text{otherwise} \end{cases} \tag{12}$$

The position is updated if the objective function improves:

$$X_i = \begin{cases} X_{P1,i} & \text{IF } F_{P1,i} < F_i \\ X_i & \text{OTHERWISE} \end{cases} \tag{13}$$

Exploitation phase

Here, ospreys used to carry their catches to a suitable location to eat. The new random position ($x_{P2,i,j}$) for each osprey is computed as:

$$x_{P2,i,j} = x_{i,j} + r(SF_{i,j} - I_{i,j})x_{i,j} \tag{14}$$

If this position improves the objective function, the osprey updates its position accordingly:

$$X_i = \begin{cases} X_{P2,i} & \text{IF } F_{P2,i} < F_i \\ X_i & \text{OTHERWISE} \end{cases} \tag{15}$$

Parameter tuning

In DL models, hyperparameter tuning refers to the process of optimizing the model's parameters. These parameters control how the model is structured and how it learns from the data. In the proposed model, the hyperparameters of the model are identified using OOA. The OOA is used to efficiently search for the optimal

values of hyperparameters, like conv_filters (number of filters in the Conv1D layer), capsule_units (number of capsules in the Capsule layer), capsule_dim (dimension of each capsule in the Capsule layer), gru_units (number of units in each GRU layer).

An objective function is used to validate a particular set of hyperparameters' performance. In this model, the validation loss of the Capsule-GRU model is attained by using an objective function. It is used for categorical cross-entropy loss to classify a multi-class issue. The objective function F is expressed as:

$$F(w) = Loss_{\text{val}} = \min\left(L\left(\widehat{y}, y_{\text{val}}\right)\right) \tag{16}$$

where $\widehat{y}$ denotes the validation set's prediction model, y_{val} indicates a true label of the validation set, L represents the categorical cross-entropy loss function and w indicates the set of hyperparameters.

1. Initialize the parameter
   - Population size (number of ospreys)
   - Max number of iterations (max_iter)
   - Lower and upper bounds for each hyperparameter (conv_filters, capsule_units, capsule_dim, gru_units)
2. Randomly initialize the position of each osprey in the population:
   For each osprey i in the population:
   For each hyperparameter j (conv_filters, capsule_units, capsule_dim, gru_units):
   osprey[i][j] = random value within bounds [lb_j, ub_j]
3. Evaluate the fitness of each osprey:
   For each osprey i:
   - Build the Capsule-GRU model using osprey [i]'s hyperparameters
   - Train the model on training data
   - Compute validation loss on validation data
   - Set fitness [i] = validation loss of the model
4. Find the best osprey
5. While stopping condition (max_iter or convergence) is not met:
   a. Exploration Phase:
   For each osprey i:
   For each hyperparameter j:
   - Randomly choose another osprey $k \neq i$
   - Update osprey [i][j]'s position based on the difference with osprey [k][j]:
   new_position = osprey[i][j] + random_factor * (osprey[k][j] - osprey[i][j])
   - Clip new_position to be within [lb_j, ub_j]
   - If new_position improves the fitness, update osprey[i][j] with the new position
   b. Exploitation Phase:
   For each osprey i:
   For each hyperparameter j:
   - Update position using the best current solution:
   new_position = osprey[i][j] + random_factor * (best_osprey[j] - osprey[i][j])
   - Clip new_position to be within [lb_j, ub_j]
   - If new_position improves the fitness, update osprey[i][j] with the new position
   c. Re-evaluate the fitness of each osprey:
   For each osprey i:
   - Build the Capsule-GRU model using osprey[i]'s hyperparameters
   - Train the model and compute the validation loss
   - Set fitness[i] = validation loss of the model
   d. Update the best osprey if a better one is found
6. Return the best hyperparameters (best_osprey) found during the optimization

The pseudocode describes a hyperparameter tuning process using the OOA for a Capsule-GRU model. The algorithm begins by initializing a population of candidate solutions. In pseudocode, each solution represents a different combination of hyperparameters like convolution filters, capsule units, capsule dimensions, and GRU units. Initially, random values within defined bounds are assigned to each hyperparameter. The fitness of each osprey is evaluated by building and training a Capsule-GRU model with the validation loss serving as the fitness score.

The algorithm then enters two phases - exploration and exploitation. In the exploration phase, ospreys update their positions by randomly comparing themselves with others and searching for better solutions. In the exploitation phase, ospreys refine their hyperparameters by moving closer to the current best solution. Fitness is recalculated after each phase, and the best-performing osprey is updated if a better one is found. It continues until it satisfies the maximum number of iterations or convergence is met and returns the optimal hyperparameters

4 RESULT AND DISCUSSION

The proposed model is coded in Python and verified to predict key environmental parameters using historical data from three distinct domains: air quality, water quality, and plastic waste generation in India. The proposed model IS trained using an NVIDIA RTX 3060 GPU with 12GB RAM. The raw datasets underwent preprocessing to ensure quality and consistency. Missing timestamps were addressed using linear interpolation along with forward and backwards filling techniques. All features were then normalised to a [0, 1] range using Min-Max scaling. The OOA is configured with a population size of 20 and runs for 50 iterations. The performance was evaluated using metrics such as *MSE*, *RMSE*, R^2 Score, and *MAE*. It can be computed as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} | y_i - y'_i | \tag{17}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (y_i - y'_i)^2 \tag{18}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - y'_i)^2} \tag{19}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - y'_i)^2}{\sum_{i=1}^{n}(y_i - y')^2} \quad (20)$$

where, y_i is the actual value, y'_i is the predicted value and y' is the mean of the actual values, n is the number of data points. The first dataset used in this work is sourced from the Central Pollution Control Board (CPCB), available on the official CPCB - Air Quality (https://cpcb.nic.in/) website. The dataset consists of hourly and daily air quality measurements across various cities in India.

The second dataset used was obtained from Kaggle and is available at Indian Water Quality Data (https://www.kaggle.com/datasets/anbarivan/indianwater-quality-data)). This dataset includes several pollutants with the values representing average measurements over time. The third dataset focused on plastic waste generation, sourced from the National Informatics Centre (NIC) and available at Plastic Waste Data (https://www.data.gov.in/keywords/waste). The dataset covers data from 2018 to 2024, detailing the amount of plastic waste generated across various states of India.

The visualisation of air quality data is shown in Fig. 3. The validation loss of the proposed model for air quality prediction is given in Fig. 4. The validation accuracy closely follows the training accuracy and shows minimum overfitting. Overall, the convergence of both curves indicates that the model is performing consistently well on both training and validation data and achieves high accuracy levels by the end of the training process.

Chennai	########	5	988.01	0	28.79	52.93	14.19	55.68	68.43	4.12
Chennai	########	5	1628.88	1.22	57.58	10.82	22.65	86.87	108.78	8.11
Chennai	########	5	1789.09	7.94	54.84	1.13	23.6	99.69	119.51	11.15
Chennai	########	5	1508.71	3.02	48.67	4.65	19.31	120.98	139.1	9.63
Chennai	########	5	2536.77	48.28	49.35	0	43.39	136.72	168.81	37.49
Chennai	########	5	1321.79	6.82	42.84	1.29	16.45	72.45	89.57	8.74
Chennai	########	5	974.66	1.22	31.87	8.76	11.09	55.94	69.25	6.27
Chennai	########	5	1468.66	17.43	38.73	0.05	20.74	75.87	98.02	17.73
Chennai	########	4	687.6	0.03	23.31	27.54	9.89	44.42	55.75	3.58
Chennai	########	5	1268.39	4.25	46.61	2.53	15.97	89.31	107.8	8.36
Chennai	########	5	1188.28	1.73	45.93	6.62	15.02	79.79	94.38	7.41
Chennai	########	5	2777.1	54.09	47.98	0	30.28	154.63	179.27	12.92
Chennai	########	5	1695.63	22.13	43.87	0.03	27.66	103.45	127.23	19.25
Chennai	########	5	1214.98	15.2	28.45	0.07	15.97	63.9	79.97	10.77
Chennai	########	3	507.36	0.67	13.71	8.67	6.91	23.74	29.98	2.44
Chennai	########	5	1388.55	32.63	14.91	0	19.79	51.63	63.88	8.49
Chennai	########	4	814.44	5.92	23.31	1.02	14.42	28.4	36.38	6.21
Chennai	########	2	423.91	0.2	14.91	15.74	9.18	11.86	17.61	2.98

Figure 3 Visualization of air quality dataset

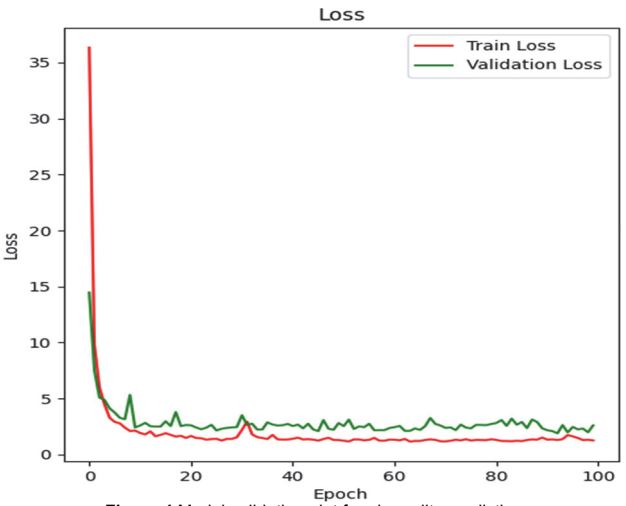

Figure 4 Model validation plot for air quality prediction

Table 1 Performance analysis of a proposed model for air quality prediction

Method	R^2	*MAE*	*MSE*	*RMSE*
Proposed	0.712	1.013	3.165	1.785
ARIMA	0.57	1.12	3.90	1.97
hybrid BiConvLSTM	0.60	1.08	3.60	1.89
CNN-LSTM	0.53	1.22	4.10	2.02
GA-KELM	0.50	1.35	4.30	2.08
CNN-GRU	0.54	1.28	3.95	1.99
Tuna Optimized CNN-LSTM	0.56	1.15	3.75	1.93
LSTM with BNN	0.52	1.30	4.20	2.05
AT-GCN	0.58	1.10	3.85	1.96
TD-CS-Transformer	0.59	1.13	3.95	1.98

The performance of the model for air quality prediction is given in Tab. 1. The proposed model attains the best solution with an R^2 score of 0.712 which indicates good predictive accuracy and the lowest *MSE* of 3.16 and *RMSE* of 1.785. This denotes relatively low error rates. The ARIMA model shows a lower R^2 value of 0.57 and higher error metrics values of *MSE* of 3.90 and *RMSE* of 1.97. It denotes the less accurate predictions. The Hybrid BiConvLSTM model achieves a moderate R^2 of 0.60, with MAE and MSE values of 1.08 and 3.60, respectively. This proves relatively low error but does not outperform the proposed method. The CNN-LSTM model also exhibits moderate performance with an R^2 of 0.53 and a higher *MSE* of 4.10 which denotes higher error rates. Models like CNN-GRU, Tuna Optimized CNN-LSTM, and LSTM with BNN perform similarly with R^2 scores ranging from 0.52 to 0.56, and their *MSE* and *RMSE* values are within a similar range from 3.75 to 3.95 for *MSE*. The AT-GCN and TD-CS-Transformer models show moderate results with R^2 values of 0.58 and 0.59, respectively. Overall, the proposed method stands out as the most effective and achieves the best balance of accuracy and low error metrics.

The visualization of waste generation data is shown in Fig. 5. The visual representation of the frequency of different WQI values is given in Fig. 6. The WQI is a numerical scale used to assess the overall quality of water with higher scores indicating better quality.

	A	B	C	D	E	F	G	H	I	J
1	ph	Hardness	Solids	Chloramin	Sulfate	Conductiv	Organic_c	Trihalomet	Turbidity	Potability
2		204.8905	20791.32	7.300212	368.5164	564.3087	10.37978	86.99097	2.963135	0
3	3.71608	129.4229	18630.06	6.635246		592.8854	15.18001	56.32908	4.500656	0
4	8.099124	224.2363	19909.54	9.275884		418.6062	16.86864	66.42009	3.055934	0
5	8.316766	214.3734	22018.42	8.059332	356.8861	363.2665	18.43652	100.3417	4.628771	0
6	9.092223	181.1015	17978.99	6.5466	310.1357	398.4108	11.55828	31.99799	4.075075	0
7	5.584087	188.3133	28748.69	7.544869	326.6784	280.4679	8.399735	54.91786	2.559708	0
8	10.22386	248.0717	28749.72	7.513408	393.6634	283.6516	13.7897	84.60356	2.672989	0
9	8.635849	203.3615	13672.09	4.563009	303.3098	474.6076	12.36382	62.79831	4.401425	0
10		118.9886	14285.58	7.804174	268.6469	389.3756	12.70605	53.92885	3.595017	0
11	11.18028	227.2315	25484.51	9.0772	404.0416	563.8855	17.92781	71.9766	4.370562	0
12	7.36064	165.5208	32452.61	7.550701	326.6244	425.3834	15.58681	78.74002	3.662292	0
13	7.974522	218.6933	18767.66	8.110385		364.0982	14.52575	76.48591	4.011718	0
14	7.119824	156.705	18730.81	3.606036	282.3441	347.715	15.92954	79.50078	3.445756	0
15		150.1749	27331.36	6.838223	299.4158	379.7618	19.37081	76.51	4.413974	0
16	7.496232	205.345	28388	5.072558		444.6454	13.22831	70.30021	4.777382	0

Figure 5 Visualization of water quality data set

The validation loss of the proposed model for air quality prediction is given in Fig. 7. The consistent and small gap between the two curves suggests that the model is learning meaningful patterns rather than simply memorizing the training examples. Overall, the graph proves a successful training process for the proposed model.

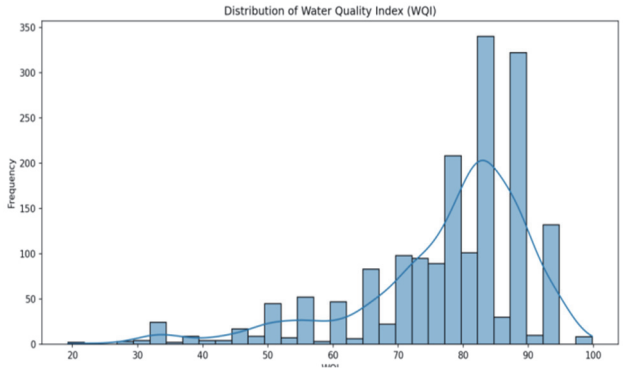

Figure 6 Visualization of frequency of different WQI values

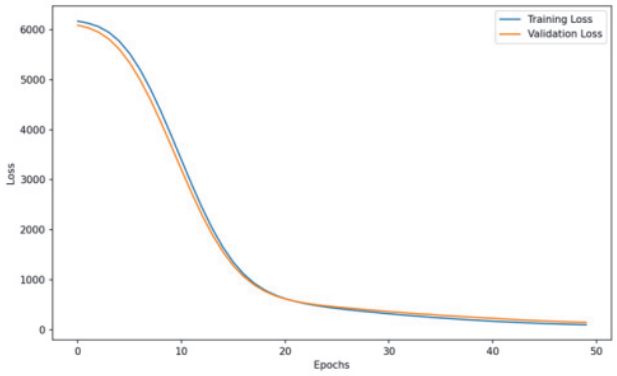

Figure 7 Model validation plot for water quality prediction

The performance of the model for water quality prediction is given in Tab. 2. The proposed model shows the best overall performance with an *MSE* of 128.50, an *RMSE* of 11.34, and an R^2 score of 0.54. This proves the strong predictive accuracy of the proposed model. The *MAE* for the proposed model is 7.40 which denotes a relatively low error in predictions. The GRU model performs slightly worse with an *MSE* of 145.23 and an R^2 score of 0.42. The Random Forest model with an *MSE* of 152.35 and R^2 of 0.40 shows a lower level of accuracy compared to the proposed model. The Stacking Ensemble Model performs better than the Random Forest with an *MSE* of 134.22 and R^2 of 0.49, showing a relatively good performance but still trailing behind the Proposed Model. The SVM model has an *MSE* of 160.30 and the lowest R^2 of 0.35 which denotes poor predictive accuracy. The PCA with the KNN model shows even worse with the highest *MSE* of 172.50 and the lowest R^2 score of 0.28. It denotes the significant limitations in prediction. The Hybrid Model RF+DT has an *MSE* of 180.10 and an R^2 of 0.25. Lastly, the CRNN+CNN model performs moderately with an *MSE* of 150.50 and R^2 of 0. 38 respectively.

Table 2 Performance analysis of the proposed model for water quality prediction

Model	*MSE*	*RMSE*	R^2 Score	*MAE*
Proposed	128.50	11.34	0.54	7.40
GRU	145.23	12.04	0.42	8.20
Random Forest	152.35	12.34	0.40	8.56
Stacking ensemble model	134.22	11.85	0.49	8.02
SVM	160.30	12.66	0.35	9.20
PCA with KNN	172.50	13.14	0.28	9.80
Hybrid model RF+DT	180.10	13.42	0.25	10.00
CRNN+CNN	150.50	12.26	0.38	8.70

The visualization of waste generation data is shown in Fig. 8. The training and validation loss curve for waste generation prediction is given in Fig. 9. Both the training and validation loss decreased sharply which showed the model is learning quickly during the training process. The model is learning effectively and generalizing well to unseen data and not overfitting.

	A	B	C	D
1	Month	Plastic Waste Generated (Tonnes)		
2	2017-18-01	42722.33		
3	2017-18-02	33720		
4	2017-18-03	75007.04		
5	2017-18-04	74200.41		
6	2017-18-05	21120.8		
7	2017-18-06	3462.1		
8	2017-18-07	38715.52		
9	2017-18-08	80392.52		
10	2017-18-09	88177.68		
11	2017-18-10	89977.36		
12	2017-18-11	65898.37		
13	2017-18-12	47392.87		
14	2018-19-01	389911.54		
15	2018-19-02	257450.45		
16	2018-19-03	245016.03		
17	2018-19-04	46130.79		
18	2018-19-05	366998.1		
19	2018-19-06	314671.5		

Figure 8 Visualization of waste generation data set

The plastic waste generated over a period of 30 days is predicted by the proposed model. The predicted versus real plastic waste generation is given in Fig. 10. A key observation is that the predicted values (blue line) generally follow the trend of the real values (red line) which denotes that the model is capturing the overall pattern of plastic waste generation. Both lines show a rise in waste generation up to around day 15, followed by a decline, and then a slight increase towards the end of the period. This suggests the model is considerably effective at capturing the cyclical nature of the data.

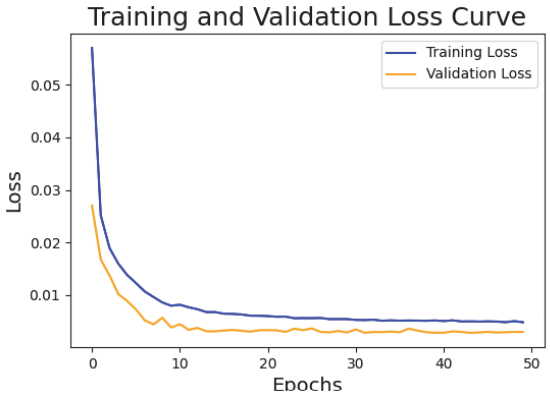

Figure 9 Model validation plot for waste generation prediction

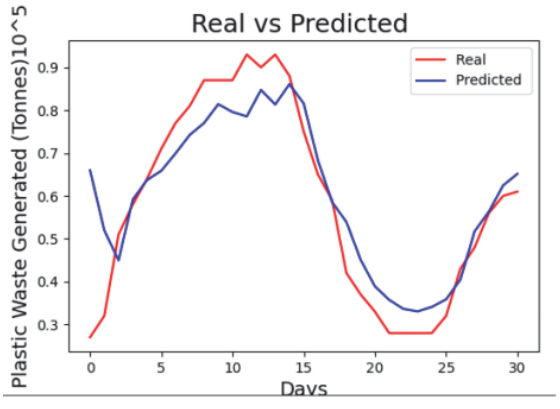

Figure 10 Predicted waste generation values for the upcoming 30 days

The performance of the model for waste generation prediction is given in Tab. 3. The proposed model shows the best performance with the lowest *MSE* of 0.01016, *RMSE* of 0.10082 and *MAE* of 0.07079 along with a high R^2 score of 0.8025. This denotes strong predictive accuracy. In comparison, CNN with Self-Attention achieves an *RMSE* of 0.26457, with a lower R^2 score of 0.5300, which indicates moderate predictive performance. The MWG model has an *MSE* of 0.09000 and an *RMSE* of 0.30000, which indicates weaker predictive capabilities. The GNN model performs slightly better than MWG with an MSE of 0.06000 and an R^2 score of 0.5400. The Graph-LSTM and GRU model show decent performance with an R^2 score of 0.7690 and an *RMSE* of 0.19142. The EEMD+DL method performs better than the hybrid models with an *MSE* of 0.01500 and shows a moderate R^2 score of 0.7500. The Hybrid Model and CNN+GRU exhibit balanced performance, with the Hybrid model performing slightly better in terms of *RMSE* and *MAE*.

Table 3 Performance analysis of the proposed model for waste generation prediction

Model	*MSE*	*RMSE*	R^2 Score	*MAE*
Proposed Model	0.01016	0.10082	0.8025	0.07079
CNN+ self-attention	0.07000	0.26457	0.5300	0.19000
MWG	0.09000	0.30000	0.4800	0.22000
GNN	0.06000	0.24494	0.5400	0.18000
Graph-LSTM and GRU	0.05000	0.19142	0.7690	0.18000
EEMD+DL	0.01500	0.12247	0.7500	0.11000
Hybrid model	0.04200	0.16954	0.7520	0.18500
CNN+GRU	0.01800	0.13416	0.7300	0.12000

Table 4 Ablation study of the proposed model

Model Variant	*MSE*	*RMSE*	R^2 Score	*MAE*
Full Model (CapsNet + GRU + OOA)	43.875053	4.40827	0.6845	2.16193
Model without CapsNet	55.325	5.535	0.5300	3.056
Model without GRU	65.214	6.098	0.4000	4.102
Model without OOA (No Optimization)	78.142	7.202	0.3000	5.215
Model with only Conv1D	70.876	6.345	0.4500	4.178

To perform an ablation study for the proposed model, the impact of each component on the model's performance is analyzed. The Full Model (CapsNet + GRU + OOA) is the baseline configuration and performs the best among all variants. The ablation study results of the proposed model are given in Tab. 4. The model effectively captures both spatial and temporal features from the data with Capsule Networks and GRUs working together. Additionally, the OOA fine-tunes the model's hyperparameters which ensures it generalizes well and delivers strong predictions. The performance metrics show good results with an *MSE* of 43.875053, *RMSE* of 4.40827, R^2 Score of 0.6845, and *MAE* of 2.16193.

The Model without CapsNet removes the CapsNet layer which leaves only the Conv1D and GRU components. The model struggles to extract spatial features effectively without CapsNet which causes an increase in the error metrics. The *MSE* rises to 55.325, *RMSE* to 5.535, and the R^2 Score drops to 0.5300, showing a noticeable performance degradation. The *MAE* also increases to 3.056 which denotes that the spatial feature extraction from CapsNet plays a crucial role in model performance.

The Model without GRU excludes the GRU layers where the model no longer captures the temporal dependencies in the data. As expected, this results in a further decrease in performance with an *MSE* of 65.214 and *RMSE* of 6.098. The R^2 Score drops significantly to 0.4000, and the *MAE* increases. This proves the importance of GRUs for understanding time-series data. The Model without OOA leads to the worst performance among all variants, with an *MSE* of 78.142, *RMSE* of 7.202, and an R^2 Score of 0.3000. The *MAE* also spikes to 5.215 which denotes that hyperparameter optimization is crucial for improving model accuracy and generalization.

Finally, the Model with only Conv1D uses only the Conv1D layer for feature extraction without CapsNet or GRU. This version captures some spatial features but lacks the ability to model temporal dependencies. As a result, it performs better than the models without CapsNet or GRU, but still underperforms compared to the full model, with an *MSE* of 70.876, *RMSE* of 6.345, and an R^2 Score of 0.4500. The *MAE* is 4.178, which denotes that Conv1D alone is not sufficient to achieve the optimal performance in this complex multi-domain prediction task. This ablation study demonstrates the significance of each component in the model, where CapsNet for spatial feature extraction GRU to capture temporal dependencies and OOA for optimizing hyperparameters.

Table 5 Computational analysis of the proposed model

Metric	Value
Training Time per Epoch	~4.5 seconds
Total Training Time	~220 seconds (50 epochs)
Model Parameters	~1.3 million
Inference Time per Sample	~6 milliseconds

The computational analysis of the proposed model is given in Tab. 5. Despite its hybrid nature, the model maintains a reasonable computational cost due to efficient capsule routing (3 iterations) and light GRU layers (64 units each). The OOA's population-based search (20 ospreys for 30 iterations) also remained computationally feasible due to early convergence.

5 CONCLUSION

In this work, an innovative smart city environment prediction is proposed. It integrates Capsule Networks with GRUs for effective prediction. The combination of these two advanced architectures allows the model to capture both spatial and temporal features from the data. In addition, the parameter tuning using OOA supports improving the accuracy further. The implementation results indicate that the proposed model effectively captures hidden patterns for accurate prediction. Future work will focus on expanding the dataset and will apply different architectures in practical scenarios to validate its effectiveness in real-time classification tasks.

6 REFERENCES

[1] Meng, H., Song, J., Huang, L., Zhu, Y., Zhu, M., & Zhang, J. (2024). Water Quality Prediction Method Based on OVMD and Spatio-Temporal Dependence. *Technical Gazette*, *31*(1), 286-295. https://doi.org/10.17559/TV-20230519000647

[2] Malche, T., Maheshwary, P., & Kumar, R. (2019). Environmental Monitoring System for Smart City Based on Secure Internet of Things (IoT) Architecture. *Wireless Personal Communications*, *107*, 2143-2172. https://doi.org/10.1007/s11277-019-06376-0

[3] Ullo, S. L. & Sinha, G. R. (2020). Advances in Smart Environment Monitoring Systems Using IoT and Sensors. *Sensors*, *20*(3113). https://doi.org/10.3390/s20113113

[4] Tahat, A., Aburub, R., Al-Zyoude, A., & Talhi, C. (2018). A Smart City Environmental Monitoring Network and Analysis Relying on Big Data Techniques. *Proceedings of the 2018 International Conference on Software Engineering and Information Management (ICSIM '18)*, 82-86. https://doi.org/10.1145/3178461.3178464

[5] Gupta, N. K., Mishra, A., Shukla, P. K., Sharma, A., & Pandey, S. (2023). Smart Environment Monitoring Using Internet of Things and Big Data. *Proceedings of the 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN)*, 1166-1170. https://doi.org/10.1109/ICPCSN58827.2023.00198

[6] Liu, L. & Zhang, Y. (2021). Smart environment design planning for smart city based on deep learning. *Sustainable Energy Technologies and Assessments*, *47*, 101425. https://doi.org/10.1016/j.seta.2021.101425

[7] Lakshmi, S. & Krishnamoorthy, A. (2024). Effective Multi-Step PM2.5 and PM10 Air Quality Forecasting Using Bidirectional ConvLSTM Encoder-Decoder with STA Mechanism. *IEEE Access*, *12*, 179628-179647. https://doi.org/10.1109/ACCESS.2024.3509142

[8] Cao, W., Qi, W., & Lu, P. (2024). Air Quality Prediction Based on Time Series Decomposition and Convolutional Sparse Self-Attention Mechanism Transformer Model. *IEEE Access*, *12*, 155340-155350. https://doi.org/10.1109/ACCESS.2024.3484579

[9] Yang, J., Ismail, A. W., Li, Y., Zhang, L., & Fadzli, F. E. (2023). Transfer Learning-Driven Hourly PM2.5 Prediction Based on a Modified Hybrid Deep Learning. *IEEE Access*, *11*, 99614-99627. https://doi.org/10.1109/ACCESS.2023.3314490

[10] Liu, C., Pan, G., Song, D., & Wei, H. (2023). Air Quality Index Forecasting via Genetic Algorithm-Based Improved Extreme Learning Machine. *IEEE Access*, *11*, 67086-67097. https://doi.org/10.1109/ACCESS.2023.3291146

[11] Nguyen, H. A. D. (2023). Long Short-Term Memory Bayesian Neural Network for Air Pollution Forecast. *IEEE Access*, *11*, 35710-35725. https://doi.org/10.1109/ACCESS.2023.3265725

[12] Iskandaryan, D., Ramos, F., & Trilles, S. (2023). Graph Neural Network for Air Quality Prediction: A Case Study in Madrid. *IEEE Access*, *11*, 2729-2742. https://doi.org/10.1109/ACCESS.2023.3234214

[13] Wang, J., Jin, L., Li, X., He, S., Huang, M., & Wang, H. (2022). A Hybrid Air Quality Index Prediction Model Based on CNN and Attention Gate Unit. *IEEE Access*, *10*, 113343-113354. https://doi.org/10.1109/ACCESS.2022.3217242

[14] Luo, J. & Gong, Y. (2023). Air pollutant prediction based on ARIMA-WOA-LSTM model. *Atmospheric Pollution Research*, *14*(6), 101761. https://doi.org/10.1016/j.apr.2023.101761

[15] Gunasekar, S., Kumar, J. R., & Kumar, Y. D. (2022). Sustainable optimized LSTM-based intelligent system for air quality prediction in Chennai. *Acta Geophysica*, *70*, 2889-2899. https://doi.org/10.1007/s11600-022-00796-6

[16] Kushwah, V. & Agrawal, P. (2025). Hybrid model for air quality prediction based on LSTM with random search and Bayesian optimization techniques. *Earth Science Informatics*, *18*, 32. https://doi.org/10.1007/s12145-024-01514-0

[17] Aslam, B., Maqsoom, A., Cheema, A. H., Ullah, F., Alharbi, A., & Imran, M. (2022). Water Quality Management Using Hybrid Machine Learning and Data Mining Algorithms: An Indexing Approach. *IEEE Access*, *10*, 119692-119705. https://doi.org/10.1109/ACCESS.2022.3221430

[18] Johar, A., Lahari, S. A., & RN, P. (2023). Water Quality Index Prediction System Using IoT Enabled Sensor Suite and Machine Learning. *Proceedings of the 16th International Conference on Sensing Technology (ICST)*, 1-6. https://doi.org/10.1109/ICST59744.2023.10460817

[19] Ehteram, M., Ahmed, A. N., Sherif, M., & El-Shafie, A. (2024). An advanced deep learning model for predicting water quality index. *Ecological Indicators*, *160*, 111806. https://doi.org/10.1016/j.ecolind.2024.111806

[20] Prasad, D. V. V., Venkataramana, L. Y., Kumar, P. S., Prasannamedha, G., Harshana, S., Srividya, S. J., Harrinei, K., & Indraganti, S. (2022). Analysis and prediction of water quality using deep learning and auto deep learning techniques. *Science of the Total Environment*, *821*, 153311. https://doi.org/10.1016/j.scitotenv.2022.153311

[21] Shams, M. Y., Elshewey, A. M., El-kenawy, E. S. M. et al. (2024). Water quality prediction using machine learning models based on grid search method. *Multimedia Tools and Applications*, *83*, 35307-35334. https://doi.org/10.1007/s11042-023-16737-4

[22] Chen, H. & Yunus, A. P. (2025). A novel predictive framework for water quality assessment based on socio-economic indicators and water leaving reflectance. *Groundwater for Sustainable Development*, *28*, 101405. https://doi.org/10.1016/j.gsd.2025.101405

[23] Pang, H., Ben, Y., Cao, Y., Qu, S., & Hu, C. (2025). Time series-based machine learning for forecasting multivariate water quality in full-scale drinking water treatment with various reagent dosages. *Water Research*, *268*(B), 122777. https://doi.org/10.1016/j.watres.2024.122777

[24] Wong, W. Y., Hasikin, K., Khairuddin, A. S. M., Razak, S. A., Hizaddin, H. F., Mokhtar, M. I., & Azizan, M. M. (2023). A Stacked Ensemble Deep Learning Approach for Imbalanced Multi-Class Water Quality Index Prediction. *Computers, Materials & Continua*, *76*(2), 1361-1384. https://doi.org/10.32604/cmc.2023.038045

[25] Talukdar, S., Shahfahad, Ahmed, S., Naikoo, M. W., Rahman, A., Mallik, S., Ningthoujam, S., Bera, S., & Ramana, G. V. (2023). Predicting lake water quality index with sensitivity-uncertainty analysis using deep learning algorithms. *Journal of Cleaner Production*, *406*, 136885. https://doi.org/10.1016/j.jclepro.2023.136885

Contact information:

JANAKI RAMAL P.
(Corresponding author)
Department of CSE,
Saveetha School of Engineering,
Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamil Nadu, India
E-mail: janakiramal.p@hotmail.com

ANBALAGAN E.
Department of CSE,
Saveetha School of Engineering,
Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamil Nadu, India