JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

Vol. 6 No.1 (2025) 319-332
https://publisher.uthm.edu.my/ojs/index.php/jscdm

JSCDM

An IoT-Based Active Learning Convolutional Neural Networks Model for Predicting River Water Quality

Taher M. Ghazal[1]*

[1] *Department of Networks and Cybersecurity, Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman, 19111, JORDAN*

Corresponding Author: taher.ghazal@ieee.org
DOI: https://doi.org/10.30880/jscdm.2025.06.01.021

Article Info

Received: 27 April 2025
Accepted: 25 June 2025
Available online: 30 June 2025

Keywords

River water quality, prediction, convolutional neural network, supervised learning, active learning

Abstract

The conventional methods used to assess river water quality involve slow processes that incur significant costs and fail to provide immediate results. Without automated predictive systems, there is no means to detect pollution trends in advance. The goal of this investigation is to develop an Internet of Things (IoT)-driven deep learning model that enables real-time predictions of river water quality using Deep Neural Network (DNN) algorithms with enhanced precision and effectiveness. This study employs an IoT system to acquire pH parameters and utilizes a DNN to predict water quality in the Tigris River of Iraq. Subsequently, an IoT-based Active Learning Convolutional Neural Network (IoT-ALCNN) model is developed to predict river water quality. The model is trained through a backpropagation algorithm that incorporates an active learning mechanism. This mechanism provides an interface that enables users to update the training data, thereby enhancing the learning process. The performance of the IoT-ALCNN model is evaluated using classification assessment metrics. The model's fit to the data is assessed through the relationship between the residuals and the updated values. Test results for the IoT-ALCNN model indicate that it can be utilized as a method for predicting and monitoring river water quality, achieving a top accuracy of 98.72% and surpassing other baseline models in terms of both generalization and efficiency.

1. Introduction

Rivers exhibit a decline in water quality due to human activities and the effects of urbanization in the surrounding areas, resulting in increased pollutant volumes. River water remains filled with sludge and garbage due to the ineffective enforcement controls maintained by local authorities. The rivers serve as vital habitats that accommodate fish and wildlife reproduction and feeding, since these contaminants enter their waters [1]. Life in water at a particular pH level becomes detrimental to most aquatic species when they encounter minimal pH variations. All aspects of human life, as well as ecosystem health, depend on water, making water quality management a critical issue. Water quality parameter evaluations enable assessment operations to perform more effectively and aid in creating improved strategies for water resource management and planning [2].

The prediction of water pollution depends on statistical or machine learning techniques, which form the basis of water quality modeling. The classical process-based modeling system or process-based model approach delivers acceptable forecasts for river water quality measurement outcomes. The models operated on lengthy datasets, requiring substantial input information, which is sometimes unavailable to the system. Models necessitate approximate calculations of various procedures, but such estimated calculations may overlook essential components that affect water processes [3]. Data-driven techniques serve as an effective alternative to traditional process-based modeling methods. Computationally fast data-driven models require fewer input parameters compared to process-based models. Supervised Learning (SL) has made significant contributions to various fields, including environmental science, such as water quality evaluation, and the medical field, as seen in the work of [4].

The Artificial Neural Networks (ANN) function as one of the top SL algorithms. ANN stands as a machine-operated computational method. Proof has been established regarding the capabilities and applicability of ANN in modeling various physical water engineering phenomena [5]. The technical approach works best on issues that require manipulating numerous parameters together with non-linear interpolation functions. The model proves difficult to handle through traditional mathematical and theoretical approaches. The nonlinear structure of NN has been utilized to extract the spatial and unsteady characteristics of the investigated problems more effectively than conventional modeling approaches. Convolutional Neural Networks (CNNs) represent a form of Artificial Neural Network (ANN) that provides three extraordinary capabilities, including detailed learning and adaptation, as well as generalization potential [4].

Pu et al [5] established a water quality association with the spectral and spatial features of multispectral data using CNNs. The input data for CNN included remote sensing photographs, which received their corresponding water quality measurements as output labels. Remote sensing pictures have a direct relationship with in situ water quality levels that a CNN network architecture can model. Multiple remote sensing products and in situ measurements of water quality, conducted by Chinese government agencies, are combined to establish three-tiered classification systems of inland lakes. This research study completed two main goals: (1) A CNN model worked to map the Landsat8 image-in-year water-quality correlation independently from water-quality dependent variable regression, and (2) it used a transfer-learning technique to increase inland Remote Sensing classification accuracy. Remote Sensing. 2019, 11, 1674 The research included three lakes with restricted in situ measurement availability from a total of fourteen lakes. The water quality classification for a water body relies on a CNN model, which utilizes Landsat images and water-quality measurements from corresponding locations and times. Each model contained a different number of CNN layers, ranging from two to seven layers. The accuracy results showed the two-layer model scored 89.90%, while the accuracy level of the four-layer model reached 92.49%.

Oga et al [6] suggested a technique for estimating the water quality of a river using CNNs, a semantic segmentation approach, and a novel dataset. First, CNN architectures are trained using a collection of natural river photos assigned the classifications "Clear" and "Muddy." In the pre-processing, the training photos are cropped using a semantic segmentation algorithm. Second, the proposed model estimates the river's water quality. As with training, input photos are pre-processed, and trained models identify input images as "Clear" or "Muddy." Their objective is to evaluate the CNN standard against their suggested upgrade. A variety of CNN variants, including AlexNet, GoogleNet, VGGNet, etc., were used. They have shown a significant improvement between the regular CNN and their suggested one; for example, the AlexNet model has achieved an accuracy of 76%, compared to 86% after the augmentation. Consequently, their best model was ResNexT, which had a standard form accuracy of 83% that was improved to 92%.

It is clear from earlier work that the IoT prediction of water quality is possible only through the real-time acquisition of distributed sensor networks deployed along rivers and water bodies. These key physicochemical parameters (pH, turbidity, dissolved oxygen, temperature, and conductivity) are constantly monitored using sensors, which wirelessly transmit the data to cloud-based platforms for processing and drawing useful conclusions based on the key parameters [7], [8]. Elevating temporal and spatial data resolution while enabling early detection of pollution events increases the potential to manage resources more effectively, thereby preventing or reducing the impacts of future pollution events. Integrating advanced analytics and deep learning algorithms, including Convolutional Neural Networks (CNNs), in IoT systems can help predict water quality trends with precision, enabling timely interventions and improving water resource management [9].

SL approaches to active learning select or choose the most informative or uncertain data samples from an unlabeled dataset and ask them to be labeled by an oracle, for example, a human expert unlike traditional supervised learning, which assumes a large quantity of labeled data, active learning attempts to achieve high accuracy by reducing the quantity of labeled instances and utilizing data points that will contribute to improving the model's performance [10], [11]. However, it is especially important in the domain when labeling the data is expensive, time-consuming, or requires domain expertise. Active learning is also combined with deep learning models like CNNs to make deep learning more efficient and generalize better on problems like image classification, natural language processing, environmental monitoring, and most other data science tasks [12], [13].

This research is motivated by the ability of the CNN to perform predictions on complex environmental monitoring real-world problems. Several points along the river were selected, and a monitoring network of IoT-enabled sensors was deployed to measure key water quality parameters, including pH, turbidity, temperature, dissolved oxygen, and electrical conductivity. However, the data from these sensors, transmitted in real-time, was fed into a cloud storage process. First, since the collected dataset contained noise, we removed that noise to ensure data consistency and reliability. Additionally, we normalized the data and imputed missing values. It then investigates and applies the capability of the CNN algorithm with an active learning mechanism, in order to predict river water quality. Through this process, we demonstrate that active learning is crucial for reducing the prediction error of CNNs for water quality analysis by selecting the most informative data points to label, thereby minimizing the required label effort. Acquiring labeled data in water quality monitoring is expensive and often time-consuming; particular specimens may require expert analysis or a field trip to sample. In response to this challenge, active learning addresses it by exploiting the idea of iteratively querying the most uncertain or diverse samples from an unlabeled dataset and then labeling the queried samples to retrain the CNN, thereby achieving faster convergence and better generalization. This leads to a highly efficient model for the characterization, particularly if the data is distributed in a changing environment such as seasonal or pollution variations.

This paper breaks down the development and evaluation of an IoT-based active deep learning CNN model for predicting river water quality into several important sections. The paper begins with an Introduction that explains the importance of real-time water quality monitoring and the shortcomings of existing techniques. Previous studies on IoT and Deep Learning applications in environmental monitoring are reviewed in the Literature Review, and gaps in the research are pointed out that this study addresses within the context of these gaps. In the Materials and Methods section, the deployment of IoT sensors, data preprocessing, the CNN architecture, and the integration of active learning techniques for training the CNN are outlined. The experimental results are presented in the Results and Discussion section, where the model performance is evaluated using standard metrics, and the findings are compared with those of available models. Next, the Conclusion summarizes the main contributions, discusses the practical implications of the proposed system, and suggests further possibilities for improvement.

2. Materials and Methods

River water quality is effectively predicted using the proposed methodology that integrates IoT-based real-time sensing, data preprocessing, Convolutional Neural Network (CNN) modeling, and active learning. The methodology is broken down into 5 important steps: (1) sensor deployment and data collection, (2) data transmission and cloud integration, (3) data preprocessing, (4) model training with active learning, and (5) model evaluation. Figure 1 presents the steps of the research methodology for developing the IoT-based Active Learning Convolutional Neural Network (IoT-ALCNN) model.

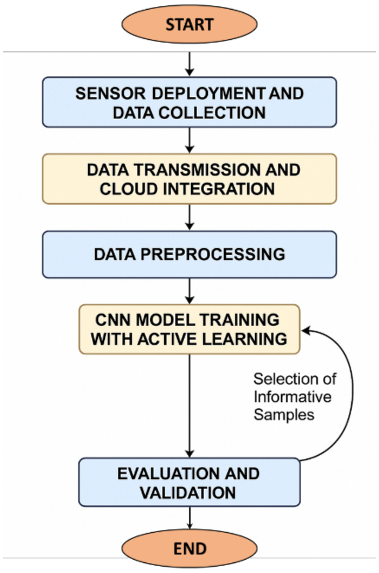


Fig. 1 *presents the research methodology steps*

- Sensor Deployment and Data Collection: A distributed network of IoT-enabled water quality sensors is deployed over selected monitoring points along a river to undertake sensor deployment and data collection. They are sensors that measure critical environmental parameters such as temperature, pH, turbidity, electrical conductivity, and dissolved oxygen. The sensor node gives each of them wireless transmission ability and a microcontroller that periodically collects and transmits data. The goal is to create a continuous and scalable data collection pipeline for the purpose of real-time monitoring.
- Data Transmission and Cloud Integration: Protocols, like MQTT or HTTP, are used for data transmission from the sensor for wireless data transmission to a cloud-based platform. It also provides a repository of real-time data aggregation, visualization, and remote access from this platform. The mechanisms for time-stamping and data integrity are applied to ensure reliable and chronological ordering. The infrastructure makes water quality data seamlessly available for downstream analytics and training models.

- Preprocessing Data: The Raw data usually contains noise, missing values, and outliers. Normalization, imputation of missing values, and filtering out inconsistent readings are steps involved in preprocessing. All of the samples are then labelled according to standard water quality indices (i.e., WHO or EPA guidelines), putting the samples into buckets for "Good", "Moderate", and "Poor" quality. The multidimensional sensor inputs are applied to feature engineering, so that the CNN be able to learn patterns from them effectively.
- CNN Model Training with Active Learning: The CNN Model is created with input layers of an equal number of features (i.e., water parameters) and a sequence of convolutional, pooling, and fully connected layers. The model is initially trained on a small amount of labeled data. To improve model performance, an active learning loop is incorporated, whereby the CNN identifies high-uncertainty samples from the unlabeled dataset, which are then manually labeled and appended to the training set. This enables optimizing the model with the smallest possible amount of labeled data and the highest possible accuracy.
- Evaluation and Validation: In this case, the trained model is evaluated using the standard metrics like accuracy, precision, recall, F1 score, and confusion matrix analysis. The dataset is split into training and testing, where the testing is carried out using 3-fold cross-validation. Comparisons are made to baseline methods like MLP, AL MLP, and CNN classifiers to show the degree to which the performance is improved. Robustness across different data splits is also ensured using the cross-validation approach.

2.1 Artificial Neural Networks

Artificial Neural Networks (ANN) function as information processing paradigms because they emulate brain processes that enable information processing within human brains. The extensive capabilities of the human brain generate excitement among researchers and computer scientists. The basic model behind an artificial neural network (ANN) utilizes three significant aspects of brain neuron structure. These include dendrites, which serve as receiving zones that connect with other neurons, and somas, which execute necessary nonlinear processing steps. These steps send output signals through axons, which function like transmission cables to reach downstream neurons. The functional conjunction between two neurons exists as a synapse. Synapses function as basic elements that connect neurons through their interneuronal networks. The primary signal transmitted by most real neurons relies on chemicals to generate spike-based electrical pulses. In ANNs, the spikes from biological neurons are converted into a continuous variable X_j that functions as an averaged pulse measurement.

The operating principle of ANN remains equivalent to its biological process. With numerous basic processors, it features layers that may contain individual memory storage connected through weighted interlinks. The goal of this system is to replicate neuron processing within specialized hardware elements or software structures that comprise multiple interconnected simple neuron components. The processing unit design of ANN allows good performance through its extensive network of basic computing cells. The ANN system develops its computational strength from numerous basic processing elements operating in a system that resembles the brain structure, with its extensive neural fibers distributed across each brain volume.

ANNs represent a universal set of non-linear models that engineers and scientists have proven effective across various research fields. The applications include real-time predictions of time series [14], [15] and image processing [16] alongside pattern identification [17] and medical image analysis [18], besides system optimizations [19] and many other uses. Applications from various domains fall into distinct areas, including regression and generalization, classification, association and clustering, pattern completion, and optimization.

The Multi-Layer Perceptron (MLP) creates a feedforward network by connecting summing units through their linked weights. Multiple perceptron layers form hierarchical structures in this network, which solves the problems of single-layer networks. MLP connects different layers to process data sent from the input nodes to the output nodes through one or more hidden layers positioned between the input and output. Network processing occurs between external inputs and output results because hidden nodes provide functional operation. The drawing in Figure 2 shows the arrangement of a single-layer MLP network. The network figure represents a complete connection because each node from each layer connects to all nodes in the following layer. Bias nodes are not shown here for simplicity. MLP computes the network output according to the following equation:

$$Y = \sigma(\sum_{j=1}^{J} W_{jk}\,\sigma(\sum_{i=1}^{N} W_{ij}X_i + W_{oj}) + W_{ok}) \tag{1}$$

where x_i denotes the input value, W_{ij} is the weights from the input layer to the hidden layer, W_{jk} is the weights from the hidden layer to the output layer, W_{oj} is a bias for the hidden node, σ is a sigmoid transfer function, and Y is the network output.

The bias term W_{oj} is critical in increasing the adaptability and the learning capacity of the model. Although the weight matrix W_{ij} identifies the strength and direction of connections across the layers, the bias shifts the activation curve, allowing the network to approach the data more precisely. In the absence of the bias, all activation functions would be required to pass through the origin, severely limiting the kind of patterns the MLP can learn. Practically, the bias term means that each neuron can learn an offset, so that the network can represent non-centered, or not strictly linearly-separable, data. The biases and weights are updated during training through backpropagation, which significantly contributes to minimizing the loss and enhancing generalization. This way, biases ensure the activation of neurons even when input signals are low or absent, making the network less prone to fluctuations in sensor content.

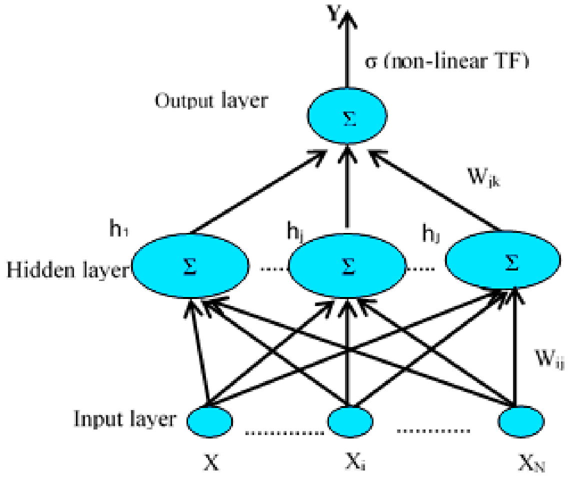


Fig. 2 *The Multi-Layer Perceptron (MLP) ANN architecture*

The network topology of MLP features dense interconnections because each input node links with all nodes in the first hidden layer, which further extends to every unit in consecutive layers. The input nodes distribute values to the first nodes within the hidden layer. The forward propagation extends from one hidden layer to the next, ultimately generating the network's output at the output layer. MLP demonstrates the ability to approach reasonable functions with high accuracy through one hidden layer by employing sufficient sigmoid-activated hidden nodes [20] [21].

The acquisition of meaningful forecast capabilities depends on training an NN with a specific learning algorithm as the method for conducting the learning process. A learning algorithm enables error calculation after presenting the training vector, allowing gradient descent to be performed on this error value. Supervised training of MLP uses the BP algorithm because it has become the most popular learning algorithm. The algorithm possesses two notable features: straightforward local computation and performing weight-update stochastic gradient descent through pattern learning, as described in [22].

As a supervised learning method, the BP algorithm minimizes an appropriate error function through gradient descent, utilizing the actual and desired network outputs to determine cost values. A weight calculation method based on iterative training determines the influence of the network weights. By selecting simple gradient descent, the error minimization aims to perform weight adjustments in the direction opposite to the gradient slope. The error showed the fastest decrease along this particular direction. The weights receive corrections through the implementation of the Delta Rule. This procedure entails subtracting the actual network output from the desired output in the given example. The weights receive adjustments that steer the network output nearer to its expected outcome. The formulated error function consists of:

$$E = \frac{1}{2}\sum_k (t_k - y_k)^2 \tag{2}$$

where t_k is the desired output and y_k is the network output. Each component of the gradient provides the slope of the error function with respect to that weight:

$$\frac{\partial E}{\partial W} = \left(\frac{\partial E}{\partial W_0}\right), \frac{\partial E}{\partial W_1}, \ldots, \frac{\partial E}{\partial W_n} \tag{3}$$

The partial derivative of the error function with respect to the weights and biases in the BP algorithm is determined as follows:

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial S_i}\frac{\partial S_i}{\partial net_i}\frac{\partial net_i}{\partial W_{ij}} \tag{4}$$

where W_{ij} is the weight from neuron j to neuron i, S_i is the neuron output, and neti is the weighted sum of the inputs of neuron i.

Given the gradient, each weight is adjusted by the negative of the gradient to reduce the error. The value of the derivative is then used to minimize the error function by performing a gradient descent as below:

$$W_{ij}(t+1) = W_{ij}(t) - \varepsilon \frac{\partial E}{\partial W_{ij}}(t) \tag{5}$$

The learning step's control mechanism is handled by the parameter ε, which determines the critical factor that affects convergence time. Extremely high learning rates can produce weight space oscillations, which often result in reaching suboptimal local minima instead of the global maximum. A small learning rate value causes training to become slow since many weight adjustments are necessary. The addition of momentum terms is a common solution to control the effects of previous derivatives on current calculations, thereby making the learning process stable.:

$$\Delta W_{ij}(t) = -\varepsilon \frac{\partial E}{\partial W_{ij}}(t) + \mu \Delta w(t-1) \tag{6}$$

where μ is the momentum term.

The momentum term serves as a solution to prevent oscillation that occurs in situations with high learning rates. The weight adjustment in each round maintains a tiny part of its previous change direction. The weights act as if they possess permanent inertia. Optimizing the convergence rate and preventing the algorithm from becoming stuck in local minima becomes possible through the application of momentum in the BP algorithm.

2.2 Convolutional Neural Network

ANN of the CNN type exists as one of the network variants. The data processing capabilities of SL systems, specifically CNNs, enable deep learning tasks to generate information and provide descriptions. Machine vision is the most frequent application of CNNs, as it encompasses image and video recognition, paired with recommender systems and natural language processing, according to [27] and [28]. CNN operates using a technology equivalent to multilayer perceptron architecture, which optimizes processing requirements. The CNN is built from three component layers, including an input layer and an output layer, and a hidden layer with multiple convolutional layers, fully connected layers, normalizing layers, and pooling layers. The elimination of constraints and improvement in efficiency for image processing result in a system that is significantly more powerful, easier to train, and more flexible. This applies to both natural language processing and image processing [29].

The MLP and CNN-based algorithm for predicting river water quality starts by collecting labeled sensor data associated with parameters such as pH, turbidity, temperature, and dissolved oxygen (DO). It normalizes and reshapes this data into a 1D array suitable for convolution. The CNN model processes a series of 1D convolutional layers to automatically learn spatial correlations among the features, utilizing ReLU activation and max pooling to reduce complexity and emphasize dominant patterns. It then filters the feature maps through one or more fully connected layers. It classifies the outcome using a softmax activation function, which outputs the probability that the input belongs to one of the predefined water quality classes [30].

The CNN structure for handling time-varying sensor data is a series of layers specifically designed to capture and abstract temporal features. It begins with a 1D convolutional layer that has 64 filters, each of size 3, with a stride of 1, 'same' activation mode, and 'same' padding to detect low-level temporal patterns. This is followed by a second 1D convolutional layer, which features 128 filters, also with a size of 3, a stride of 1, the same padding, and ReLU activation, to extract deeper features. The feature map is then downsampled, and its dimensionality is reduced using a 1D max pooling layer with a pool size of 2 and a stride of 2. A dropout layer (i.e., at a rate of 0.5) is added to prevent overfitting. The output is then flattened and passed through a densely connected layer with 128 units, which includes ReLU activation. Finally, the model has either a softmax output layer (for multi-class classification). This setup is a tradeoff among feature richness, temporal sensitivity, and generalization performance.

Categorical cross-entropy loss is used to train the model, with optimization through algorithms such as Adam and Adamax [31]-[33]. Common active learning uncertainty sampling strategies include entropy-based sample selection, where the goal is to label the data points that the model is most uncertain about. This approach estimates uncertainty by calculating the entropy of the class probability estimates for each unlabeled example. Specifically, for a sample x, the entropy is computed as

$$H(x) = -\sum_{i=1}^{k} p_i(x) \log p_i(x) \tag{7}$$

where $p_i(x)$ is the predicted value of the class i over the k classes.

Samples with maximized entropy are deemed the most ambiguous, indicating that the model is least confident in its predictions; therefore, these samples are selected for annotation. This method effectively improves model performance by concentrating on informative examples. Algorithm 1 presents the CNN-based numerical prediction algorithm for water quality.

Algorithm 1: The CNN-based numerical prediction for the water quality algorithm

Input:
- Feature matrix $X = \{ x^{(1)}, x^{(2)}, \dots, x^{(n)} \}$, where $x^{(i)} \in R\ d$ $x^{(i)} \in R^d$ represents a water quality sample with d features (e.g., pH, 2)
- Target value $Y=\{y^{(1)}, y^{(2)}, \dots, y^{(n)}\}$, where $y^{(i)} \in R$ y(i) ∈R for regression or $y(i) \in \{ 1 , \dots , C \}$ $y^{(i)} \in \{1,\dots,C\}$ for classification;
- Hyperparameters: learning rate η, batch size b, number of epochs E;

Output:
- Trained CNN model $f\ \theta : X \rightarrow Y$;
- Predicted output $y = f\ \theta\ (x)$;
- Evaluation Metrics: Accuracy, Loss, Precision, Recall, F1-score;

Data Preprocessing:

$x_j^{(i)} = \frac{x_j^{(1)} - \min(x_j)}{\max(x_j) - \min(x_j)}, \forall j=1,\dots,d;$

Reshape X to input shape $X \in R^{n \times d \times 1}$ for 1D CNN;

CNN Architecture:

Let W_k be the filter weights and b_k the bias for the k-th filter.
- Convolution Layer (1D): $z_k^{(i)} = x^{(i)} * W_k + b_k$ where $*$ denotes 1D convolution;
- Activation (ReLU): $a_k^{(i)} = \max(0, z_k^{(i)})$ $a_k^{(i)}$
- Pooling Layer (Max Pooling): $p_k^{(i)} = \max(a_k^{(i)}[s{:}\ s + P])$, stride;
- Flatten and Dense Layer: $h = \sigma\ (W_{fp} + bf)$;
- Classification (Cross-Entropy):

Output Layer:
- Regression: $\hat{y}\ = Woh + bo$;
- Classification (Softmax): $\hat{y}\ c = \frac{e^{zc}}{\sum_{k=1}^{c} e^{zk}}$, $c = 1, \dots, C$;

Training (Gradient Descent):
- Update model parameters θ using optimizer (Adam or SGD): $\theta \leftarrow \theta - \eta \cdot \nabla\ \theta$;
- Repeat for E epochs or until convergence;

Testing:

3-fold cross validation;

Comparison: MLP, ALMLP, CNN, ALCNN

The 1D convolutional layer is well-suited for modeling temporal sensor data because it efficiently and straightforwardly preserves local temporal patterns. In comparison to LSTMs or hybrid CNN-LSTM architectures, which are adept at learning long-term dependencies but are computationally expensive and serially tractable, the 1D CNNs are trained with fewer parameters, allowing for faster training and the ability to train in parallel. They are also noise-resistant, a typical quality of sensor measurements. When it comes to detecting significant temporal changes, such as anomaly prediction or activity detection, their performance is exceptional. Furthermore, 1D CNNs align better with the structure of time-series data compared to their 2D counterparts, which often reshape labels to accommodate 1D data and reverse the process when returning predictions. This alignment is more suitable for real-life applications that have resource constraints, such as runtime, or require real-time capabilities. Finally, based on real-time sensor inputs, accuracy, precision, recall, and F1 score are used to evaluate the classification performance, enabling the model to identify the specific quality category.

2.3 Active Learning

The Active Learning (AL) algorithms of the IoT-ALMLP and IoT-ALCNN models provide a relearning mechanism. This mechanism enables the network to add new input variables and set new weight values [23]. The AL algorithm consists of three main components: the quality of learning detector (QLA), the quality of learning indexer (QLI), and the poor learning indicator (PLI). These components operate during both the training phase (pre-learning) and the testing phase (post-learning) to identify the need for the MLP in AL. In the pre-learning, they gather data from the input variables that reflect the quality of the training data and the training process. Then identify the abnormal data that affects the training progress or increases the BP error value [25]. In the post-learning phase, they are used to determine the need for active learning [26]. This decision implies adding new training data and performing new training [30], [31]. The three AL components of the ALCNN are functioning in the following sequence:

Pre-learning:

- The QLA detects the anomalies in the training samples that increase the error margin in the BP learning phase.
- The QLI identifies the anomalies in the training samples that increase the error margin in the BP learning phase.

Post-learning:

- The PLI determines the need for an AL based on the QLA, QLI, and the current performance of the MLP. It applies the entropy, weighted average entropy, and information gain to detect randomness or uncertainty in the data (i.e., the entropy equation).

$$E(X) = \sum_{i=1}^{n} P(x_i) \log 2 P(x_i) \tag{8}$$

where $E(X)$ is the entropy of X which is a discrete random variable that has a set of possible outcomes $x_i, \dots, x_n$ and each of which occurs according to a probability of $P(x_i), \dots, P(x_n)$.

Entropy is used to identify the most uncertain samples in active learning, where a higher entropy implies less certain predictions. Two general strategies are employed, namely, top-k selection, where samples with the k highest entropy are selected, and entropy thresholding, where samples with an entropy above a predefined level of uncertainty are selected. Top-k tends to converge more quickly, as it always chooses the most informative samples, whereas thresholding is more dynamic but can hinder learning when too few samples pass the test. Both can achieve high accuracy, but Top-k is more likely to be efficient in cases where budgets are fixed. Determining the relevance of the new training data from the given unknown and unlabeled data is outside the scope of this paper. Here, we assume that there is a significant change in the data sources during testing, which implies the need for active learning. For example, a pollution source or heavy rainfall incident near the river can dramatically change the water quality index (WQI).

2.4 River Water Quality Dataset

A real-time monitoring system utilizing IoT water quality sensors has been deployed at key locations along the Tigris River in Iraq. The combination of IoT sensing, knowledge transfer across rivers, and active learning provides an intelligent and scalable approach for environmental monitoring in resource-constrained regions. These sensors sit on and measure specific physicochemical parameters, including pH, turbidity, dissolved oxygen (DO), temperature, and electrical conductivity (EC), in a continuous manner. They then send the data wirelessly to a cloud server via protocols such as MQTT and LoRaWAN. Because manual sampling and expert labeling of the Tigris River are expensive and time-consuming, there is initially limited labeled data. However, there are also plenty of labeled datasets for similar river systems (e.g., the Euphrates River or Shatt al Arab), for which more research often exists. To pre-train a CNN for classifying water quality in an ordinal form, such as “Good”, “Moderate”, and “Poor”, external river water quality datasets are utilized [34], [35].

An AL loop is then incorporated within the system to fine-tune the model for the Tigris River's specific water profile [36]. This technique is expected to improve the efficiency of model learning, allowing for the timely classification of Tigris River water quality without expert intervention. The data supply process through the AL and processing is presented as follows:

- Initially, domain shifts were handled by applying z-score normalization to standardize spatial-temporal features, ensuring consistent distributions before transfer learning to the Tigris River data.
- The pre-trained CNN model is applied to incoming unlabeled data of the Tigris River received from that site.
- In the case of the active learner, samples that are high in uncertainty – those instances where the model has low confidence in its predictions (i.e., softmax output is close between multiple classes)- are identified.
- Some of these uncertain samples are then sent to human experts (with the help of occasional physical water sampling to check ground truth) to be manually labeled by an environmental scientist or technician, depending on this variable.
- We add the new data points to the training set, once labeled, then retrain the CNN to refine the architecture, in this case, to accommodate only Tigris River data.
- The loop keeps iterating, focusing on labeling only the most informative data to minimize the effort needed for annotation while achieving the highest adaptation and accuracy on the model

3. Results and Discussion

3.1 Experimental Design

This study aims to develop a valid and reliable real-time tool for classifying river water quality based on real-time data collected from a network of IoT-enabled water quality sensors in the Tigris River, Iraq. To facilitate the learning process, the initial training phase incorporates labeled datasets from other regions along the river, utilizing transfer learning principles to enhance learning. The model uses these external datasets as the foundational knowledge base and is then fine-tuned on highly impactful, small sections of the Tigris River. Preprocessing, normalization, and structuring of raw sensor data (pH, temperature, turbidity, dissolved oxygen, and electrical conductivity) into time-series samples is necessary to present the CNN with a data representation that can reveal temporal patterns and the corresponding water quality status classification (Good, Moderate, Poor).

A hybrid training using an AL loop is used to train the CNN. The loop allows the model to successively search for uncertain predictions in the unlabeled upcoming data from the Tigris River. The process of expert review or sampling of ground truth is used to prioritize these high-uncertainty samples for labeling. The model architecture is of convolutional layers of 1D with ReLU activations, Pooling layers, a flattening layer, and an output layer with softmax activation for multi-class classification. Categorical cross-entropy loss is used to train the network, and hyperparameters like learning rate, batch size, and number of filters are adjusted through experimentation. Early stopping criteria and dropout layers are included to ensure that convergence occurs efficiently and overfitting is avoided. Each input is normalized to lie in the range [0, 1] for reasons related to the Sigmoid activation function used in previous layers, ensuring consistent input scaling for CNN stability.

A prototype IoT-ALCNN system has been developed to support Iraqi environmental authorities in the successful monitoring and management of river water quality. The sensor data from Tigris River stations is continuously fed into the CNN model, which outputs predicted water quality classifications in real-time. The module of active learning operates in the background, allowing the model to evolve and learn from newly labeled critical samples. Users also see results represented graphically in the form of training accuracy, testing results, and error trends. The simulation results presented in this section are used for demonstration purposes and were conducted on one division of the Tigris River to illustrate how the model improves its prediction performance with each AL cycle.

3.2 Implementation

In this work, I implement the proposed IoT-based ALCNN model using a Python 3.10 environment where several scientific and machine learning libraries are utilized to facilitate data handling, model training, and model evaluation. We used the Keras API, which has a TensorFlow backend, as it is scalable, flexible, and easy to integrate into active learning workflows. For instance, data normalization, label encoding, and calculating performance metrics (i.e., accuracy, precision, recall) were performed in Scikit-learn. By extensively utilizing Pandas and NumPy, the sensor data collected through the IoT system deployed along the Tigris River was loaded, manipulated, and transformed. Matplotlib was used for visualizing and reporting comparative plots of training and testing accuracy and error trends during model evaluation. An active learning loop was custom-coded to iteratively query the most uncertain samples based on prediction entropy, allow manual labeling, and enable retraining of the model.

They were run on a standard personal computer equipped with Windows 10 (64-bit), an Intel Core i7-8750H CPU at 2.20 GHz, and 16 GB of RAM. The task of predicting water quality was framed as a multi-class classification problem; thus, the CNNs were trained accordingly using categorical cross-entropy loss and optimized through the Adam Optimizer. A fixed hyperparameter setting was employed, and the model was trained for 100 epochs with early stopping based on validation loss. For each experiment, we used an 80%-20% train-test split, with accuracy and F1-score as performance measures. Every 10 epochs, the active learning process was activated, which selected the top-k most uncertain predictions and prompted users to manually label them. The main experimental settings used in the study are summarized in the table below.

Table 1 *Experimental settings*

Parameter	Value/Setting
Programming Language	Python 3.10
Deep Learning Framework	Keras with TensorFlow backend
Optimizer	Adam
Loss Function	Categorical Cross-Entropy
Epochs	100 (with early stopping)
Batch Size	32
Train-Test Split	80% Training / 20% Testing
Active Learning Query	Entropy Sampling (Top-k = 10)
Hardware	Intel i7-8750H @ 2.20GHz, 16 GB RAM
Visualization Tools	Matplotlib
Data Libraries	Pandas, NumPy
ML Tools	Scikit-learn

3.3 Results and Analysis

This work proposes the ALCNN model and compares its performance with three other models to become ultimately: MLP, Active Learning enhanced MLP (ALMLP), and CNN, along with Active Learning enhanced CNN (ALCNN). These models are used to classify the river's water quality based on IoT sensor data collected from the Tigris River and nearby sources. Table 2 displays the water quality prediction results for the four models. Firstly, the standard MLP and CNN models were trained using the entire normalized dataset without implementing the iterative data selection strategy. These baseline models performed acceptably in terms of accuracy and metrics compared to more complex models (MLP = 93.56%, CNN = 94.88%). However, their higher loss values (0.1819 for MLP and 0.2054 for CNN) and weak recall scores may indicate some generalization issues and a slight degree of overfitting.

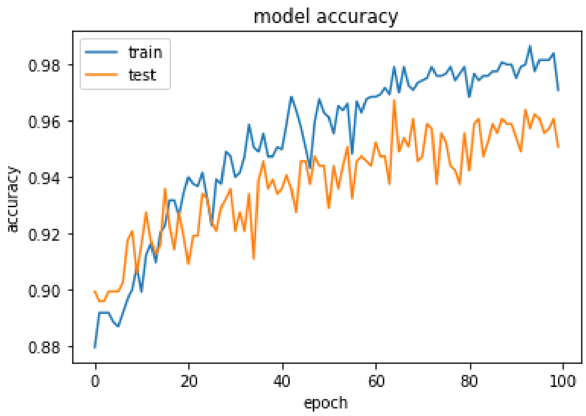
Both models saw significant improvements with the introduction of AL. The AL strategy's results, when paired with the MLP, confirmed that it enhances the model's ability to identify informative samples while learning more from fewer but more informative samples. Consequently, ALMLP achieved an accuracy of 97.60%, a considerably lower loss of 0.1726, and improved recall and F1-score (0.9899 and 0.9869, respectively), indicating a better mapping of the temporal water quality patterns. Conversely, the ALCNN outperformed all other models with the highest accuracy of 98.72% and the lowest loss of 0.1244, achieving almost perfect precision, recall, and F1 (0.9900, 0.9966, 0.9933, respectively). This highlights the advantage of combining active learning with deep CNN architectures, particularly for classification tasks involving dynamic environmental sensor data in real-time.

Table 2 *The water quality prediction results of the four models*

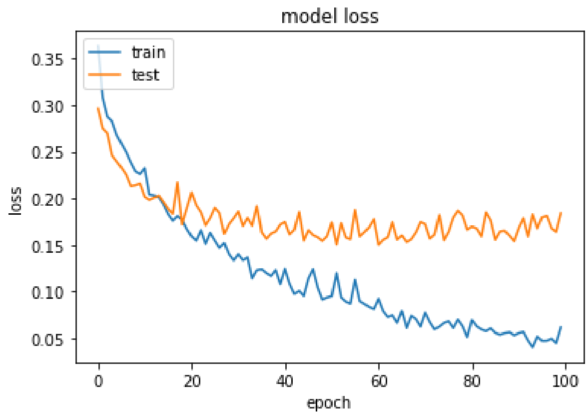
Model	Accuracy	Loss	Precision	Recall	F1-score
MLP	0.9356	0.1819	0.84128	0.7749	0.8034
ALMLP	0.9760	0.1726	0.9840	0.9899	0.9869
CNN	0.9488	0.2054	0.9852	0.8041	0.8411
ALCNN	0.9872	0.1244	0.9900	0.9966	0.9933

Furthermore, the plots for the training of ALMLP and ALCNN illustrate how closely the predicted classifications align with the true unseen (testing) sensor readings. The results for pH prediction indicate that the ALCNN model maintained a clear nonlinear progression with the original data throughout the entire out-of-sample testing phase. These findings demonstrate that active learning not only mitigated overfitting but also enhanced predictive performance for both models through better generalization. This led to a significant improvement in performance in data-poor or changing environments, such as those found in the Tigris River.

Figure 3 displays the accuracy and loss results of the MLP and ALMLP during the training and testing phases. It is clear to everyone who examines Figure 3, which contains four graphs illustrating the accuracies and losses for MLP and ALMLP, that ALMLP has significantly outperformed the standard MLP by a substantial margin. Furthermore, it shows that the regular MLP model was subject to a certain degree of overfitting, which Active MLP successfully rectified. This overfitting can be observed by examining Figure 3.

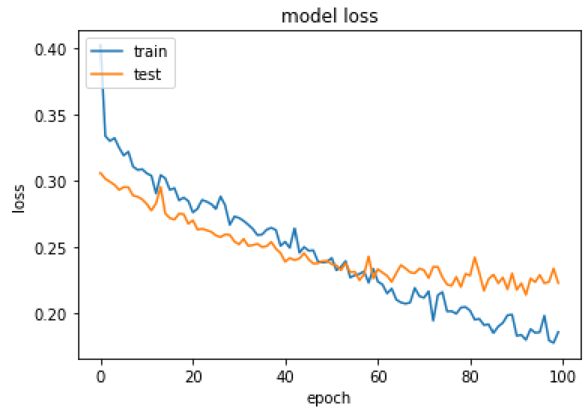


(a) MLP accuracy plot



(b) MLP loss plot

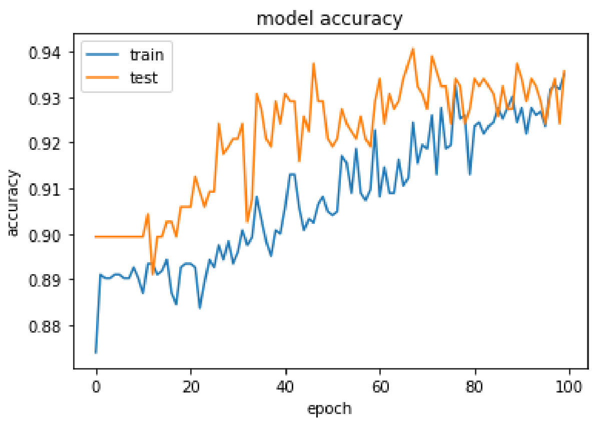
(c) ALMLP accuracy plot



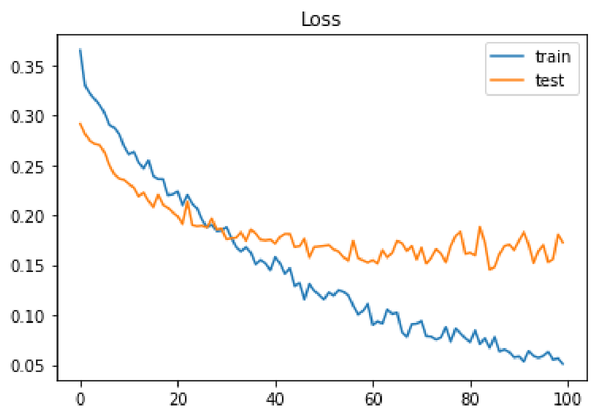
(d) ALMLP loss plot

Fig 3. *The accuracy and loss results of the MLP and ALMLP*

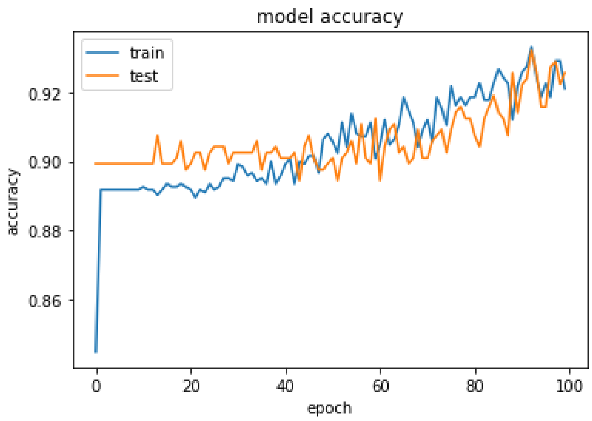
Figure 4, on the other hand, displays the performance of both the CNN and ALCNN models over one hundred epochs. It is evident that with AL, there is a substantial improvement compared to the traditional CNN, as the accuracy has risen to nearly 92% as early as the 10th epoch. However, the CNN model exhibited some overfitting due to the excessive number of layers, resulting in a slight data leak. Figure 4 illustrates the accuracy and loss outcomes of the CNN and ALCNN during the training and testing phases.



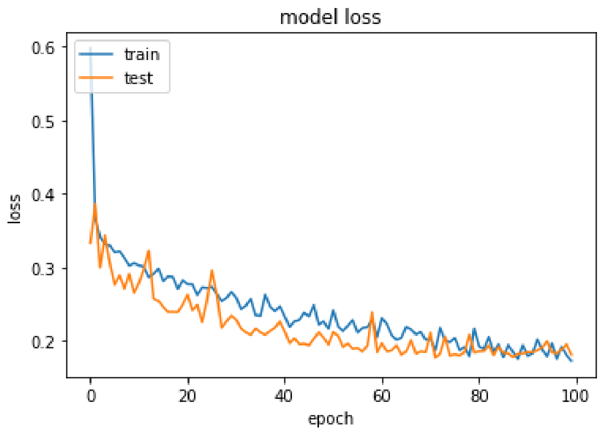
(a) CNN accuracy



(b) CNN loss



(c) ALCNN accuracy



(d) ALCNN loss

Fig. 4 *The accuracy and loss results of the CNN and ALCNN*

A significant evaluation of deep learning and active learning strategies was conducted using four models: MLP, ALMLP, CNN, and ALCNN, on an IoT-based river water quality dataset. The baseline MLP achieved 93.56% accuracy with a loss of 0.1819, lower recall, and F1 scores, which implied its weakness in handling temporal relations. The active learning integration actually improved MLP (ALMLP) with a noticeable boost in terms of accuracy (97.60%), precision (0.9840), and recall (0.9899). Equally, precision was increased to 0.9852 in the standard Case of CNN, but with a lower recall of 0.8041 and a slight increase in loss of 0.2054, indicating over-fitting. However, the AL-CNN achieved the best accuracy of 98.72%, with scores close to perfect in all metrics, indicating the effectiveness of combining the convolution principle for feature extraction with active sampling of high-uncertainty instances.

The key contributions of this study are (i) a novel integration of active learning with CNN for efficient and accurate water quality classification based on IoT sensor data, (ii) evaluation of the active learning based CNN model that shows it can outperform in terms of generalization even with less labeled data such that the proposed model can be efficient for resource constrained monitoring such as Tigris River case, and (iii) presentation of a reusable architecture that can easily be adapted to different rivers and environmental settings with minimal retraining. However, the study suffers from three limitations: (i) the initial labeled data required for the study of the external rivers may be bias prone; (ii) The active learning process relies on periodic human labeling and may not scale to real time; (iii) The model performance can vary based on sensor drift or harsh environmental noise and therefore it has to be retrained or adapted. Even with these challenges, the proposed framework offers a scalable and intelligent solution for managing smart water resources.

4. Conclusion

Water quality monitoring involves the systematic collection and analysis of physical, chemical, and biological parameters of water to assess its condition and suitability for various uses. It is of great significance because it helps protect ecosystems, public health, and makes it possible for us to manage water resources sustainably. In this paper, we introduce and integrate an IoT-based Active Learning Convolutional Neural Network (ALCNN) model for real-time classification of river water quality on the Tigris River in Iraq. IoT sensor data-based monitoring of essential water parameters served as input to the system, and MLP, CNN, and their variants were used for active learning classification. The experimental results confirmed that integrating active learning significantly improved model performance across all evaluation metrics, and that the ALCNN achieved the best accuracy of 98.72%, outperforming other baseline models in terms in terms of both generalization and efficiency. By employing an active learning framework, we can select data intelligently to reduce the requirement for a large amount of labeled data while still achieving robust classification performance. For future work, we will implement this model in a multi-site deployment across several rivers in Iraq, incorporating spatial and temporal correlations to enhance scalability. Additionally, we aim to include capabilities for edge computing to process and classify sensor data closer to the source, thereby reducing communication delay. Additionally, semi-supervised and self-supervised learning techniques can be employed further to reduce the system's dependency on manual labeling. Finally, future work will involve integrating climate and hydrological data to enhance accuracy during extreme weather events or seasonal variations. All this will give the system the status of a reliable tool for sustainable water resource management decisions.

Acknowledgement

The authors would like to acknowledge the constructive feedback received during the peer review process, which helped improve the quality of this paper.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

[1] Patients, S., et al. (2012). uMngeni Basin water quality trend analysis for river health and treatability fitness. Intech.

[2] Ali, A. A., Saadi, S. M., Mahmood, T. M., & Mostafa, S. A. (2022). A smart water grid network for water supply management systems. *Bulletin of Electrical Engineering and Informatics*, *11*(3), 1706-1714.

[3] Yang, Y., Xiong, Q., Wu, C., Zou, Q., Yu, Y., Yi, H., & Gao, M. (2021). A study on water quality prediction by a hybrid CNN-LSTM model with attention mechanism. *Environmental Science and Pollution Research*, *28*(39), 55129-55139.

[4] Nafi, S. N. M. M., Mustapha, A., Mostafa, S. A., Khaleefah, S. H., & Razali, M. N. (2019, September). Experimenting two machine learning methods in classifying river water quality. In *International Conference on Applied Computing to Support Industry: Innovation and Technology* (pp. 213-222). Cham: Springer International Publishing.

[5] Pu, F., Ding, C., Chao, Z., Yu, Y. and Xu, X., 2019. Water-quality classification of inland lakes using Landsat8 images by convolutional neural networks. Remote Sensing, 11(14), p.1674.

[6] Oga, T., Umeki, Y., Iwahashi, M. and Matsuda, Y., 2018, November. River water quality estimation based on convolutional neural network. In 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) (pp. 1305-1308). IEEE.

[7] Acharjee, U., Roy, S., & Das, A. K. (2021). An IoT-based smart water quality monitoring system using cloud computing. *Materials Today: Proceedings, 46*, 5796–5800. https://doi.org/10.1016/j.matpr.2020.11.946

[8] Singh, P., Dutta, S., & Yadav, S. (2023). Real-time water quality monitoring and analysis using IoT and machine learning. *Journal of Cleaner Production, 412*, 137276. https://doi.org/10.1016/j.jclepro.2023.137276

[9] Li, H., Wu, X., & Zhao, H. (2022). An IoT and deep learning-based framework for river water quality prediction. *Environmental Monitoring and Assessment, 194*, Article 395. https://doi.org/10.1007/s10661-022-10167-9

[10] Shen, F., Yu, H., Sakurai, K., & Hasegawa, O. (2011). An incremental online semi-supervised active learning algorithm based on self-organizing incremental neural network. Neural Computing and Applications, 20(7), 1061–1074.

[11] Hasenjäger, M., & Ritter, H. (2002). Active learning in neural networks. In New Learning Paradigms in Soft Computing (pp. 137–169). Physica-Verlag.

[12] Wang, W., Yang, T., Harris, W. H., & Gómez-Bombarelli, R. (2020). Active learning and neural network potentials accelerate molecular screening of ether-based solvate ionic liquids. Chemical Communications, 56(63), 8920–8923.

[13] Schran, C., Brezina, K., & Marsalek, O. (2020). Committee neural network potentials control generalization errors and enable active learning. The Journal of Chemical Physics, 153(10), 104105.

[14] Alhnaity, B., Kollias, S., Leontidis, G., Jiang, S., Schamp, B., & Pearson, S. (2021). An autoencoder wavelet-based deep neural network with attention mechanism for multi-step prediction of plant growth. Information Sciences, 560, 35–50.

[15] Shen, J., Ren, Y., Wan, J., & Lan, Y. (2021). Hard disk drive failure prediction for mobile edge computing based on an LSTM recurrent neural network. Mobile Information Systems, 2021, Article 8852361.

[16] Zhang, Y. G., et al. (2021). Application of an enhanced BP neural network model with water cycle algorithm on landslide prediction. Stochastic Environmental Research and Risk Assessment, 35(6), 1273–1291.

[17] Escrivá-Escrivá, G., Álvarez-Bel, C., Roldán-Blay, C., & Alcázar-Ortega, M. (2011). New artificial neural network prediction method for electrical consumption forecasting based on building end-uses. Energy and Buildings, 43, 3112–3119.

[18] Sun, R., Chen, Y., Dubey, A., & Pugliese, P. (2021). Hybrid electric buses fuel consumption prediction based on real-world driving data. Transportation Research Part D: Transport and Environment, 91, 102684.

[19] Gao, J., et al. (2021). STAN: Spatio-temporal attention network for pandemic prediction using real-world evidence. Journal of the American Medical Informatics Association, 28(4), 733–743.

[20] Ghazali, R., & Hussain, A. J. (2011). Dynamic ridge polynomial neural network: Forecasting the univariate non-stationary and stationary trading signals. Expert Systems with Applications, 38(3), 3765–3776.

[21] Connor, J. T., Martin, R. D., & Atlas, L. E. (1994). Recurrent neural networks and robust time series prediction. IEEE Transactions on Neural Networks, 5(2), 240–254.

[22] Jia, F., Liu, J., & Tai, X. C. (2021). A regularized convolutional neural network for semantic image segmentation. Analysis and Applications, 19(1), 147–165.

[23] Kong, Q., Cao, Y., Iqbal, T., Wang, Y., Wang, W., & Plumbley, M. D. (2020). PANNs: Large-scale pretrained audio neural networks for audio pattern recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 28(1), 2880–2894.

[24] Yadav, S., Rathod, R., Pawar, S. R., Pawar, V. S., & More, S. (2021). Application of deep convolutional neural network in medical image classification. In 2021 International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 120–129). IEEE.

[25] Chandrasekhar, A., & Suresh, K. (2021). TOuNN: Topology optimization using neural networks. Structural and Multidisciplinary Optimization, 63(3), 1135–1149.

[26] R., M. S., & T., V. (2021). Design of improved version of sigmoidal function with biases for classification task in ELM domain. Journal of Soft Computing Paradigm, 3(2), 70–82.

[27] Paluzo-Hidalgo, E., Gonzalez-Diaz, R., & Gutiérrez-Naranjo, M. A. (2020). Two-hidden-layer feed-forward networks are universal approximators: A constructive approach. Neural Networks, 131, 29–36.

[28] Haykin, S. (2004). Neural networks: A comprehensive foundation (2nd ed.). Prentice Hall.

[29] Shen, F., Yu, H., Sakurai, K., & Hasegawa, O. (2011). An incremental online semi-supervised active learning algorithm based on self-organizing incremental neural network. Neural Computing and Applications, 20(7), 1061–1074.

[30] Schran, C., Brezina, K., & Marsalek, O. (2020). Committee neural network potentials control generalization errors and enable active learning. The Journal of Chemical Physics, 153(10), 104105.

[31] Hasenjäger, M., & Ritter, H. (2002). Active learning in neural networks. In New Learning Paradigms in Soft Computing (pp. 137–169). Physica-Verlag.

[32] Wang, W., Yang, T., Harris, W. H., & Gómez-Bombarelli, R. (2020). Active learning and neural network potentials accelerate molecular screening of ether-based solvate ionic liquids. Chemical Communications, 56(63), 8920–8923.

[33] Ji, W., & Deng, S. (2021). Autonomous discovery of unknown reaction pathways from data by chemical reaction neural network. The Journal of Physical Chemistry A, 125(4), 1082–1092.

[34] Srivastava, P. K., et al. (2021). Sensitivity analysis of artificial neural network for chlorophyll prediction using hyperspectral data. Environment, Development and Sustainability, 23(4), 5504–5519.

[35] Turkoglu, M., Aslan, M., Ari, A., Alçin, Z. M., & Hanbay, D. (2021). A multi-division convolutional neural network-based plant identification system. PeerJ Computer Science, 7, e423.

[36] Chen, J., Zhou, M., Zhang, D., Huang, H., & Zhang, F. (2021). Quantification of water inflow in rock tunnel faces via convolutional neural network approach. Automation in Construction, 123, 103500.