Complex & Intelligent Systems (2021) 7:961–976
https://doi.org/10.1007/s40747-020-00259-9

ORIGINAL ARTICLE

Machine learning based soft sensor model for BOD estimation using intelligence at edge

Bhawani Shankar Pattnaik[1] · Arunima Sambhuta Pattanayak[2] · Siba Kumar Udgata[3] · Ajit Kumar Panda[4]

Received: 7 July 2020 / Accepted: 10 December 2020 / Published online: 7 January 2021

Abstract

Real-time water quality monitoring is a complex system as it involves many quality parameters to be monitored, the nature of these parameters, and non-linear interdependence between themselves. Intelligent algorithms crucial in building intelligent systems are good candidates for building a reliable and convenient monitoring system. To analyze water quality, we need to understand, model, and monitor the water pollution in real time using different online water quality sensors through an Internet of things framework. However, many water quality parameters cannot be easily measured online due to several reasons such as high-cost sensors, low sampling rate, multiple processing stages by few heterogeneous sensors, the requirement of frequent cleaning and calibration, and spatial and application dependency among different water bodies. A soft sensor is an efficient and convenient alternative approach for water quality monitoring. In this paper, we propose a machine learning-based soft sensor model to estimate biological oxygen demand (BOD), a time-consuming and challenging process to measure. We also propose a system architecture for implementing the soft sensor both on the cloud and edge layers, so that the edge device can make adaptive decisions in real time by monitoring the quality of water. A comparative study between the computational performance of edge and cloud nodes in terms of prediction accuracy, learning time, and decision time for different machine learning (ML) algorithms is also presented. This paper establishes that BOD soft sensors are efficient, less costly, and reasonably accurate with an example of a real-life application. Here, the IBK ML technique proves to be the most efficient in predicting BOD. The experimental setup uses 100 test readings of STP water samples to evaluate the performance of the IBK technique, and the statistical measures are reported as correlation coefficient = 0.9273, MAE = 0.082, RMSE = 0.1994, RAE = 17.20%, RRSE = 37.62%, and edge response time = 0.15 s only.

Keywords Intelligent system · Internet of things · Machine learning · Biological oxygen demand (BOD) · Soft sensor · Edge intelligence

Introduction

Intelligent system development for environment monitoring remains a challenge due to the complexity of a large number of parameters and the difficulty associated with their measurement. Water quality monitoring is one of the most critical aspects of environmental monitoring, apart from air quality monitoring. Access to safe drinking water is essential for health and also for good quality of life. It is not only important for human beings but equally important for the

✉ Siba Kumar Udgata
udgata@uohyd.ac.in

Bhawani Shankar Pattnaik
bhawani.pattnaik@gmail.com

Arunima Sambhuta Pattanayak
pattanayakarunima@gmail.com

Ajit Kumar Panda
akpanda62@hotmail.com

[1] Department of Computer Science and Engineering, National Institute of Science and Technology and Biju Patnaik University, Rourkela, Orissa, India

[2] Department of Computer Science and Engineering, National Institute of Science and Technology and Biju Patnaik University, Rourkela, Orissa, India

[3] School of Computer and Information Sciences, University of Hyderabad, Hyderabad, India

[4] Department of Electronics and Communication Engineering, National Institute of Science and Technology, Berhampur, Orissa, India

aquatic life and other living beings. Water Quality monitoring is the most important global risk interception [1], because it directly avoids public health-related issues. The World Health Organisation (WHO) has set up the guidelines of drinking water quality for several specific circumstances [2]. While performing the water treatment, the primary functions of a water treatment plant are to satisfy water demand, quality, and uniformity [3]. This needs qualitative and quantitative analysis of water in both the inlets and outlets. Artificial Intelligence (AI)/Machine Learning (ML) models have recently been widely used to predict the water quality parameters apart from many other significant applications [4–7]. These are pioneering works in this domain, but the authors mostly used cloud environment for analyzing the data and come out with the predictions. However, to ensure the safe supply of the drinking water, the quality of the water needs to be monitored in real time. Various low-cost systems exist for real-time monitoring of the water quality in the Internet of things (IoT) environment. The existing system can sense physical and chemical parameters of water through sensors, process it through the edge layer, and store the processed data in the cloud layer to monitor the water quality [8–10]. However, while analyzing and monitoring the water quality, the accuracy and reliability of the sensors are of primary concern [11]. The complex behavior of the measured parameters through each sensor is also a challenge in water quality management [12]. In water quality monitoring system, the important parameters which influence the quality of water are permeate-hydrogen concentration (pH), turbidity, dissolved oxygen (DO), bio-chemical oxygen demand (BOD), chemical oxygen demand (COD), total organic compound (TOC), total suspended solid (TSS), salinity, electrical conductivity, oxidation reduction potential (ORP), free chlorine, residual chlorine, heavy metals (iron, magnesium, cadmium, nickel, copper, mercury, and zinc), fluoride, arsenic, cyanide, nitrate, pathogens, and bacteria (*E. coli*). BOD is one of the vital parameters used to determine the quality of water [13]. There are a lot of low-cost sensors that exist in the market to measure water quality parameters. Still, some water parameters require a laboratory approach for analysis due to the lack of online real-time sensors. The reasons are high sensor cost, high sampling time, the requirement of frequent calibration and cleaning process, and regular sensor replacement due to a lesser lifetime of sensors [14]. For example, when we are focusing on the water quality parameter BOD, its sensor is of very high cost and also not quite reliable. The offline laboratory-based approach for measuring BOD is a time-consuming process. IoT-based Water quality monitoring setup always needs real-time sensing. Therefore, a significant delay in laboratory testing affects the performance of the system and defeats the basic objective of an IoT-based water quality monitoring system.

These problems, mentioned above, can be addressed using the soft sensor technique. The soft sensors approach is becoming a way to deal with these types of situations in the absence of specific sensors. Soft sensor is a virtual sensing technique that creates an inferential model to estimate different parameters of interest, based on other available measured parameters to provide feasible and economical alternatives to costly or impractical physical measurement sensors [3,15–18]. Soft sensor technique demands computation at the back end to perform its task. Therefore, it uses high computational server (cloud computing) for different applications [3,16–21]. Cloud computing provides a centralized pool of storage and computing resources. It has a global view of the network [22], but it is not suitable for applications that demand real-time response with low latency and high quality of service (QoS) [23]. However, almost all IoT application demands a response in real-time. Thus, there is a need for a modified computing environment for soft sensor to ensure real-time response of the IoT applications.

For computation, IoT applications adopt two techniques called cloud computing and edge computing. These two emerging paradigms can handle the massive amount of distributed data generated by IoT devices. However, these paradigms have their pros and cons. Cloud computing is not suitable for applications that demand real-time response with low latency and QoS, but it provides enough computational capabilities and a global storage concept [23]. On the other hand, edge computing is suitable for applications that need a real-time response, mobility support, and location awareness. Still, it does not have sufficient computing and storage resources [24,25]. Merging these two techniques (edge-cloud processing) together with effective machine learning algorithms can lead to an intelligent solution for enabling live data analytic in IoT applications [26]. In this work, we propose a BOD soft sensor model using edge-cloud processing of IoT framework, which is effective, scalable, and intelligent for real-time monitoring of water quality.

Motivation

Real-time water quality monitoring in the twenty-one century is complex and challenging because of the large number of chemicals and waste exhausted from the industries and commercial institutes; those make their way into the local water bodies and rivers. Although few commercially available sensors are available to measure the water quality, there are a few limitations in their real-time usage for all parameters due to high cost, different sampling rates, increased measurement time, frequent maintenance requirements, and environmental dependency. Soft sensor models are used in industrial processes for a long time as a replacement of hardware sensors in different deterministic environments. Using the IoT environment, soft sensing techniques, and edge intelligence concept, an attempt is made to develop a low-cost, robust IoT water quality monitoring system to address the current limitations.

Contributions

According to the existing literature, the complete soft sensor concept (both training and inference model) is complex and mostly implemented in the cloud architecture. However, the IoT application of a water quality monitoring system demands a real-time, uninterrupted, and reliable response. If the complete soft sensor concept runs on the cloud, the system cannot respond in real time due to in-network processing delay that includes propagation delay and transmission delay, connectivity loss, and network routing load. The present research proposes the distribution of soft sensor models in between cloud and edge to facilitate real-time action by the complete IoT setup. To respond to the environmental problem in real time, the prediction in the setup should be immediate. However, the training of the system can be performed periodically offline. To train the system, the BOD is calculated offline from the water samples using the standard laboratory approach. The main contributions of this paper are:

- Proposed a soft sensor model for BOD measurement which can act as an alternative to commercially available BOD sensor or as an additional method to validate the BOD sensor.
- Implemented the BOD soft sensor where the training algorithm can run on the cloud to train the system offline and periodically.
- Inference algorithm is executed on edge to make the edge intelligent and decide in real time.
- Determined efficient machine learning algorithm for soft sensor modeling using the experimental data before implementing the complete system.
- The developed model is validated with the data of sewage water treatment plant of the institute and the data collected from river "Ganga", an important river in India.

The rest of the paper is organized as follows; "Related works" describes the related work in water quality monitoring. "Problem statement and objective" focuses on the problem statement and objective of the paper. "Proposed system architecture for IoT water quality monitoring setup" describes the detailed system architecture for water quality monitoring, and "Experimental set-up and detailed steps for data collection" is focused on the experimental setup and the data collection steps. "Proposed methodology" proposes the methodology for the development of the BOD sensor for water quality monitoring, and "Experimental result and discussion" contains the analysis of results and discussions. Moreover, conclusions and future scope of this research are discussed in "Conclusions and future scope".

Related works

Several types of research have addressed the development of soft sensors with fairly large numbers of real-time applications [3,27–29]. Different approaches exist to develop a soft sensor like the model-based approach or empirical approach [30]. Model-based approaches describe the fundamental physical and chemical phenomena taking place in the process. It needs detailed knowledge about the system, as well as an accurate estimate for all the parameters involved, which is difficult in many modern contexts. On the other hand, the data-driven or empirical approach build predictive models based on historical data using different domains of data science [31]. Examples of methodology used in these approaches are principal components regression [32], artificial neural network [33], neuro-fuzzy systems [34], ML algorithms [35] like IBK, random forest, random tree, Kstar, REPTree, support vector machine (SVM) [21,36], and Gaussian processes [37,38]. The soft sensor concept is now widely being used in different application areas, such as biological wastewater treatment [19], bioprocess monitoring [29], bio-chemical systems [39], and many complex process predictions [16,30].

However, only considering water, Haimi et al. [19] have focused on data derived soft sensor applications in biological wastewater treatment and given a general guideline for soft sensor designing process. Huang et al. [20] have investigated the wastewater treatment using a genetic algorithm, a fuzzy neural system based soft sensor. The process can reliably estimate the nutrient dynamics of anoxic/oxic operations using online measured parameters like DO, pH, and ORP. A soft sensor method combined with Particle Least Square (PLS) and Neural Network, designed to realize the real-time online detection of the concentration of DO is given by Wei et al. [21]. Lamrini et al. [40] presented a soft sensor model using multi-layer perceptron (MLP), which can predict the coagulant dosage from raw water quality measurements from drinking water treatment plants. Wang et al. [41] developed a soft sensor model using radial basis function (RBF), to estimate the parameters of water, such as pH concentration, residual hydrogen concentration, and permeate gas flux. Petri et al. [42] presented a novel dynamic computational approach for predicting the turbidity of treated water using both linear and non-linear regression techniques.

Zhang et al. [43] considered the inflow (Q) as well as the COD, pH, TSS, and the total nitrogen (TN) to model a feed-forward three-layer multiple inputs and single output (MISO) neural network named as adaptive growing and pruning (AGP) network using back propagation (BP) algorithm. This soft sensor model was used to predict the BOD concentration. Luo [39] proposed an online soft BOD measurement method based on Laplacian Eigenmaps-relevance vector machine (LE-RVM). LE technique was used to process the pre-processed parameters and then is applied as the input of SVM to build the BOD soft sensor model. In this case, the prediction accuracy is not sufficient enough to be used in the real-time environment.

Support vector machine (SVM) [44,45] is a supervised learning technique used in a different field to model the soft sensor [46]. Extreme learning machine (ELM) is also a recent fast learning technique with a single hidden layer feed-forward neural network used for classification and regression purposes [47]. ELM technique is used to model the soft sensor for measuring DO concentration in the aquaculture field application [48]. Here, the authors also compared the ELM technique with backpropagation and SVM regression, and concluded that the prediction accuracy of ELM is high in this field compared to the other two approaches. Djerioui et al. [49], in their paper, developed a soft sensor to measure chlorine using a statistical learning technique to identify the water quality. They compared the ELM and SVM techniques where both methods require almost similar time for decision-making, but ELM takes less time for learning.

In all the above cases, the training (learning) and inference (prediction) algorithms for soft sensors run in the high computational cloud server. The server evaluates the data and train the system and make a decision whenever required. However, IoT-based solutions demand a real-time response. This is because sending data to the cloud for computation and decision-making is time-consuming due to communication overhead, network failure, and network latency. In the recent past, with the advancement in the IoT domain [50–53], the edge node is also becoming capable of performing a fairly large amount of computation. If the prediction takes place in edge by running an inference algorithm on the edge node itself, data do not need to make any round trip to the cloud, which reduces latency and leading to real-time, automated decision-making [54]. The learning (training) algorithms require heavy computation and, hence, are modeled to run in the cloud server periodically.

As a step forward, this work analyzes the water quality data collected and figures out the ML regression algorithm, which is best suited to implement the BOD soft sensor concept. This paper also performs a comparative study between cloud and edge training and prediction time required to run the suitable regression algorithms. Finally, this paper finalizes a system architecture for BOD sensors where the ML algorithm runs in a distributed manner to develop a BOD soft sensor that can make a prediction and decision in real time.

Problem statement and objective

Problem statement

The IoT-based water quality analysis and monitoring system aims to analyze different parameters present in water that influences the quality of the water like BOD, COD, DO, turbidity, ORP, pH, and temperature. Measuring BOD online through sensors is a challenge due to economical or technical limitations. However, to give a real-time response to computing quality of the water, it is essential to have real-time sensing. Therefore, this paper tries to model the BOD soft sensor, which can estimate the BOD value based on the parameters of other available sensor measurements. The soft sensor also provides feasible and economical alternatives to costly or impractical physical measurement sensors.

With the assumption that the oxygen consumption rate is directly proportional to the concentration of degradable organic matter remaining at any time, the expression for BOD, according to the first-order reaction kinetics can be represented as: [55]:

$$\mathrm{d}L_t/\mathrm{d}t = -KL_t, \tag{1}$$

where L_t is the amount of first-order BOD remaining in waste water at time t; K is the BOD reaction rate constant, time^{-1}.

Integrating both sides:

$$\int_0^t \mathrm{d}L_t = \int_0^t -KL_t \cdot \mathrm{d}t \tag{2}$$

$$\left[\log L_t\right]_0^1 = -K \cdot t, \tag{3}$$

where L_t/L_0=e^{-Kt} or 10^{-Kt}, where L_0 or BOD$_u$ at time t = 0, i.e., BOD initially present in the sample. The amount of BOD remaining at time 't' equals:

$$L_t = L_0(e^{-Kt}). \tag{4}$$

The amount of BOD that has been exerted (oxygen consumed) at any time t is given by:

$$\mathrm{BOD}_t = L_0 - L_t = L_0(1 - e^{-Kt}). \tag{5}$$

And the 5-day BOD is equal to:

$$\mathrm{BOD}_5 = L_0 - L_5 = L_0(1 - \mathrm{e}^{-K5}). \tag{6}$$

For polluted water and wastewater, a typical value of K (base e, 20 °C) is 0.23 per day and K (base 10, 20 °C) is

0.10 per day. The ultimate BOD (L_0) represents the maximum BOD exerted by the wastewater. Theoretically, it is challenging to achieve L_0, because it takes an infinite time. However, practically, the concentration of BOD can be expressed by measuring the concentration of degradable organic matter based on the total oxygen required to oxidize it. Therefore, using offline laboratory approach, the initial DO after collecting the sample for experiment needs to be checked (DO_1) and kept inside the darkroom at 20 °C and again checked for the DO value after 5 days (DO_5) and the BOD can be calculated after 5 days as:

$$\mathrm{BOD}_5 = (\mathrm{DO}_1 - \mathrm{DO}_5)/P, \tag{7}$$

where 'P' is a volumetric fraction of wastewater and expressed as volume of the sample divided by the volume of the container.

Due to the 5-day test period, BOD5 cannot be considered as a suitable parameter for a real-time water quality monitoring system.

As an alternative, a soft sensor (virtual sensors) technique can be preferred to predict the BOD5 of a water sample in real time from the BOD-dependent parameters with ML regression analysis. Regression analysis is a mathematical approach where it considers a dependent variable that is more difficult to determine, as a function of the independent variable(s) easy to measure directly. The relationships between the dependent variable with the independent variable(s) can be expressed as linear or non-linear functions [56]. Mathematically, regression uses a linear function to predict the dependent variable given as:

$$Y = \beta_0 + \beta_1 X + \epsilon, \tag{8}$$

where Y—dependent variable, the variable we predict; X—independent variable, the variable use to make a prediction; β_0—intercept, it is the prediction value when $X = 0$; β_1—slope, it represents the change in Y when X changes by 1 unit; ϵ—error, i.e., the difference between actual and predicted values.

Above is the equation of simple linear regression. In multiple regression, there are many independent variables (Xs). Error is a non-negligible part of the prediction-making process. The regression model can be evaluated using variety of performance matrices [57,58]. To benchmark performances of the proposed technique in this study, correlation coefficient, mean absolute error (MAE), root-mean-square error (RMSE), relative absolute error (RAE), and root relative squared error (RRSE) are used.

Correlation coefficient is used in statistics to measure the strong relationship between two variables. It returns the value between − 1 to + 1. The value is + 1 if there is a strong positive relationship between variables, and − 1 if there is a strong negative relationship between variables. Also, a result of zero indicates no relationship between variables.

Similarly, MAE is the average of the absolute error. It is the average difference between the actual and the predicted output value:

$$\mathrm{MAE} = 1/n \sum_{i=1}^{n} (y_i - \hat{y}_i), \tag{9}$$

where n is the number of samples used, y_i is the actual output, and $\hat{y}_i$ is the predicted output by the model.

RMSE is a measure to express the difference between the predicted value and actual value. It is the square root of the average of all squared error:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}}, \tag{10}$$

where n is the number of samples, y_i is the actual output, and $\hat{y}_i$ is the predicted output by the model:

$$\mathrm{RAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{\sum_{i=1}^{n} |\bar{y}_i - \hat{y}_i|}, \tag{11}$$

where n is the number of samples under taken, y_i is the actual output, $\hat{y}_i$ is the predicted output by the model, and $\bar{y}_i$ is the average value of all the actual output samples:

$$\mathrm{RRSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (\bar{y}_i - \hat{y}_i)^2}}, \tag{12}$$

where n is the number of samples of the model, y_i is the actual output, and $\hat{y}_i$ is the predicted output by the model. $\bar{y}_i$ is the average value of all the actual output samples.

To develop a soft sensor model using ML regression, an ML algorithm needs to select where MAE, RMSE, RAE, and RRSE errors are minimized, and correlation coefficient value is maximized, so that the model gives an accurate result. The developed soft sensor model needs to predict the value of water parameter, i.e., BOD in real time, to monitor and control the quality of the water. Therefore, a distributed IoT system architecture needs to recommend where the soft sensor computation is distributed between edge and cloud to make the system respond in real time.

Objective

The objective of this paper is to evaluate different ML techniques in terms of accuracy, time to predict, and then to design a suitable predictive BOD soft sensor model for IoT applications. The model computation is distributed between the edge and cloud layer to respond in real time, and to monitor and control the quality of the water before it can cause any substantial damage. Here, the BOD soft sensor model considers DO, pH, electrical conductivity, turbidity, ORP, and temperature as input, and predicts the BOD in real time using edge intelligence.

Proposed system architecture for IoT water quality monitoring setup

In the proposed architecture, the ML task of soft sensing technique (training and inference) is distributed between the edge layer and cloud layer of IoT to achieve a real-time response from the developed BOD soft sensor. Here, Fig. 1 represents the detailed system architecture for the soft sensor model as well as control in real time. To train the system, all online sensors, as well as laboratory sensors, present in the edge are used for measuring different parameters of the water resource. The measured parameters are pushed to the cloud server through the edge node. The server sitting in the cloud performs the data processing with recorded data, and generates a trained model file by running the ML training algorithm and sends back the model file to the edge node for future real-time decision-making. The comprehensive training approach is a periodic process with a specific interval of time, technically termed as incremental learning. The edge node fetches the data from the physical sensors. With the fetched data and trained model file (model file periodically sent through the cloud) by running an inference algorithm, the edge node evaluates the estimated value of BOD. The online physical sensor data, along with predictive BOD (soft sensor) value, decide the quality of the water.

According to the proposed architecture, the inference algorithm runs in edge and training algorithm runs in the cloud to achieve a real-time response. The proposed system analyzes and validates with different datasets in "Experimental result and discussion".

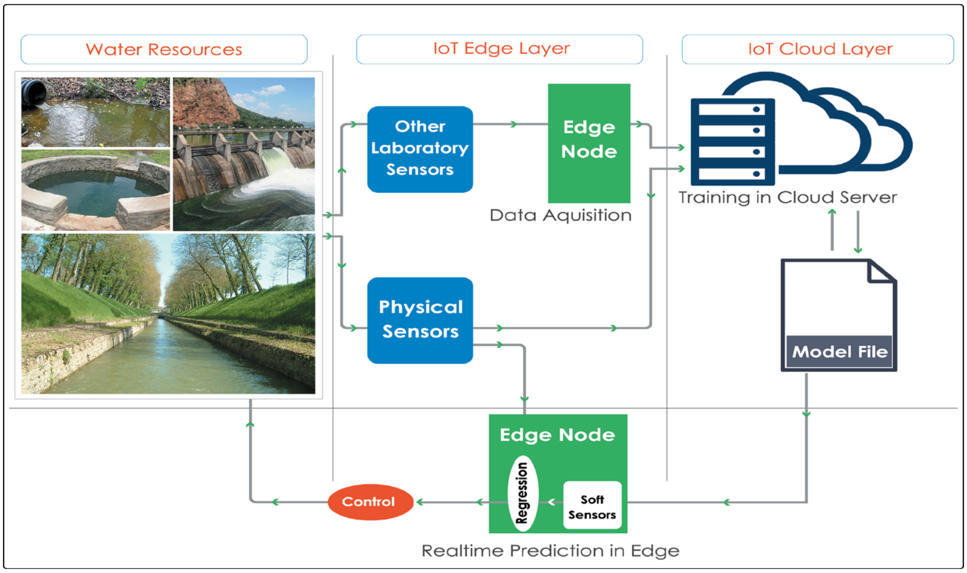

Fig. 1 Complete system architecture for water quality monitoring IoT setup

Experimental setup and detailed steps for data collection

Experimental setup 1

A solar-powered and self-navigated buoy with slots to install multiple sensors in a bay having access to water samples has been put inside the discharging tank of the sewage water reservoir of the sewage treatment plant (STP) at authors institute which is shown in Fig. 2. A controller act as an edge node helps to fetch the sensor data and push it to the cloud.

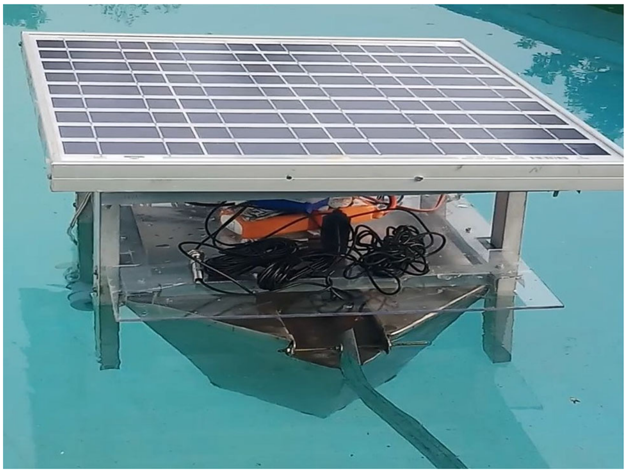

Fig. 2 Experimental setup 1 in the STP reservoir

The experimental buoy is fitted with the primary water quality sensors as DO, Temperature, pH, Electrical Conductivity, ORP, and turbidity. BOD sensor is not included in the sensor batch because of its high cost, and the value for BOD of the sample is extracted using an offline laboratory approach. Here, the sample from the STP is taken to the lab (10 ml of water in 300 ml container); the DO is checked (DO_1) and kept inside the darkroom at 20 °C and again checked for the DO value after 5 days (DO_5). The BOD is calculated after 5 days, as represented in Eq. 7.

The primary water quality parameters, including BOD, are recorded for 400 such samples following the above approach. The model file is generated after the recorded data are preprocessed and trained, which helps to predict the BOD value (real-time estimated) in the future from the real-time data retrieved from the sensor batch installed on the buoy.

Experimental setup 2

To validate the suggested architectural model more, in this setup, the real-time water quality monitoring data of river Ganga are recorded. The live data are provided by the Ministry of Environmental, Forest and climate change and Ministry of Water Resources, River Development, and Ganga Rejuvenation (available website link: http://122.166.234.42:8992/cr/). We collected the data published during the "Kumbh Mela" time, which happened from February to April 2019. Five hundred records containing the primary water quality parameters, namely DO, pH, Temp, TSS, and BOD of different locations of Ganga, are recorded from the website and pre-processed. With this pre-processed data, the model is trained, and a model file is generated for future samples collected by the edge node.

Proposed methodology

The experimental setup discussed in "Experimental set-up and detailed steps for data collection" contains several sensors and fetches the corresponding data. In water quality monitoring, if the number of sensors installed in the setup is more, it fetches a vast number of attributes. It may consume more time in the processing stage, and the cost of the complete setup increases, as well. Therefore, using input variable selection the Principal Component Analysis (PCA) technique is used in this paper to minimize the number of attributes and data from the massive number of samples collected from the water source, so that the data processing time reduces as well the entire system setup cost.

Although PCA [32] is the best known and most widely used dimension reduction technique, it can also be applied for selecting a subset of inputs based on their association with the output [15,59]. It helps to perform statistical data analysis, feature extraction from the given dataset, and identify the correlation between the attributes for data compression and data selection. In [15,60], the detail procedure for input selections using PCA is described.

If x_i is a set of dataset where $i = 1, 2, 3, 4 \ldots n$ and X is the observed data matrix. Therefore, the calculated mean value vector is:

$$\bar{X} = 1/m \sum_{i=0}^{n} x_i, \tag{13}$$

where m is the size of the data. The co-variance matrix is found out through the formula:

$$S = 1/m \sum_{i=0}^{n} (x_i - \bar{X})(x_i - \bar{X})^{\mathrm{T}}. \tag{14}$$

After applying the Eigen decomposition on the co-variance matrix:

$$S = E \wedge E^{-1}. \tag{15}$$

Here, E is the eigenvector matrix, and $\wedge$ is an eigenvalue's diagonal matrix. After this step, data points can calculate as well as the latent variables.

After pre-processing, the real-time data are ready to train and inferred using the ML algorithm to model Soft Sensors. Here, the authors have considered few popular algorithms [35] to train the pre-processing dataset like linear regression, multi-layer perceptron, SVM-SMO, Lazy-IBK, KStar, random forest, random tree, and REPTree.

Linear regression works by estimating coefficients for a line or hyperplane that best fits the training data. It is a simple regression algorithm where training can perform faster. It gives better performance if the output variable for the dataset is a linear combination of the inputs [35].

The multi-layer perceptron algorithms support both regression and classification problems. It is an algorithm derived by a model of biological neural networks in the brain where small processing units called neurons are organized into layers that, if configured well, are capable of approximating any function. In regression problems, the interest is to approximate a function that best fits the real value output [35].

Support vector machine models were developed for binary classification problems. As an extension, this technique has been made to support multi-class classification and regression problems. SMO is further used to solve quadratic problems. SVM automatically convert nominal values to numerical values. SVM-SMO regression is an optimization process that works by finding a line of best fit that minimizes the error of a cost function. It considered only those instances in the training dataset closest to the line with the minimum cost. These instances are called support vectors. In almost all problems of interest, a line cannot be drawn to fit the data best; therefore, a margin is added around the line to relax the constraint. Sometimes, a line with curves or even polygonal regions needs to be marked out; this can be done by projecting the data into a higher-dimensional space to draw the lines and make predictions. Different kernels are used to control the projection and flexibility in this technique [35].

Lazy-IBK is a k-nearest neighbor approach that marks an unclassified instance with the label of the majority of k-nearest neighbors. The distance between instances is measured using the Euclidean metric. If $k = 1$, the instance is assigned to the class of its closest neighbor in the training set [35]. This algorithm is quite useful in real-world applications where most of the data may not follow any distribution.

KStar is an instance-based classifier. It classifies the test instances based on the similarity function in training instances. It uses an entropy-based distance function to identify the similarity between the test set and training set instances [35].

Randomforest is an extension of bagging for decision trees that can use for classification or regression. Random forest is an improvement technique that disrupts the greedy splitting algorithm during tree creation, so that split points can only be selected from a random subset of the input attributes [35].

Random tree constructs a tree that considers K randomly chosen attributes at each node and performs no pruning. It also can allow estimation of class probabilities or target mean in case of regression based on a hold-out set using back-fitting [35].

REPTree is the fast decision tree learner. It builds a decision/regression tree using back-fitting. It only sorts values once for numeric attributes. Missing values are dealt with, splitting the corresponding instances into pieces [35].

These algorithms run in a distributed manner in the cloud as well as edge as proposed in “Proposed system architecture for IoT water quality monitoring setup” to develop soft sensor modeling in the proposed system, and the performance of the algorithm is discussed in “Experimental result and discussion”.

Experimental results and discussion

As the first step of the analysis, this paper performs PCA to determine which variables have the most substantial influence on the soft sensor and to reduce the number of sensors used in the real-time implementation. This process smooths the visualization of the dataset. The relevant datasets with the use of different ML methods predict the BOD concentration online to monitor the quality of the water. In Experiment 1, this paper is trying to monitor the water quality of the STP reservoir of the author’s institute. In Experiment 2, it validates the proposed model with the real-time water quality data of the Ganga River.

The hardware and software used to perform the experiments are set up as follows. The complete IoT setup contains a cloud server and an edge node having all communication setups and protocols to connect to the cloud server. The cloud layer contains a Virtual Machine (VM), which is working in the Linux environment with the specification of 4 cores 4GB of RAM and Ubuntu 16.04 Operating System. The edge node is a Raspberry Pi3 with Raspbian operating system, kernel version 4.14 (CPU configuration 1 GHz and 1 GB RAM) onboard connectivity with wireless LAN and Bluetooth. The complete setup program is written in Python, and the WEKA tool is used to cross-verify the results.

Experimental results and discussion for setup 1

Experiment 1 is conducted with an edge device (Pi3), which is connected with a set of water quality checking sensors like electrical conductivity, ORP, temperature (temp), DO, and turbidity. The cloud and edge layers communicate with each other through the REpresentational State Transfer (REST) Application Program Interface (API).

In this work, with the experimental setup, a total of 400 samples with eight attributes of water quality data are recorded. Table 1 represents the descriptive statistic of the recorded variables. For this experiment, BOD data are measured through an offline laboratory approach, and all other attributes are collected through physical sensors.

The input variables DO, temp, pH, Electrical Conductivity, ORP, turbidity, and output variable BOD are selected to analyze the water quality and to develop a BOD soft sensor model. Before applying the PCA to the dataset, the dataset is standardized and then feed-forward for the PCA approach. PCA application for the total dataset is depicted in Table 2, and the histogram representation is shown in Fig. 3.

From Table 2 and Fig. 3, it can be observed that the Eigenvalue decreases rapidly, so the variance proportion decreases simultaneously. The first five Principal Components (PC) of Table 2 represent 96.8% portion of the total variance, (PC1 represents 43.101%, PC2 represents 23.48%, PC3 represents 15.11%, PC4 represents 8.63%, and PC5 represents 6.478% portion of the total variance). Therefore, the six input variables are simplified into five variables, which are retrieved from the first five principal axes, and shown in Table 3, which represents the first five PCs and the correlation between the variables in the first five axes. Here, we can observe that there is no correlation among variables in PC1, PC2, and PC3. PC4 correlates with two variables turbidity, and ORP and PC5 correlate with the two variables DO and temp. Finally, to reduce the number of sensors and sensor costs, this paper considered four input variables only, i.e., turbidity, DO, pH, and temp, to predict BOD value.

Table 1 Descriptive statistic of the recorded variables of STP water samples

Input (variables)	Minimum	Maximum	Mean	Standard deviation
DO (mg/l)	1.022	6.348	3.043	0.971
Temp (°C)	21.9	29.03	26.097	1.836
PH	7.27	10.03	8.398	0.587
Electrical conductivity (µs/cm)	188.6	4426	1967.164	1048.539
ORP (V)	– 0.83	0.87	0.436	0.333
Turbidity (NTU)	4.8	88.61	25.10	15.09
BOD (mg/l)	30.203	81.295	50.413	11.095

Table 2 Principal component analysis of STP water sample (to select the suitable parameters)

	F1	F2	F3	F4	F5	F6
Eigen values	2.58607	1.40867	0.90666	0.51795	0.38866	0.19199
Proportion	0.43101	0.23478	0.15111	0.08633	0.06478	0.032
Cumulative proportion	0.43101	0.66579	0.8169	0.90323	0.968	1
Cumulative proportion (%)	43.101	66.579	81.96	90.323	96.8	100
Eigen vector	V1	V2	V3	V4	V5	V6
DO	– 0.1923	0.6229	0.505	0.164	0.5355	0.0799
Temp	– 0.3602	– 0.5935	– 0.0464	– 0.0972	0.5714	– 0.4241
pH	0.1748	– 0.4969	0.7717	0.1504	– 0.0866	0.3113
Electrical conductivity	– 0.5589	– 0.0634	– 0.237	0.18	0.1542	0.7558
ORP	– 05198	– 0.0026	0.1036	0.56	– 0.5098	– 0.3814
Turbidity	0.4692	– 0.0936	– 0.2836	0.7714	0.3092	0.0034

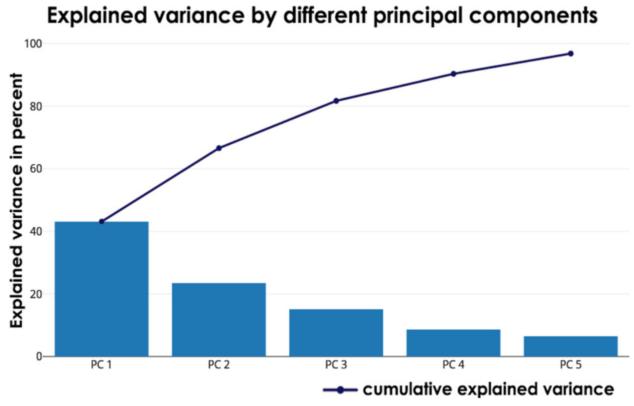

Fig. 3 Histogram of component Eigenvalue of STP water samples

The functional dependence among input and output parameters is represented in Eq. (16): [5].

$$\mathrm{BOD} = f(\mathrm{Turbidity}, \mathrm{DO}, \mathrm{pH}, \mathrm{Temp}). \quad (16)$$

After pre-processing using PCA, BOD is considering as an output of the soft sensor with the input of four sensors data, i.e., turbidity, DO, pH, and Temp. To model the soft sensor, different ML algorithms are used with the new features, and the best one, which gives a competitive performance in terms of accuracy, training, and prediction time, is selected.

The ML algorithm runs both on the cloud and edge layer of IoT to analyze which layer is giving better prediction accuracy and real-time response with less time. In terms of accuracy, both edge and cloud perform the same, which is observed and recorded in Table 4.

Table 4 shows the parameters like correlation coefficient, MAE, RMSE, RSE, RAE, and RRSE by performing tenfold cross-validation with all 400 data points. Here, the data points are divided into ten sets (called fold) out of which nine sets are used for training, and one set is used for testing. The cross-validation process is then repeated ten times, with each of the ten subsamples used exactly once for the validation of data. The ten results from the folds can then be averaged to produce a single estimation. In this process, all observations are used for both training and testing, and each observation thereafter is used for validation (testing) exactly once. From Table 4, it is observed that the Linear regression approach gives the worst result among all approaches, which implies that input variables are not linearly dependent on the output. Among all other non-linear algorithms, the IBK (*K* nearest-neighbor approach) gives better prediction accuracy or less error with *K* = 1. The comparative analysis between actual BOD and the predicted BOD using the IBK approach is performed, and

Table 3 Correlation matrix, Eigenvalue, variance proportion, and the cumulative variance proportion of the first four principal components (PC) of STP water

Input variables	PC1	PC2	PC3	PC4	PC5
DO	– 0.1923	0.6229	0.505	0.164	0.5355
Temp	– 0.3602	– 0.5935	– 0.0464	– 0.0972	0.5714
pH	0.1748	– 0.4969	0.7717	0.1504	– 0.0866
Electrical conductivity	– 0.5589	– 0.0634	– 0.237	0.18	0.1542
ORP	– 0.5198	– 0.0026	0.1036	0.56	– 0.5098
Turbidity	0.4692	– 0.0936	– 0.2836	0.7714	0.3092
Eigen values	2.58607	1.40867	0.90666	0.51795	0.38866
Variance proportion (%)	43.101	23.478	15.111	8.633	6.478
Cumulative variance proportion (%)	43.101	66.579	81.69	90.323	96.8

Table 4 Prediction accuracy in both cloud and edge for STP water data points (by tenfold cross-validation approach)

Training algorithm	Correlation coefficient	MAE	RMSE	RAE (%)	RRSE (%)
Linear regression	0.6622	0.3116	0.3921	68.4885	74.7244
Multi-layer perceptron (1 hidden layer)	0.8375	0.2385	0.2978	52.4275	56.7512
SVM-SMO (RBF kernel)	0.9996	0.004	0.0158	0.9104	3.0774
IBK (K = 1)	0.9998	0.0007	0.0117	0.1509	2.2282
Kstar	0.9983	0.0137	0.0308	3.0215	5.874
Random forest	0.9977	0.0124	0.0359	2.7199	6.8367
Random tree	0.997	0.0067	0.0407	1.4729	7.7665
REPTree	0.9858	0.023	0.0882	5.0488	16.8017

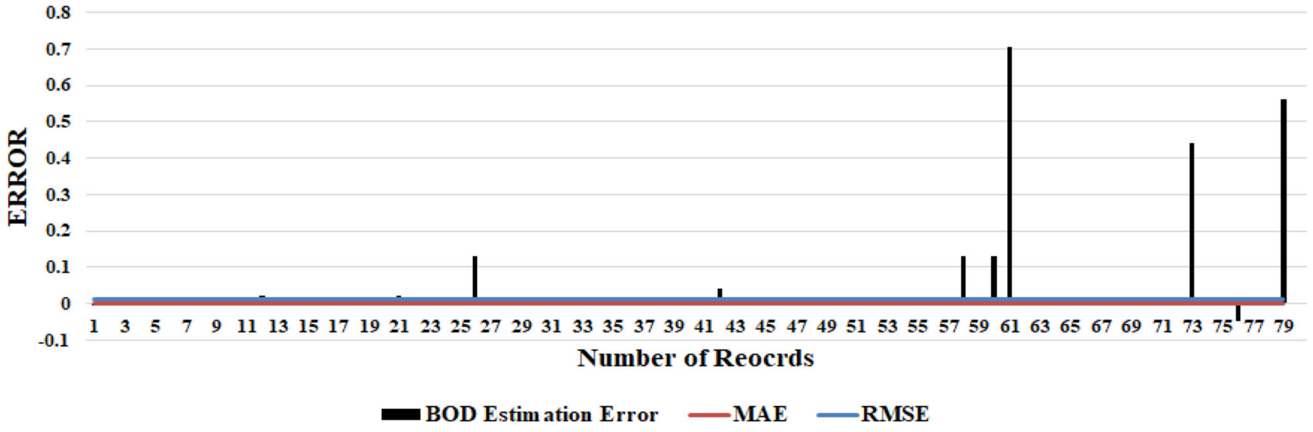

Fig. 4 BOD estimation error for experimental setup 1

the BOD estimation error, MAE, and RMSE are illustrated in Fig. 4.

In Fig. 4, the bar graph shows the testing result of the first twofold, i.e., BOD estimation errors of the first 80 samples through the IBK approach. The line graph shows the final MAE and RMSE of tenfold cross-validation approach.

To compare the efficiency of different ML algorithms in terms of training and prediction time, a total of 400 samples are taken for training, and one record is taken for prediction. The time required to train the dataset and predict the result in both the cloud and edge layer is recorded in Table 5.

From Table 5, it is observed that for all ML algorithms, for both training and prediction, the cloud takes less time than edge due to its high computational capability. Therefore, the cloud is preferred over the edge to run an ML algorithm. However, in the proposed IoT-based water quality monitoring system, BOD concentration is required to be measured in real time, and the cloud end prediction is always associated with latency, network overload, and network connection failure. If the prediction is performed in the cloud, the raw data file uploading time from edge to cloud for prediction and the decision result downloading time from cloud to edge

Table 5 Training and prediction time cloud and edge for STP water

ML algorithm	Cloud training time (in s)	Edge training time (in s)	Cloud prediction time (in s)	Edge prediction time (in s)
Linear regression	0.01	0.05	0.01	0.01
Multi-layer perceptron (1 hidden layer)	0.11	1.74	0.01	0.01
SVM-SMO	0.3	0.52	0.01	0.01
IBK	0.01	0.01	0.01	0.06
KStar	0.01	0.01	0.14	0.6
Random forest	0.05	0.76	0.01	0.03
Random tree	0.01	0.02	0.01	0.01
REPTree	0.01	0.09	0.01	0.01

Table 6 Total prediction time including dual communication time in cloud layer and real-time prediction in edge layer of STP datasets

Training algorithm	Prediction file upload time (in s)	Cloud prediction time (in s)	Prediction model file download time (in s)	Total time for cloud end prediction (in s)	Edge prediction time (in s)
Linear regression	0.28	0.01	0.277	0.567	0.01
Multi-layer perceptron (1 hidden layer)	0.28	0.01	0.277	0.567	0.01
SVM-SMO	0.28	0.01	0.277	0.567	0.01
IBK	0.28	0.01	0.277	0.567	0.06
KStar	0.28	0.14	0.277	0.697	0.6
Random forest	0.28	0.01	0.277	0.567	0.03
Random tree	0.28	0.01	0.277	0.567	0.01
REPTree	0.28	0.01	0.277	0.567	0.01

after prediction are substantial and cannot be ignored. Table 6 shows the comparative analysis of total prediction time, including communication time in the cloud and the prediction time required by the edge node. The same is also represented graphically in Fig. 5.

From Table 6 and Fig. 5, it is seen that the edge prediction time is approximately 39 times faster than cloud end prediction (with both upload and download communication time). Here, the dual communication time in the cloud is in a millisecond, which is tolerable for the intended application.

However, by supplementing the edge intelligence, the proposed system can predict the demanded parameter independently even when the edge node is effected due to internet connectivity loss or network congestion. Furthermore, in the case of a large number of sensing nodes, if the prediction algorithm runs on the cloud for each sensing node, then the server might be overloaded, which can be avoided by distributing the load locally using edge intelligence. Therefore, from the above analysis, it can be concluded that prediction should be made at the edge node in real time for STP water quality monitoring. Moreover, the training algorithm runs at the beginning and periodically at a specified interval of time to train and retrain the system and send the model file to the edge for real-time prediction by the edge node. The real-time BOD predicted using edge intelligence helps to trigger an alert in terms of alarm in response to an abort change in estimation.

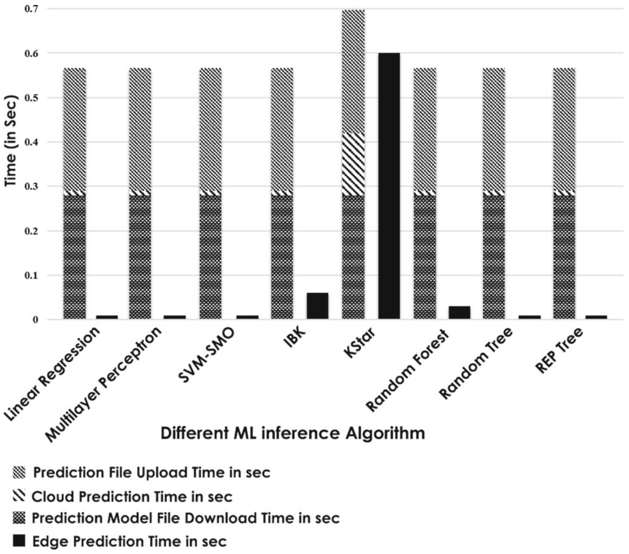

Fig. 5 Cloud vs edge prediction including dual communication time (in s) of STP datasets

Table 7 Descriptive statistic of the recorded Ganga data samples

Input (variables)	Minimum	Maximum	Mean	Standard deviation
DO (mg/l)	0.14	11.06	6.09	4.13
pH	5.61	8.91	7.519	0.667
Temp (°C)	9	25.89	18.385	2.666
Turbidity (NTU)	9	136.87	46.821	42.377
BOD (mg/l)	1.26	30.74	12.786	11.345

Table 8 Prediction accuracy in both cloud and edge for Ganga datasets (by tenfold cross-validation approach)

Training algorithm	Correlation coefficient	MAE	RMSE	RAE (%)	RRSE (%)
Linear regression	0.9769	1.3561	2.3962	14.3002	21.3489
Multi-layer perceptron	0.9838	1.0816	2.0174	11.4061	17.974
SVM-SMO	0.9863	0.678	1.8757	7.1499	16.7113
IBK	0.9966	0.143	0.9302	1.497	8.2877
Kstar	0.9929	0.2724	1.3355	2.8728	11.8985
Random forest	0.9905	0.3976	1.5427	4.1932	13.7446
Random tree	0.99	0.3203	1.5845	3.3771	14.1175
REPTree	0.9819	0.6188	2.1273	6.5251	18.9534

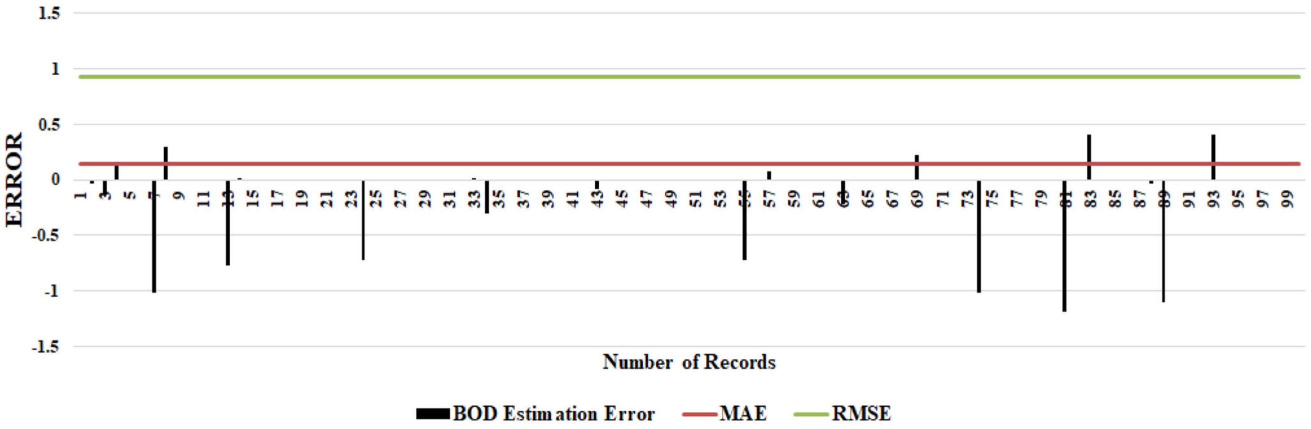

Fig. 6 BOD estimation error for experimental setup 2

Table 9 Training and prediction time cloud and edge for Ganga datasets

ML algorithm	Cloud training time (in s)	Edge training time (in s)	Cloud prediction time (in s)	Edge prediction time (in s)
Linear regression	0.01	0.01	0.01	0.01
Multi-layer perceptron (5 hidden layer)	0.12	1.47	0.01	0.01
SVM-SMO	0.1	1.65	0.01	0.01
IBK	0.01	0.01	0.01	0.04
KStar	0.01	0.01	0.26	0.76
Random forest	0.04	0.55	0.01	0.04
Random tree	0.01	0.01	0.01	0.01
REPTree	0.01	0.09	0.01	0.01

Table 10 Total prediction time including dual communication time in cloud layer and real-time prediction in edge layer for Ganga datasets

Training algorithm	Prediction file upload time (in s)	Cloud prediction time (in s)	Prediction model file download time (in s)	Total time for cloud end prediction (in s)	Edge prediction time (in s)
Linear regression	0.3	0.28	0.01	0.59	0.01
Multi-layer perceptron (1 hidden layer)	0.3	0.28	0.01	0.59	0.01
SVM-SMO	0.3	0.28	0.01	0.59	0.01
IBK	0.3	0.28	0.01	0.59	0.04
KStar	0.3	0.28	0.26	0.84	0.76
Random forest	0.3	0.28	0.01	0.59	0.04
Random tree	0.3	0.28	0.01	0.59	0.01
REPTree	0.3	0.28	0.01	0.59	0.01

Experimental results and discussion for setup 2

From the experiment setup 1, it is concluded that using four different physical sensors as input, we can predict the BOD in real time. In experiment 2, the same can be tested with real-time data of the different locations of the Ganga River, hosted by a Government hosted website. Here, we are considering only four sensors data as input data (DO, pH, Temp, and Turbidity) to minimize sensor cost and cross-validate the result of experiment 1. The collected raw data are stored in the cloud as well as an edge for further analysis. Here, a total of 500 samples are taken, and the statistics of the recorded variables are shown in Table 7.

Soft sensor took four variables datasets as input to predict the output variable BOD in real time for Ganga River. The dataset is standardized before it is feed as input for the soft sensor model. Here, Table 8 records the performance accuracy of the ML algorithm to select the appropriate algorithm for developing soft sensor modeling for Ganga Dataset.

Table 8 depicts the parameters like correlation coefficient, MAE, RMSE, RAE, and RRSE Error after performing tenfold cross-validation with all 500 data points. Here, cross-validation is performed similarly, as done in "Experimental Results and Discussions for setup-1". From the above table, it can observe that for Ganga data points also, IBK (with K = 1) gives better prediction accuracy compared to all other approaches. The comparative analysis between actual BOD and the predicted BOD in terms of BOD estimation error, MAE, and RMSE using IBK approach are represented in Fig. 6. The bar graph of Fig. 6 illustrated the BOD estimation error of first twofold (i.e., 100 records) and the line graphs represent overall MAE and RMSE after performing tenfold cross-validation approach with the dataset.

To compare the efficiency of different ML algorithms in terms of training and prediction time, a total of 500 samples are taken for training and a single record for prediction. Table 9 records the time required to train the dataset and to predict the result in the cloud and edge layer separately.

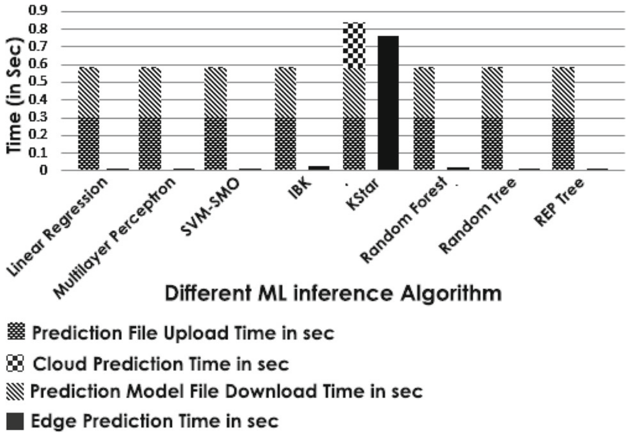

Fig. 7 Cloud vs. edge prediction including dual communication time for Ganga dataset of experiment 2

From Table 9, it is observed that for all ML algorithms, when the algorithm requires more computation, edge takes more time for training and prediction compared to the cloud. Table 10 shows the comparative analysis results of total prediction time, including communication time in the cloud and the prediction time at the edge node. The same is also represented graphically in Fig. 7.

From Table 10 and Fig. 7, it is observed that the edge prediction time is approximately 31 times faster than the cloud end prediction. From the above analysis, it is verified that the edge can predict the desired parameter independently in real time. With the consideration of the performance of soft sensor modeling for real-time Ganga River data and STP water: the IBK approach is selected to model the soft sensor in terms of prediction result accuracy and real-time response. Here, the training algorithm can run offline in the cloud, and the inference algorithm needs to run in edge to achieve a real-time response. As a validation of the complete setup, the

Table 11 Result analysis of real-time deployed system on STP water reservoir

Prediction algorithm	Correlation coefficient	MAE	RMSE	RAE (%)	RRSE (%)	Cloud prediction Time including dual communication time (in s)	Edge prediction time (in s)
IBK	0.9273	0.0812	0.1994	17.20	37.62	0.57	0.15

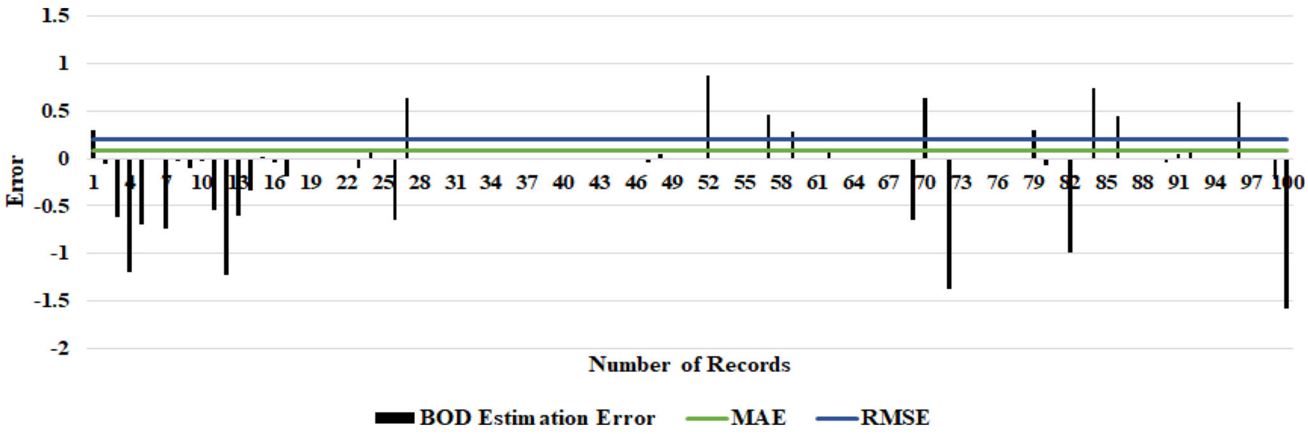

Fig. 8 BOD estimation error after the system deployment

system is deployed in the STP water outlet of the institute, and real-time BOD values are predicted and recorded. A total of 100 real-time values are recorded. The same water samples are collected and tested using the laboratory approach from which the actual result are obtained after 5 days, i.e., BOD5. Figure 8 represents the comparison between 100 real-time BOD predicted values by the system with the BOD value measured through the laboratory approach. The correlation coefficient, MAE, RMSE, RAE, and RRSE between the actual and predicted value are calculated and represented in Table 11.

From Table 11, it is observed that the calculated correlation coefficient between actual and predicted data using IBK with $K = 1$ is high, i.e., 0.9273. MAE and RMSE are low as per the requirement. The BOD estimation error is also represented through the bar graph, as shown in Fig. 8. The calculated MAE and RMSE are also shown in Fig. 8 through line graph. The average prediction time in the cloud, including uploading and downloading time, is quite higher than the average prediction time at the edge.

Conclusions and future scope

Soft sensors have a practical impact on the design and development of IoT-based water quality monitoring system. This paper presents a BOD soft sensor model that uses data-driven ML techniques to estimate the value of BOD in real time. A comparative study between different ML algorithms was carried out to select a suitable regression technique for the proposed system and it was found that the IBK algorithm is a good fit. A comparison between cloud level and edge level training and required prediction time is made to estimate the values in real time. It is found that estimation time for edge-based algorithms, which uses intelligence at the edge to predict the BOD values, is within a tolerable limit to make a decisions in comparison to the cloud based models. Finally, the real-time water quality monitoring system is designed using different physical hardware sensors and BOD soft sensor. The BOD soft sensor is modeled using the IBK approach with edge intelligence, which impacts directly on the cost of the system, and real-time response time. Based on this study, we can make decisions and take necessary actions as well as control the water quality monitoring system in real time. We also propose to develop soft sensor models for other water and air quality parameters in the future.

Acknowledgements This research is partially supported by WOS-A, ITRA, MEITY, Govt. of India and IUSSTF of Department of Science and Technology, Govt. of India

References

1. Li C, Zhang B, Luo P, Shi H, Li L, Gao Y, Lee CT, Zhang Z, Wu W-M (2019) Performance of a pilot-scale aquaponics system using hydroponics and immobilized biofilm treatment for water quality control. J Clean Prod 208:274–284
2. W H Organization (2017) Guidelines for drinking-water quality, 4th edition, incorporating the 1st addendum. [Online]. https://apps.who.int/iris/bitstream/handle/10665/254637/9789241549950-eng.pdf
3. Qiao J, Hu Z, Li W (2016) Soft measurement modelling based on chaos theory for biochemical oxygen demand (BOD). Water 8(12):581
4. Saberi-Movahed F, Najafzadeh M, Mehrpooya A (2020) Receiving more accurate predictions for longitudinal dispersion coefficients in water pipelines: training group method of data handling using extreme learning machine conceptions. Water Resour Manag 34(2):529–561
5. Najafzadeh M, Ghaemi A, Emamgholizadeh S (2019) Prediction of water quality parameters using evolutionary computing-based formulations. Int J Environ Sci Technol 16(10):6377–6396
6. Najafzadeh M, Ghaemi A (2019) Prediction of the five-day biochemical oxygen demand and chemical oxygen demand in natural streams using machine learning methods. Environ Monit Assess 191(6):380
7. Najafzadeh M, Tafarojnoruz A (2016) Evaluation of neuro-fuzzy GMDH-based particle swarm optimization to predict longitudinal dispersion coefficient in rivers. Environ Earth Sci 75(2):157
8. Chowdury MSU, Emran TB, Ghosh S, Pathak A, Alam MM, Absar N, Andersson K, Hossain MS (2019) Iot based real-time river water quality monitoring system. Procedia Comput Sci 155:161–168
9. Tripathy AK, Das TK, Chowdhary CL (2019) Monitoring quality of tap water in cities using IoT. In: Subramanian B, Chen SS, Reddy K (eds) Emerging technologies for agriculture and environment. Lecture notes on multidisciplinary industrial engineering. Springer, Singapore, pp 107–113. https://doi.org/10.1007/978-981-13-7968-0_8
10. Encinas C, Ruiz E, Cortez J, Espinoza A (2017) Design and implementation of a distributed IOT system for the monitoring of water quality in aquaculture. In: 2017 wireless telecommunications symposium (WTS). IEEE, Chicago, IL, 26–28 April 2017, pp 1–7
11. Banna MH, Najjaran H, Sadiq R, Imran SA, Rodriguez MJ, Hoorfar M (2014) Miniaturized water quality monitoring pH and conductivity sensors. Sens Actuators B Chem 193:434–441
12. Zhuiykov S (2012) Solid-state sensors monitoring parameters of water quality for the next generation of wireless sensor networks. Sens Actuators B Chem 161(1):1–20
13. Sagar S, Chavan R, Patil C, Shinde D, Kekane S (2015) Physico-chemical parameters for testing of water: a review. Int J Chem Stud 3(4):24–28
14. Murphy K, Heery B, Sullivan T, Zhang D, Paludetti L, Lau KT, Diamond D, Costa E, Regan F et al (2015) A low-cost autonomous optical sensor for water quality monitoring. Talanta 132:520–27
15. Curreri F, Fiumara G, Xibilia MG (2020) Input selection methods for soft sensor design: a survey. Future Internet 12(6):97
16. Fortuna L, Graziani S, Rizzo A, Xibilia MG (2007) Soft sensors for monitoring and control of industrial processes. Springer, London
17. Kadlec P, Gabrys B, Strandt S (2009) Data-driven soft sensors in the process industry. Comput Chem Eng 33(4):795–814
18. Pani AK, Vadlamudi VK, Mohanta HK (2013) Development and comparison of neural network based soft sensors for online estimation of cement clinker quality. ISA Trans 52(1):19–29
19. Haimi H, Mulas M, Corona F, Vahala R (2013) derived soft-sensors for biological wastewater treatment plants: an overview. Environ Model Softw 47:88–107
20. Huang M, Ma Y, Wan J, Chen X (2015) A sensor-software based on a genetic algorithm-based neural fuzzy system for modelling and simulating a waste water treatment process. Appl Soft Comput 27:1–10
21. Wei W, Changhui D, Xiangjun L, Jun G (2017) Soft-sensor software design of dissolved oxygen in aquaculture. Chin Autom Congr 2017:5413–17
22. Tang J, Quek TQ (2016) The role of cloud computing in content-centric mobile networking. IEEE Commun Mag 54(8):52–59
23. Corcoran P, Datta SK (2016) Mobile-edge computing and the internet of things for consumers: extending cloud computing and services to the edge of the network. IEEE Consum Electron Mag 5(4):73–74
24. Vallati C, Virdis A, Mingozzi E, Stea G (2016) Mobile-edge computing come home connecting things in future smart homes using lte device-to-device communications. IEEE Consum Electron Mag 5(4):77–83
25. Shi W, Cao J, Zhang Q, Li Y, Xu L (2016) Edge computing: vision and challenges. IEEE Internet Things J 3(5):637–646
26. Sharma SK, Wang X (2017) Live data analytics with collaborative edge and cloud processing in wireless iot networks. IEEE Access 5:4621–4635
27. Kadlec P, Gabrys B, Strandt S, Data-Kadlec P (2009) Data-driven soft sensors in the process industry. Comput Chem Eng 33(4):795–814
28. Sharma S, Tambe SS (2014) Soft-sensor development for biochemical systems using genetic programming. Biochem Eng J 85:89–100
29. Sagmeister P, Wechselberger P, Jazini M, Meitz A, Langemann T, Herwig C (2013) Soft sensor assisted dynamic bioprocess control: efficient tools for bioprocess development. Chem Eng Sci 96:190–98
30. Rato TJ, Reis MS (2018) Building optimal multiresolution soft sensors for continuous processes. Ind Eng Chem Res 57(30):9750–9765
31. Lu J, Liu A, Song Y, Zhang G (2020) Data-driven decision support under concept drift in streamed big data. Complex Intell Syst 6(1):157–163
32. Jolliffe IT, Cadima J (2016) Principal component analysis: a review and recent developments. Philos Trans R Soc A Math Phys Eng Sci 374(2065):20150202
33. Shang C, Yang F, Huang D, Lyu W (2014) Data-driven soft sensor development based on deep learning technique. J Process Control 24(3):223–233
34. Jang J-SR, Sun C-T, Mizutani E (1997) Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence [book review]. IEEE Trans Autom Control 42(10):1482–1484
35. Smusz S, Kurczab R, Bojarski AJ (2013) A multidimensional analysis of machine learning methods performance in the classification of bioactive compounds. Chemom Intell Lab Syst 128:89–100
36. Yan W, Shao H, Wang X (2004) Soft sensing modeling based on support vector machine and Bayesian model selection. Comput Chem Eng 28(8):1489–1498
37. Liu Y, Chen T, Chen J (2015) Auto-switch gaussian process regression-based probabilistic soft sensors for industrial multigrade processes with transitions. Ind Eng Chem Res 54(18):5037–5047
38. Chen J, Yu J, Zhang Y (2014) Multivariate video analysis and gaussian process regression model based soft sensor for online

estimation and prediction of nickel pellet size distributions. Comput Chem Eng 64:13–23

39. Luo L (2016) Biochemical oxygen demand soft measurement based on le-rvm. In: 2nd 2016 international conference on sustainable development (ICSD 2016). Atlantis Press, Xi'an, China, 2–4 December 2016, pp 164–167. https://doi.org/10.2991/icsd-16.2017.35
40. Lamrini B, Benhammou A, Le Lann M-V, Karama A (2005) A neural software sensor for online prediction of coagulant dosage in a drinking water treatment plant. Trans Inst Meas Control 27(3):195–213
41. Wang L, Shao C, Wang H, Wu H (2006) Radial basis function neural networks-based modeling of the membrane separation process: hydrogen recovery from refinery gases. J Nat Gas Chem 15(3):230–234
42. Juntunen P, Liukkonen M, Lehtola MJ, Hiltunen Y (2013) Dynamic soft sensors for detecting factors affecting turbidity in drinking water. J Hydroinform 15(2):416–426
43. Zhang M et al (2011) Research on dynamic feed-forward neural network structure based on growing and pruning methods. Zhineng Xitong Xuebao 6:101–06
44. Cristianini N, Shawe-Taylor J et al (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge
45. Cortes C, Vapnik V (1995) Support vector networks. Mach Learn 20(3):273–97
46. Yan W, Shao H, Wang X (2004) Soft sensing modeling based on support vector machine and Bayesian model selection. Comput Chem Eng 28(8):1489–98
47. Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489–501
48. Wang W, Deng C, Li X (2014) Soft sensing of dissolved oxygen in fishpond via extreme learning machine. In: Proceeding of the 11th world congress on intelligent control and automation, Shenyang. pp 3393–3395. https://doi.org/10.1109/WCICA.2014.7053278
49. Djerioui M, Bouamar M, Ladjal M, Zerguine A (2019) Chlorine soft sensor based on extreme learning machine for water quality monitoring. Arab J Sci Eng 44(3):2033–2044
50. Xia F, Yang LT, Wang L, Vinel A (2012) Internet of things. Int J Commun Syst 25(9):1101–02
51. Kopetz H (2011) Internet of things. In: Real-time systems. Springer, Boston, MA, pp 307–323
52. Atzori L, Iera A, Morabito G (2010) The internet of things: a survey. Comput Netw 54(15):2787–805
53. Gomathi P, Baskar S, Shakeel PM (2020) Concurrent service access and management framework for user-centric future internet of things in smart cities. Complex Intell Syst. https://doi.org/10.1007/s40747-020-00160-5
54. Ovenden J (2018) Edge computing and the future of machine learning | articles | big data. Innovation enterprise. DIALOG. https://channels.theinnovationenterprise.com/articles/why-machine-learning-needs-edge-computing. Accessed 23 Jan 2019
55. Ghangrekar M (2019) Bod model. IIT Kharagpur. DIALOG. https://scetcivil.weebly.com/uploads/5/3/9/5/5395830/m9_l12-water_quality_and_estimation_of_organic_content-contd.pdf. Accessed 24 Jan 2019
56. Draper NR, Smith H (1998) Applied regression analysis, vol 326. Wiley, New York
57. Najafzadeh M, Oliveto G (2020) Riprap incipient motion for over-topping flows with machine learning models. J Hydroinform 22(4):749–767
58. Sadeghi G, Najafzadeh M, Ameri M (2020) Thermal characteristics of evacuated tube solar collectors with coil inside: an experimental study and evolutionary algorithms. Renew Energy 151:575–588
59. Souza FA, Araújo R, Mendes J (2016) Review of soft sensor methods for regression applications. Chemom Intell Lab Syst 152:69–79
60. Vapnik V (2013) The nature of statistical learning theory. Springer science & business media, Berlin

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.