Dimensionality Reduction by Machine Learning for Cost-Effective Data Analysis

Abu Asaduzzaman[1], Md R Uddin[1], and Fadi N Sibai[1]

[1]Affiliation not available

April 17, 2024

Abstract

Processing large amount of data with many input features is always time consuming and expensive. In machine learning (ML), the number of input features play a crucial role in determining the performance of the ML models. Studies show that ML has potential for dimensionality reduction. This work proposes a methodology to reduce the number of input features using ML to facilitate cost-effective data analysis. Two different data sets for water quality prediction from Kaggle are used to run the ML models. First, we use Recursive Feature Elimination with Cross-Validation (RFECV), Permutation Importance (PI), and Random Forest (RF) models to find the impact of input features on predicting water quality. Second, we conduct experiments applying seven ML models: RF, Decision Tree (DT), Logistic Regression (LR), K-Nearest Neighbors (KNN), Gaussian Naïve Bayes (GNB), Support Vector Machine (SVM), and Deep Neural Network (DNN) to explore water quality using the original and reduced datasets. Third, we evaluate the impact of the optimized data features on computations and cost to test water quality. Experimental results show that reducing the number of features from nine to five for Dataset 1 helps reduce computations by up to 59% and cost up to 65%. Similarly, reducing the number of features from 20 to 16 for Dataset 2 helps reduce computations by up to 20% and cost up to 14%. This study may help mitigate the curse of dimensionality, via improving the performance of ML models by enhancing data generalization.

Dimensionality Reduction by Machine Learning for Cost-Effective Data Analysis

Abu Asaduzzaman, *Senior Member, IEEE*, Md R. Uddin, *Student Member, IEEE*, and Fadi N. Sibai

***Abstract*—Processing large amount of data with many input features is always time consuming and expensive. In machine learning (ML), the number of input features play a crucial role in determining the performance of the ML models. Studies show that ML has potential for dimensionality reduction. This work proposes a methodology to reduce the number of input features using ML to facilitate cost-effective data analysis. Two different data sets for water quality prediction from Kaggle are used to run the ML models. First, we use Recursive Feature Elimination with Cross-Validation (RFECV), Permutation Importance (PI), and Random Forest (RF) models to find the impact of input features on predicting water quality. Second, we conduct experiments applying seven ML models: RF, Decision Tree (DT), Logistic Regression (LR), K-Nearest Neighbors (KNN), Gaussian Naïve Bayes (GNB), Support Vector Machine (SVM), and Deep Neural Network (DNN) to explore water quality using the original and reduced datasets. Third, we evaluate the impact of the optimized data features on computations and cost to test water quality. Experimental results show that reducing the number of features from nine to five for Dataset 1 helps reduce computations by up to 59% and cost up to 65%. Similarly, reducing the number of features from 20 to 16 for Dataset 2 helps reduce computations by up to 20% and cost up to 14%. This study may help mitigate the curse of dimensionality, via improving the performance of ML models by enhancing data generalization.**

***Impact Statement*—Studies suggest that there are techniques to reduce the input features from data fields by removing the features that have less importance on data analysis. However, processing large amount of data with many input features is very expensive and time consuming due to the complexity to collect and test samples. The machine learning based methodology we introduce in this work helps overcome these issues and reduce cost to analyze quality of given samples. By reducing the number of features from nine to five for a Kaggle dataset for water quality prediction, the proposed methodology helps reduce the computations by up to 59% and the cost up to 65%. This study provides a scientific way to mitigate the curse of dimensionality, via improving the performance of machine learning models by enhancing data generalization and reducing overfitting.**

***Index Terms*—Machine learning, data analysis, input dataset, data feature pruning, dimensionality reduction, water quality prediction**

I. Introduction

MACHINE learning (ML) is a subset of artificial intelligence (AI), where data sets are used to train and test the ML models. Once the model is trained on the available data, it is ready to predict results based on the new input data [1-3]. Based on the execution time and accuracy of predictions, the model may be applied to real time operations. ML models help reduce the overall time to analyze large data and make decisions more accurately. Nowadays, ML models are being applied for large data analysis in many fields including water quality prediction, healthcare and medical diagnosis, financial trading, fraud detection, and natural language processing [4-10].

Large input data (both in data features, i.e., columns and data size, i.e., rows) helps the ML model learn the true pattern of the dataset and predict more accurately. However, as the input data increases, the number of computations and computational time increase [11-15]. Large input data consumes more resources including processing cores and memory. Preventive measures to reduce input features can lower the execution time and cost. ML inner working principle and high-performance computing techniques help reduce the execution time required by the large number of computations. Researchers have explored techniques for ML to reduce the input features from data fields [16-21], i.e., reduce the number of columns used for training and testing the ML algorithms. ML algorithms remove the features that have less importance on the model accuracy.

In our work, we use water quality prediction using ML as a test case. Water treatment plants process and filter water that is obtained mostly from ground water and surface water. Water usually holds sulfate, pH level, chloramines, dissolved solid (DS) particles, hardness, and other substances that are harmful to our health. The Environment Protection Agency (EPA) mentions a total of 101 contaminants that influence water quality in a report entitled “Parameters of Water Quality” [22]. Thereby, state and local agencies are made responsible for enforcing and keeping water quality standards. For doing so, water samples are collected and tested by water quality testing labs, which is time consuming and costly. Environmental Testing and Research (ETR) Laboratories, which conducts tests on water and environment for household or industry, offer three

A. Asaduzzaman is with Wichita State University, Wichita, KS 67260 USA (e-mail: abu.asaduzzaman@wichita.edu).

M. R. Uddin, is with Wichita State University, Wichita, KS 67260 USA (e-mail: mxuddin1l@shockers.wichita.edu).

F. N. Sibai is with the Gulf University for Science and Technology, Mishref, Kuwait (e-mail: sibai.f@gust.edu.kw).

This paragraph will include the Associate Editor who handled your paper.

packages of water quality testing in the U.S. [23]. The cost and time of each package are shown in Table I.

TABLE I
Costs to Test Water Quality

Category	Number of Substances	Time (days)	Cost ($)
Basic Water Test	53	2 to 4	139
Premium Water Test	113	2 to 4	229
Ultimate Water Test	249	4 to 7	699

Another water testing service provider, Precision Analytical Services, charges $475 for full private water quality test for school, college, industry, or household [24]. The water testing package may not have all the required contamination testing as required by users. Moreover, the user may not need all the tests. For such a situation, the user needs to buy the required test individually. It costs a maximum of $195 per substance testing. Moreover, it takes up to seven days to get the results.

We propose an ML-based dimensionality reduction method that has potential to reduce time and cost for applications including water quality prediction. The proposed method has two important working phases. In the first phase, an optimal (i.e., reduced) number of features are identified by prioritizing the importance of the input features. In the second phase, the performance (e.g., ML accuracy and water testing costs) of the reduced input features is obtained and evaluated.

I. Background Materials

A. Literature Review

Haq et al. investigate the effect of k-fold cross validation for Decision Tree (DT) and four Naïve Bayes (NB) models. It is found that DT achieves the best accuracy of 97.23% [25]. Naqeb et al. provide a comparison analysis on five different machine learning models to classify water potability [26]. Among DT, K-Nearest Neighbors (KNN), Random Forest (RF), Light Gradient Boosting Machine (GBM), and Support Vector Machine (SVM), RF shows the highest accuracy. Zhu et al. present a review on the current state of machine learning applications for water quality evaluation and challenges. ML is being applied to analyze marine environmental water, drinking water, ground water, and surface water [27].

Zhou et al. work on a water quality forecasting method under data-missing situation [28]. The results show that the deep learning with the post-processing approach suitably figures out the dependability between the model's output and observed water quality. Hasan and Azeez introduce a method on how Principal Component Analysis (PCA) can reduce the dimensionality of certain big data sets [29]. It improves interpretability without losing much information.

B. Techniques to Reduce Input Data Features

Existing popular techniques to reduce input data features include recursive feature elimination with cross validation (RFECV), permutation importance (PI), and RF. We apply these promising algorithms to prioritize the input features depending on the features' impact on ML performance.

1) RFECV

This is a widely used method for selecting optimized input features from the original dataset [30]. It is an iterative process that starts with all features in the dataset and progressively eliminates the least important ones until a desired number of features is reached. The algorithm evaluates the accuracy of the model with each subset of features using k-fold cross-validation, which involves splitting the dataset into training and validation sets, training the model on the training set, and evaluating its performance on the validation set. One of the key advantages of the RFECV algorithm is that it helps reduce overfitting and improve accuracy by identifying the most important features. It handles multi collinearity by removing one of the correlated features and thus improves performance. However, one drawback of this algorithm is that it does not provide any importance factor other than the optimized features. This method is used in our work to determine the optimized input features from the original dataset.

2) PI

This is another method for feature selection [31]. The algorithm measures the decrease in the model's performance when the values of a feature are randomly shuffled, which provides an estimate of the feature's importance. To determine the importance of each feature, the algorithm evaluates the model's performance on a validation set using cross-validation. For each feature, the algorithm randomly shuffles the values of that feature in the validation set, while keeping the other features constant. If shuffling a feature's values leaves the model error unchanged, then the feature is considered unimportant because the model ignored it for the prediction. The shuffled validation set is then passed through the model to get a new set of predictions. The permutation importance of a feature is calculated as the difference between the original model's performance on the unshuffled validation set and its performance on the shuffled validation set. One of the big advantages of permutation feature importance is that it does not require retraining the model unlike RFECV. Permutation importance is typically used as an analysis tool to identify useful features in big data. However, it does not provide the optimized number of features among all the features. In our work, this method is used to do the ranking of all available features based on their importance on accuracy.

3) RF

It is one of the supervised machine learning methods used for regression and classification [32]. The algorithm first performed bootstrapping which creates multiple subsets of the training data by randomly sampling. Then, a decision tree is trained for each subset using a subset of the features selected at random. The decision tree is trained using a greedy algorithm that selects the best feature to split the data based on the impurity score. The Gini impurity score or Mean Squared Error (MSE) is commonly used to measure misclassification probability. A feature importance score can also be calculated based on the reduction in impurity achieved. The more a feature reduces impurity, the more important it is considered. This process is repeated recursively until all samples are assigned to

a leaf node. Predicting output by majority voting helps this model to improve accuracy by reducing overfitting.

Existing popular ML models to classify a given dataset include RF (already discussed in Subsection II.B.3), DT, Logistic Regression (LR), KNN, Gaussian Naive Bayes (GNB), SVM, and Deep Neural Network (DNN). We use these models to classify a given dataset (with all original features and with the reduced features).

4) DT

This is one of the supervised ML models used for both classification and regression [33]. Its structure resembles a flowchart and is made up of internal nodes, branches, and leaf nodes. A leaf node shows the target value, a branch displays the decision rule, and an internal node represents the traits or attributes. Based on the values of several attributes, it creates a tree. It chooses the best feature for each node based on information gain, gini impurity, or entropy. To anticipate the target value, a homogeneous subset of data is to be created. The formula to calculate gini impurity is given in Equation (1).

$$Gini\ Impurity = 1 - \sum P_i^2 \quad (1)$$

Gini impurity indicates the probability of misclassifying. Here, P_i is the probability of each class 'i' in that node. So, a lower gini impurity score indicates a purer node. In that pure node, all samples belong to the same class. Entropy is used to measure the randomness in the information being processed. Mathematically entropy is represented by Equation (2).

$$E(S) = \Sigma_{i=1}^{c}(-p_i log_2 p_i) \quad (2)$$

Where, E(S) represents the current state and p_i is the probability of class i in a node of state S. Information gain is the difference between entropy before and after splitting the dataset. The formula is given in Equation (3).

$$Info\ Gain = Entropy(before) - \Sigma_{j=1}^{k} Entropy(j, after) \quad (3)$$

Where, k is the subset generated by the split and $(j, after)$ is the subset after the split. This model is simple to comprehend and apply. With numerical and categorical data, it performs well. However, overfitting occurs when the tree becomes too deep to detect noise in the data.

5) LR

This is another popular supervised ML model used for binary classification [34]. It is named from logistic function, also called sigmoid function. The logistic function converts any real number value to a value between 0 to 1. Taking this concept from statistics, logistic regression predicts the probability of certain classes from 0 to 1, as shown in Equation (4).

$$\mathrm{P}(y = 1|x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_n x_n)}} \quad (4)$$

Where, $P(y = 1|x)$ is the probability of positive class. β_0 is the constant and $\beta_1, \beta_2, \ldots \beta_n$ are the coefficients associated with the input features $x_1, x_2, \ldots x_n$, respectively. The value of coefficients is obtained using maximum likelihood estimation or gradient descent method. The likelihood is estimated by Equation (5).

$$\mathcal{L}(\beta_0, \beta_1, .., \beta_n) = \Pi_{i=1}^{N} \mathrm{P}(y_i|x_i) \quad (5)$$

Where, $\mathcal{L}(\beta_0, \beta_1, .., \beta_n)$ is the likelihood function. $P(y_i|x_i)$ is the probability of the positive class for the i-th data point. N is the number of data points in the dataset. It is calculated by Equation (6).

$$\mathrm{P}(y_i = 1|x_i) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_{i,1}+\beta_2 x_{i,2}+\ldots+\beta_n x_{i,n})}} \quad (6)$$

One benefit of LR is its simplicity, interpretability, and effectiveness. It is frequently used for tasks like disease outcome prediction, credit risk assessment, customer churn prediction, and sentiment analysis in industries, healthcare, finance, marketing, and social sciences.

6) KNN

This is also used for both classification and regression and one of the supervised ML models [35]. In this algorithm, the distance between the new data point and the training data point is measured. From all those distances, k number of neighbors are considered to vote for the class of the new data point. Usually, the k number is chosen as odd. To measure the distance, Euclidian distance, Manhattan distance, and other metrics are used. The mathematical expression of Euclidean distance for n dimensional space $(x1, x2, \ldots, xn)$ and $(y1, y2, \ldots, yn)$ is given in Equation (7).

$$d = \sqrt{((x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2)} \quad (7)$$

In the case of Manhattan distance, instead of square and root, it is using modulus operation as shown in Equation (8).

$$d = |x_1 - y_1| + |x_2 - y_2| + \ldots + |x_n - y_n| \quad (8)$$

Because of this, KNN is computationally expensive when performing prediction, especially when working with huge datasets. Furthermore, the choice of k and the distance metric have an impact on KNN.

7) GNB

This is a probabilistic ML model used for classification and regression task [36]. This algorithm assumes all the features are independent and distributed according to Gaussian distribution. During the training phase, it calculates the mean and variance of the Gaussian distribution for each feature in the training dataset, individually for each class label. It uses the Gaussian probability density function to determine the likelihood that a new data point's attributes belong to each class when predicting the class label for that data point as shown in Equation (9).

$$P(y|x1, x2, \ldots, xn) = P(y) * \Pi(P(xi|y)) \quad (9)$$

Where, $P(y|x1, x2, \ldots, xn)$ represents the new probability of class y with respect to the new feature values of $x1, x2, \ldots, xn$. The previous probability of class y is shown by $P(y)$. $P(xi|y)$ represents the conditional probability of feature xi with respect to class y. Π represents the product of all features. Computationally efficient with high dimensional data is one of the advantages of this model.

8) SVM

It is another supervised learning model suited for classification and regression analysis [37]. In this model, all the data points are plotted in n dimensional spaces where n is the number of features. The value of each feature is then tied up to a particular coordinate, making it easy to classify the data. The main objective is to separate hyperplanes that maximizes the margin between two classes. Using the equation of margin, the distance between a data point and hyperplane is computed using Equation (10).

$$distance = (w * x + b)/||w|| \quad (10)$$

Where, w is the weight vector, x represents the feature vector, b denotes the bias value, and $||w||$ is the Euclidian norm of the weight vector. The data points which are closest to the hyperplane are called support vectors. Using these support vectors, that hyperplane is determined. This is done by maximizing the margin between the support vectors from each class. SVM can handle high dimensional data with nonlinear relationship between features. It is also less prone to overfitting which makes this model more advantageous than other models. This SVM will be greatly related to our work due to the large number of features. To classify based on the available features, we expect that SVM will give high accuracy.

9) DNN

This is a special type of Artificial Neural Network (ANN), where the number of hidden layers is more than one [38]. DNN is a deep learning method that consists of three main parts: input layer, hidden layers, and output layer. The input layer depends on the input parameters and the hidden layers extract some of the most relevant patterns from the inputs and sends them to the next layer for further analysis. It accelerates and improves the efficiency of the model by recognizing the most essential information from the inputs and discarding the redundant information. Each neuron is connected to all the neurons of next layer through weighted connections. Each neuron of the next layer sum those weighted connections and applies activation function such as sigmoid, tanh, rectified linear unit (ReLU), etc. This process continues through each layer to get the output. In the output layer, it calculates gradient of the loss function with respect to the initial weights. Based on the error function, the weight is adjusted through backpropagation. One of the key benefits of DNN is that it can learn and model non-linear and complicated interactions. Since many of the real-life relationships between input and output are non-linear and complex, it has been well accepted as a methodology for classification of complex datasets such as environmental processes.

II. Methodology for Dimensionality Reduction

In this section, we describe the proposed methodology to reduce the input data fields without compromising the ML performance. We start with the raw input dataset that has all features. Figure 1 illustrates the major steps of the working principle of the proposed methodology. First, the raw data is cleaned by removing any missing values and "Not a Number."

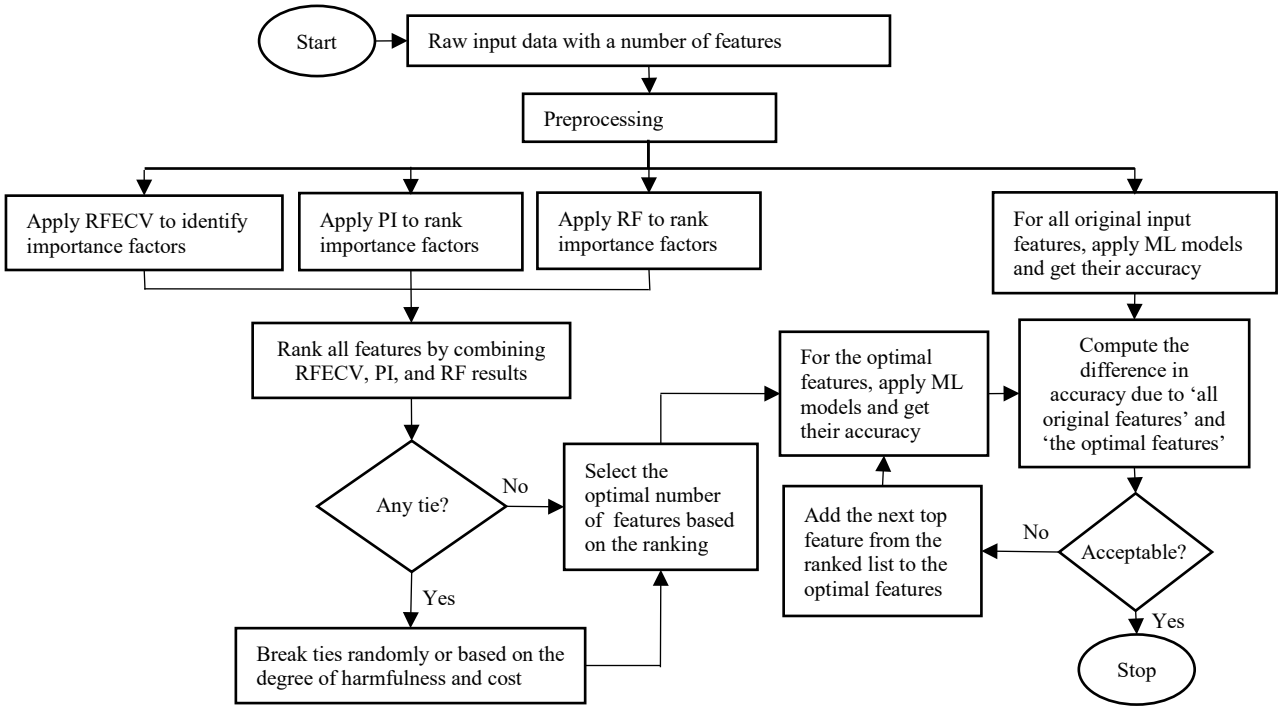

Figure 1. Workflow diagram of the proposed methodology to reduce the number of input data fields.

During the preprocessing step, no input features are removed. Initially, say, there are N number of input features. We apply RFECV and get the optimal number (say, x) of features. Where, $x \leq N$. In parallel, (i) we get the 'importance factor' for all input features using PI and RF and (ii) calculate performance of ML models (RF, LR, DT, KNN, GNB, SVM, and DNN) using

all input features. Considering the harmfulness and cost, we rank all input features by combining the outcomes of RFECV, PI, and RF. If there is any 'tie' among the outcomes, ties can be broken randomly or based on the degree of harmfulness and cost. Then we select the top x number of features from the ranked features. The dataset with those x features is called optimized dataset. Then, we calculate the performances of the ML models using the top x input features. If the difference between the accuracy due to all original input features and the accuracy due the top x input features is not acceptable, we understand that some impactful input features have been excluded. In that case, we add next important features from the ranked features and calculate performance of the ML models using more than the top x input features. Not shown in Figure 1, we may exclude the least impactful features from the top x features and calculate performance of the ML models using less than the top x input features. The process stops when the accuracy is acceptable, i.e., the optimal number of features is satisfactory.

III. Experimental Details

A. Dataset Used for ML Models

Two different publicly available sets of data from Kaggle for water quality prediction are used in this study [39]. Dataset 1 includes nine features and two labels. The features comprised of chloramines, conductivity, hardness, pH values, organic carbon, sulfate, total dissolved solids (TDS), trihalomethanes, and turbidity. In Dataset 1, there is a total of 3276 raw datapoints, among which 1278 datapoints are potable and 1998 datapoints are non-potable. The potable datapoints are labeled as 1 (one), whereas the non-potable datapoints are labeled as 0 (zero). The type of values of all features is floating point. In the raw data, 1265 data values are missing. The raw dataset is cleaned by removing the missing datapoints. The clean dataset of 2011 datapoints includes 811 potable datapoints and 1200 non-potable datapoints. To ensure that all features had equal importance during analysis, each feature is standardized by scaling them to have a mean of 0 (zero) and a standard deviation of 1 (one). In this dataset, there are no correlations between the features which makes this a good dataset to investigate dimensionality reduction.

Dataset 2 includes 20 input features and two labels. The input features are: aluminum, ammonia, arsenic, bacteria, barium, cadmium, chloramine, chromium, copper, fluoride, lead, mercury, nitrates, nitrites, perchlorate, radium, selenium, silver, uranium, and viruses. There is a total of 7996 raw datapoints among which 912 datapoints are potable and 7084 datapoints are non-potable. Like Dataset 1, potable and non-potable datapoints are labeled in the same way. The type of data values of all features is floating point. There is no missing value in the datapoints. To ensure that all features had equal importance during analysis, each feature was standardized by scaling them to have a mean of 0 (zero) and a standard deviation of 1 (one). Dataset 2 is also a good dataset to investigate dimensionality reduction because there is no correlation among the features.

B. Methods used for Dimensionality Reduction

RFECV, PI, and RF methods are used for dimensionality reduction. In this study, 10-fold stratified cross-validation is used in RFECV algorithm. Using PI algorithm, each feature is randomly permuted 25 times to obtain a more exact estimate of the feature's importance. The optimal RF model is achieved through a process of hyperparameter tuning using randomized search cross validation technique. RFECV provides an optimal number of features with importance factor value. It does not provide any importance factor value for other than listed optimal number of features. PI and RF provide importance factor value for all the features. The hyperparameters tested include the number of decision trees, ranging from 50 to 1000, and the maximum depth of each tree, from 1 to 50. To reduce computational costs, a 5-fold stratified cross validation is used. The model is then trained on 4-fold and confirmed on the remaining fold. This process is repeated five times such that each fold is used exactly once.

C. Effectiveness of the Dimensionality Reduction Methods

To validate the effectiveness of the proposed dimensionality reduction, we employ seven ML models, namely, RF, DT, LR, KNN, GNB, SVM, and DNN.

The total number of computations for a RF model can be estimated based on the number of trees t, number of nodes in a tree n, and number of features f as shown in Equation (11).

$$Total\_Comp = O(t * n * f) \tag{11}$$

The DT model is trained with tree depth of 4 (four). Total number of nodes can be obtained in terms of depth ($2^{depth} - 1$). The simplified computational complexity for decision tree model is given in Equation (12) [40].

$$Total\_Comp = O(N * M * \log(N)) \tag{12}$$

Where, N represents the number of data samples and M is the number of features.

The LR model is trained to converge and predict with 120 iterations. Total number of computations for each iteration in logistic regression model is calculated by Equation (13).

$$T\_C = O(M)+O(N*M)+O(N)+O(N*M)+O(M) \tag{13}$$

Where, $O(M)$ is the computational complexity for initializing and updating weights and bias; $O(N * M)$ represents the computational complexity for forward pass and backpropagation; and $O(N)$ is the computational complexity for the loss function.

The KNN model is configured and trained with a 20 leaf size that controls the node. The number of neighbors for this model is set to 9. The total computation for KNN is calculated by the formula shown in Equation (14) for each leaf [41].

$$Total\_Comp = O(N * M * K) \tag{14}$$

Where, N and M are as above, and K represents the number of neighbors.

The GNB model is trained using the GaussianNB() function for both datasets. The Pred() function is used to predict on the test data. The total computation for GNB is also calculated by the formula in Equation (14).

For SVM, the radial basis function kernel is used, and random state is set to 42. The computational complexity of SVM is represented by Equation (15) [42].

$$Total\_Comp = O(N^2 * M) \quad (15)$$

Similarly, the DNN model is trained with one input layer, two hidden layers, and one output layer. Each hidden layer consists of 20 neurons, where the number of input neurons is equal to the number of input features. The output layer consists of only one neuron. Activation function is ReLU. The total number of computations for one input layer, one hidden layer, and one output layer can be calculated by using Equation (16) [43].

$$T\_C = O((2 * M * p) + (2 * p * q) + (2 * M * p)) \quad (16)$$

Where, p represents the number of neurons in hidden layer and q represents the number of neurons in output layer. Each neuron does one multiplication and one addition. Therefore, $(2 * M * p)$ computations are performed at each hidden layer and backpropagation, and $(2 * p * p)$ computations are performed at the output layer.

D. Data Used for Water Quality Testing Cost Analysis

To conduct a cost analysis of and understand the financial aspects associated with water quality testing, various types of data must be collected and analyzed. In our work, we use publicly available data from popular websites [44-49] that show costs associate with drinking water testing. For example, Community Science Institute [44] provides price for each water substance test. AMTEST Lab [45] provides cost per drinking water analyte. Costs for quality analyses for household water, livestock water, irrigation water, etc. are taken from Oklahoma State University Laboratories Services and Price List [49].

IV. Results and Discussion

In this section, we present experimental results using two Kaggle datasets for water quality testing. First, we discuss how the input features are optimized based on the importance of the features by using RFECV, PI, and RF methods. Then, we examine the effectiveness of the dimensionality reduction method by employing seven different popular ML models. Finally, we discuss the computation and cost saving due to the proposed method.

A. Optimizing Input Features

The original Dataset 1 for water quality prediction has nine features (chloramines, conductivity, hardness, pH value, organic carbon, sulfate, TDS, trihalomethanes, and turbidity). RFECV with 10-fold stratified cross validation suggests that only five features (sulfate, pH values, chloramines, TDS, and hardness) are impactful for predicting water quality. PI and RF separately rank all nine original features. Both PI and RF provide similar results; the top five features are the same for RFECV, PI, and RF; although in a different order. The aggregated importance factor for Dataset 1 is shown in Table II. Here the optimized number of features is five and there is no tie in the ranking. Hence, the optimized dataset consists of sulfate, pH values, chloramines, TDS, and hardness.

TABLE II
Dataset 1 Substances and Ranking

Substances	RFECV	PI	RF	Total	Ranking
Chloramines	0.194	0.047	0.121	0.362	*3*
Conductivity	NA	0.013	0.092	0.105	7
Hardness	0.189	0.038	0.114	0.341	*5*
pH Value	0.217	0.120	0.141	0.478	*2*
Organic Carbon	NA	0.011	0.095	0.106	6
Sulfate	0.218	0.120	0.142	0.480	*1*
TDS	0.181	0.061	0.113	0.355	*4*
Trihalomethanes	NA	0.007	0.091	0.098	8
Turbidity	NA	0.004	0.091	0.095	9

The original Dataset 2 for water quality prediction has 20 features (aluminum, ammonia, arsenic, bacteria, barium, cadmium, chloramine, chromium, copper, fluoride, lead, mercury, nitrates, nitrites, perchlorate, radium, selenium, silver, uranium, and viruses). RFECV selects 16 features (aluminum, ammonia, arsenic, barium, cadmium, chloramine, chromium, bacteria, viruses, lead, nitrates, nitrites, perchlorate, radium, silver, and uranium) as the optimized number of features. Similarly, using PI and RF, we obtain the importance factor of each substance (i.e., feature)---copper, fluoride, selenium, and mercury become the least important features. There is one tie for Rank 8 (between nitrates and uranium). The tie is broken by ranking those features randomly (new rank of uranium is 9). Table III shows the aggregated importance factor of the optimized dataset: aluminum, ammonia, arsenic, bacteria, barium, cadmium, chloramine, chromium, lead, nitrates, nitrites, perchlorate, radium, silver, uranium, and viruses.

TABLE III
Dataset 2 Substances and Ranking

Substances	RFECV	PI	RF	Total	Ranking
Aluminum	0.211	0.075	0.203	0.489	1
Ammonia	0.054	0.012	0.047	0.113	6
Arsenic	0.066	0.014	0.065	0.145	4
Bacteria	0.031	0.011	0.028	0.070	14
Barium	0.031	0.002	0.029	0.062	15
Cadmium	0.119	0.058	0.115	0.292	2
Chloramine	0.051	0.006	0.047	0.104	7
Chromium	0.037	0.001	0.033	0.071	13
Copper	NA	0.007	0.024	0.031	*17*
Fluoride	NA	0.001	0.023	0.024	*18*
Lead	0.029	0.003	0.025	0.057	16
Mercury	NA	0.001	0.016	0.017	*20*
Nitrates	0.043	0.008	0.037	0.088	8

Substances	RFECV	PI	RF	Total	Ranking
Nitrites	0.037	0.012	0.033	0.082	10
Perchlorate	0.121	0.042	0.108	0.271	3
Radium	0.038	0.004	0.033	0.075	12
Selenium	NA	0.000	0.018	0.018	*19*
Silver	0.059	0.016	0.051	0.126	5
Uranium	0.039	0.015	0.034	0.088	8 / 9
Viruses	0.035	0.009	0.033	0.077	11

B. *Effectiveness of the Dimensionality Reduction*

Using RFECV, PI and RF, we select the optimized list of input features for Datasets 1 and 2. To evaluate the effectiveness of the optimized features, we obtain the accuracy due to seven different ML models (RF, DT, LR, KNN, GNB, SVM, and DNN) for four different situations: with all original features, one more than the optimized number of features, optimized features, one less than the optimized number of features. Figure 2 shows the accuracy for Dataset 1. Where, RF shows the highest accuracy of 95% with the optimized dataset. Figure 3 shows the ML accuracy due to Dataset 2. Where, the accuracy increases about 1% due to the dimensionality reduction.

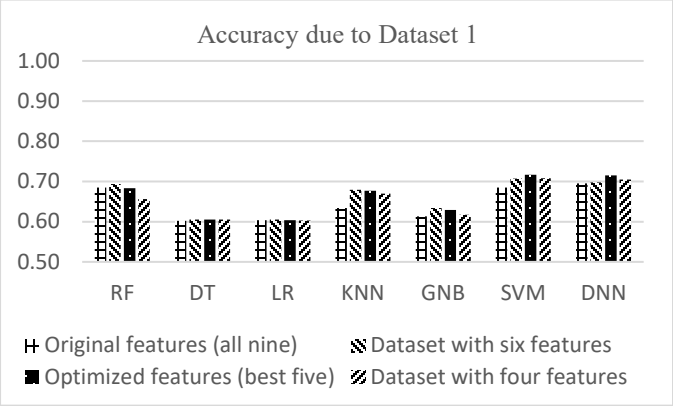

Figure 2. ML accuracy for Dataset 1.

Figures 2 and 3 suggest that the reduction of input features using the proposed method is effective. For both datasets, most ML models (except RF for Dataset 1) offer the highest accuracy for the optimized input features.

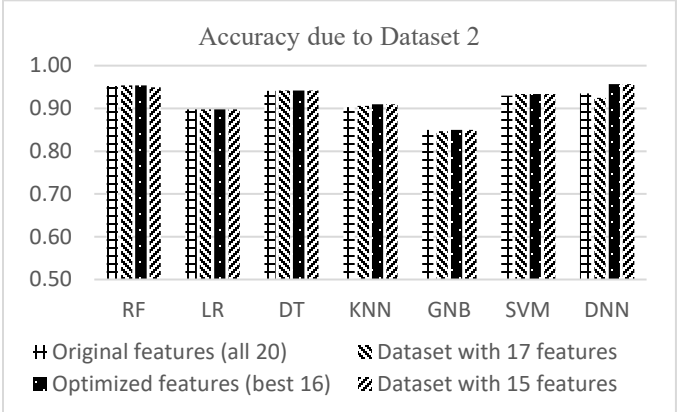

Figure 3. ML accuracy for Dataset 2.

Next, we present the precision and recall values obtained from seven different ML models. Table IV shows the precision and recall values due to Dataset 1. SVM provides the highest precision of more than 75.2% with the optimized dataset. All models (except RF and LR) show the best or similar precision value for the optimized features. DT shows the highest recall with the optimized dataset. Table V shows the precision and recall values for Dataset 2. RF provides the highest precision and recall with the optimized dataset. All models give their best or similar precision and recall with the optimized dataset.

C. *Computation and Cost Saving*

In this subsection, first we analyze the impact of input feature reduction on computation. As shown in Tables VI and VII, the total number of computations decreases for all ML models with the reduction of features for both Datasets 1 and 2. By reducing the input features from nine to five for Dataset 1 (see Table VI), DT shows the maximum (up to 59.3%) reduction in the number of computations. Similarly, by reducing the input features from 20 to 16 for Dataset 2, on average, the reduction in the number of computations is about 20% for all ML models as shown in Table VII.

TABLE IV
Precision and Recall for Dataset 1

Model	Precision				Recall			
	Dataset with original nine features	Dataset with seven features	Dataset with optimized five features	Dataset with four features	Dataset with original nine features	Dataset with seven features	Dataset with optimized five features	Dataset with four features
RF	0.709	0.674	0.654	0.597	0.358	0.452	0.442	0.430
LR	0.545	0.714	0.667	0.667	0.045	0.018	0.015	0.015
DT	0.502	0.504	0.505	0.505	0.580	0.584	0.585	0.562
KNN	0.566	0.621	0.631	0.605	0.355	0.501	0.464	0.502
GNB	0.533	0.596	0.583	0.551	0.242	0.256	0.253	0.226
SVM	0.679	0.723	0.752	0.721	0.400	0.433	0.434	0.438
DNN	0.619	0.633	0.672	0.661	0.369	0.405	0.410	0.411

TABLE V
Precision and Recall for Dataset 2

Model	Precision				Recall			
	Dataset with original 20 features	Dataset with 17 features	Dataset with optimized 16 features	Dataset with 15 features	Dataset with original 20 features	Dataset with 17 features	Dataset with optimized 16 features	Dataset with 15 features
RF	0.945	0.943	0.947	0.936	0.663	0.678	0.678	0.646
LR	0.738	0.740	0.745	0.745	0.324	0.327	0.319	0.319
DT	0.841	0.841	0.841	0.841	0.673	0.672	0.673	0.673
KNN	0.828	0.777	0.827	0.861	0.313	0.371	0.381	0.419
GNB	0.437	0.431	0.436	0.437	0.592	0.589	0.584	0.584
SVM	0.909	0.937	0.928	0.944	0.502	0.522	0.528	0.543
DNN	0.857	0.912	0.947	0.943	0.562	0.576	0.576	0.575

TABLE VI
Reduction in Computations for Dataset 1

Model	Computations for the Original Nine Data Fields	Optimized Five Data Fields	
		Computations	Reduction (%)
RF	7473556	4151976	44.5
LR	4587240	2655720	42.1
DT	259062	105422	59.3
KNN	3257820	1809900	44.4
GNB	60444	33580	44.4
SVM	16335657	9075365	44.4
DNN	2252320	1287040	42.8

TABLE VII
Reduction in Computations for Dataset 2

Model	Computations for the Original 20 Data Fields	Optimized 16 Data Fields	
		Computations	Reduction (%)
RF	78050956	62440765	19.9
LR	39359880	31679880	19.5
DT	9366114	7492891	20.0
KNN	28796400	23037120	20.0
GNB	534120	427296	20.0
SVM	574001760	459214080	19.9
DNN	19517560	15678040	19.7

Next, we analyze the impact of input data field reduction on the cost for testing substances that influence water quality. Water quality testing companies and labs charge different amount for testing different water substances [22-27]. The minimum and maximum costs required for testing the substances (i.e., input features for the ML models) associated with Dataset 1 are shown in Table VIII. The cost for testing each item can be as low as $11 (for pH level) and can be as high as $195 (for Trihalomethane). It should be noted that some service providers charge an additional $25 extra as a service charge. For every substance, the testing cost varies. For example, testing sulfate level costs as low as $15 and as high as $48. Similarly, testing pH level costs the least amount, at a minimum $11 and a maximum of $35. The highest cost is for testing Trihalomethane, reaching a minimum of $100 and a maximum of up to $195. Here, it should be noted that testing conductivity, organic carbon, trihalomethanes, and turbidity are not very important and can be excluded because of their ranking based on importance factor. The total cost to test all nine substances is $352.00.

TABLE VIII
Cost to Test for Substances in Dataset 1

Substance to Test (listed per rank)	Minimum Cost ($)	Maximum Cost ($)	Average Cost ($)
1) Sulfate	15.00	48.00	25.00
2) pH Value	11.00	35.00	23.00
3) Chloramines	15.00	35.00	25.00
4) TDS	15.00	35.00	25.00
5) Hardness	15.00	45.00	25.00
6) Conductivity	*15.00*	*35.00*	*25.00*
7) Organic Carbon	*35.00*	*64.00*	*37.00*
8) Trihalomethanes	*100.00*	*195.00*	*147.50*
9) Turbidity	*14.00*	*35.00*	*19.50*
Total Cost to Test All Nine Substances			352.00

Table IX shows the minimum and maximum costs required for testing the substances associated with Dataset 2. The cost for testing each item can be as low as $10 and can be as high as $278. The testing of barium, cadmium, chromium, copper, and silver costs the least amount, ranging between $10-$32. The highest cost is for testing Radium, reaching a minimum of $110 and a maximum of up to $278. The total cost to test all 20 substances is $863.50.

TABLE IX
Cost to Test for Substances in Dataset 2

Substance to Test (listed per rank)	Minimum Cost ($)	Maximum Cost ($)	Calculated Avg. Cost ($)
1) Aluminum	10.00	35.00	22.50
2) Cadmium	10.00	32.00	21.00
3) Perchlorate	15.00	35.00	25.00
4) Arsenic	15.00	49.00	32.00
5) Silver	10.00	32.00	21.00

Substance to Test (listed per rank)	Minimum Cost ($)	Maximum Cost ($)	Calculated Avg. Cost ($)
6) Ammonia	31.00	40.00	35.50
7) Chloramine	15.00	35.00	25.00
8) Nitrates	15.00	37.00	26.00
9) Uranium	100.00	245.00	172.50
10) Nitrites	15.00	37.00	26.00
11) Viruses	15.00	35.00	25.00
12) Radium	110.00	278.00	194.00
13) Chromium	10.00	32.00	21.00
14) Bacteria	20.00	75.00	47.50
15) Barium	10.00	32.00	21.00
16) Lead	15.00	35.00	25.00
17) Copper	*10.00*	*32.00*	*21.00*
18) Fluoride	*15.00*	*35.00*	*25.00*
19) Mercury	*35.00*	*56.00*	*45.50*
20) Selenium	*15.00*	*49.00*	*32.00*
Total Cost to Test All 20 Substances			863.50

According to Table II, out of the nine substances in Dataset 1, trihalomethanes is ranked 8; however, according to Table VIII, the trihalomethanes test is most expensive. Similarly, out of the 20 substances in Dataset 2, uranium is ranked 9th and radium is ranked 12th (see Table III); however, those tests are most expensive (see Table IX). Therefore, it can be presumed that the feature reduction may help save water testing cost.

Next, we summarize the accuracy of ML models and the total cost to test the water quality in Tables X and XI. For Dataset 1, the accuracy improves (0.634 to 0.677 for KNN) or remains about the same for all ML models when input features are reduced from nine to five. This is probably because the less impactful features are excluded. However, as shown in Table X, this reduction in input features drops the test cost from $352 to $229, resulting in a more than 65% saving.

TABLE X
ML Accuracy and Test Cost for Dataset 1

Model	Original Dataset with Nine Features		Optimized Dataset with Five Features	
	ML Accuracy	Total Cost ($)	ML Accuracy	Total Cost ($)
RF	0.685		0.683	
LR	0.600		0.604	
DT	0.602		0.605	
KNN	0.634	352.00	0.677	123.00
GNB	0.613		0.629	
SVM	0.685		0.717	
DNN	0.695		0.715	

As shown in Table XI, for Dataset 2, the accuracy slightly improves or remains about the same for all ML models when input features are reduced from 20 to 16. Again, this is probably because the less impactful features are excluded. However, this reduction in input features drops the test cost from $863.50 to $740.00 (more than 14% saving).

TABLE XI
ML Accuracy and Test Cost for Dataset 2

Model	Original Dataset with 20 Features		Optimized Dataset with 16 Features	
	ML Accuracy	Total Cost ($)	ML Accuracy	Total Cost ($)
RF	0.952		0.954	
LR	0.898		0.898	
DT	0.941		0.942	
KNN	0.903	863.50	0.910	740.00
GNB	0.850		0.850	
SVM	0.930		0.934	
DNN	0.935		0.957	

V. Conclusions

Applications such as water quality testing involves many factors to consider. Determining factors that are very expensive and time consuming to collect and test, but may not have any importance is challenging. This study introduces an effective machine learning based methodology for dimensionality reduction to facilitate cost-effective data collection and analysis. According to the proposed methodology, the RFECV, PI, and RF methods are used to identify the most informative features and discard less significant ones. Seven different machine learning models (RF, DT, LR, KNN, GNB, SVM, and DNN) are employed to evaluate the effectiveness of the proposed methodology. Through experimentation on two different datasets from Kaggle used for water quality prediction, the results demonstrate the ability of the proposed methodology in achieving substantial reduction in input features without compromising the performance of the machine learning models. According to experimental results, the proposed methodology helps reduce the computations by up to 59% and the water quality test cost by up to 65%, while keeping the accuracy up to 95%. This research not only provides an impactful solution to the prevalent issue of dimensionality but also contributes to the broader conversation on optimizing machine learning workflows. Future work may explore further refinements of the methodology, ensuring its applicability across a spectrum of domains and datasets.

References

[1] T. Ahmed, R. R. Paul, M. A. Alam, M. T. Hasan, and M. R. Rab, "Performance Comparison of Different Machine Learning Classifiers in Categorizing Bangla News Articles," IEEE International Conference on Natural Language Processing (ICNLP), Xi'an, China, 2022, pp. 376-379, doi: 10.1109/ICNLP55136.2022.00069.

[2] M. A. I. Siddique, F. Haque, and M. S. H. Shojol, "Comparative Analysis of Feature Selection Techniques and Machine Learning Classifiers for Accurate Classification of Pumpkin Seeds," IEEE International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), Dhaka, Bangladesh, 2023, pp. 214-218, doi: 10.1109/ICICT4SD59951.2023.10303434.

[3] B. K. Bhavitha, A. P. Rodrigues, and N. N. Chiplunkar, "Comparative study of machine learning techniques in sentimental analysis," IEEE

International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2017, pp. 216-221, doi: 10.1109/ICICCT.2017.7975191.

[4] H. Wang, C. Ma, and L. Zhou, “A Brief Review of Machine Learning and Its Application,” IEEE International Conference on Information Engineering and Computer Science, Wuhan, China, 2009, pp. 1-4, doi: 10.1109/ICIECS.2009.5362936.

[5] V. Menon, K. Weger, B. Mesmer and S. Gholston, “Using Big Data Analytics for Sentiment Analysis to Explore Team Communication Dynamics in Human Machine Interactions for Team Situational Awareness,” IEEE International Conference on Human-Machine Systems (ICHMS), Orlando, FL, USA, 2022, pp. 1-6, doi: 10.1109/ICHMS56717.2022.9980604.

[6] H. P. Muthukrishnan and D. A. Szafir, “Using Machine Learning and Visualization for Qualitative Inductive Analyses of Big Data,” IEEE Workshop on Machine Learning from User Interaction for Visualization and Analytics (MLUI), Vancouver, BC, Canada, 2019, pp. 1-7, doi: 10.1109/MLUI52769.2019.10075566.

[7] H. D. Nguyen, T. Q. D. Nguyen, H. N. Thi, B. Q. Lap, and T. -T. -H. Phan, “The Use of Machine Learning Algorithms for Evaluating Water Quality Index: A Survey and Perspective,” IEEE International Conference on Multimedia Analysis and Pattern Recognition (MAPR), Phu Quoc, Vietnam, 2022, pp. 1-6, doi: 10.1109/MAPR56351.2022.9924736.

[8] J. M. Bautista, Q. A. I. Quiwa, and R. S. J. Reyes, “Machine Learning Analysis for Remote Prenatal Care,” IEEE REGION 10 CONFERENCE (TENCON), Osaka, Japan, 2020, pp. 397-402, doi: 10.1109/TENCON50793.2020.9293890.

[9] Y. Zhang, Q. Shen, J. Guo, and Y. Jia, “Portfolio Trading of Financial Products Based on Machine Learning,” IEEE International Conference on Machine Learning and Cybernetics (ICMLC), Japan, 2022, pp. 72-79, doi: 10.1109/ICMLC56445.2022.9941281.

[10] T. Ganegedara and A. Lopatenko, “Natural Language Processing with TensorFlow: The definitive NLP book to implement the most sought-after machine learning models and tasks,” Packt Publishing, 2022.

[11] A. L’Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, “Machine Learning With Big Data: Challenges and Approaches,” IEEE Access, vol. 5, pp. 7776-7797, 2017, doi: 10.1109/ACCESS.2017.2696365.

[12] S. Loussaief and A. Abdelkrim, "Deep learning vs. bag of features in machine learning for image classification," IEEE International Conference on Advanced Systems and Electric Technologies (IC_ASET), Hammamet, Tunisia, 2018, pp. 6-10, doi: 10.1109/ASET.2018.8379825.

[13] T. Mahara, V. L. H. Josephine, R. Srinivasan, P. Prakash, A. D. Algarni and O. P. Verma, "Deep vs. Shallow: A Comparative Study of Machine Learning and Deep Learning Approaches for Fake Health News Detection," IEEE Access, vol. 11, pp. 79330-79340, 2023, doi: 10.1109/ACCESS.2023.3298441.

[14] X. Wang, T. Hu and L. Tang, "A Multiobjective Evolutionary Nonlinear Ensemble Learning With Evolutionary Feature Selection for Silicon Prediction in Blast Furnace," IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 5, pp. 2080-2093, May 2022, doi: 10.1109/TNNLS.2021.3059784.

[15] N. V. Varghese, A. Azim, and Q. H. Mahmoud, "A Feature-Based Machine Learning Approach for Mixed-Criticality Systems," IEEE International Conference on Industrial Technology (ICIT), Valencia, Spain, 2021, pp. 699-704, doi: 10.1109/ICIT46573.2021.9453482.

[16] X. B. Li, J. Y. Li, and R. H. Wang, “Dimensionality Reduction Using MCE-optimized LDA transformation,” IEEE International Conference on Acoustics, Speech and Signal Processing, May 2004.

[17] N. Khosla, “MS Thesis: Dimensionality Reduction Using Factor Analysis,” School of Microelectronics Engineeing, Griffith University, Australia, 2004.

[18] V. B. Shereena and J. M. David, “Comparative Study of Dimensionality Reduction Techniques Using PCA and LDA for Content Based Image Retrieval,” IEEE International Conference on Computer Science and Information Technology, doi: 10.5121/csit.2015.50905, Apr 2015.

[19] T. Zhang and B. Yang, “Big Data Dimension Reduction Using PCA,” IEEE International Conference on Smart Cloud, doi: 10.1109/SmartCloud.2016.33, Nov 2016.

[20] R. R. Zaberi, A. M. Abdulazeez, D. Q. Zeebaree, D. A. Zaberi, and J. N. Saeed, “A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feaure Extraction,” JASTT, vol. 1, no. 2, June 2020.

[21] D. Shilane, “Automated Feature Reduction in Machine Learning,” IEEE Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2022, pp. 0045-0049, doi: 10.1109/CCWC54503.2022.9720821.

[22] D. Hou, X. Song, G. Zhang, H. Zhang, and H. Loaiciga, “An early warning and control system for urban, drinking water quality protection: China’s experience,” Environ. Sci. Pollut. Res. Int., vol. 20, no. 7, pp. 4496–508, 2013.

[23] “Tap and Well water tests; water and environment testing,” ETR lab, https://etrlabs.com/water-tests/

[24] “Private drinking water testing laboratory,” certified by the New Jersey Department of Environmental Protection, Precision Analytical Services, https://drinkingwatertesting.com/private_test.php

[25] M. I. K. Haq, F. Dwi Ramadhan, F. Az-Zahra, L. Kurniawati, and A. Helen, “Classification of water potability using machine learning algorithms,” International Conference on Artificial Intelligence and Big Data Analytics, 2021.

[26] R. Alnaqeb, F. Alrashdi, K. Alketbi and H. Ismail, “Machine Learning-based Water Potability Prediction,” IEEE/ACS International Conference on Computer Systems and Applications (AICCSA), UAE, 2022, pp. 1-6, doi: 10.1109/AICCSA56895.2022.10017579.

[27] M. Zhu, J. Wang, X. Yang, Y. Zhang, L. Zhang, H. Ren, B. Wu, and L. Ye, “A review of the application of machine learning in water quality evaluation,” Eco-Environment & Health, vol. 1, no. 2, pp. 107–116, 2022.

[28] Y. Zhou, “Real-time probabilistic forecasting of river water quality under data missing situation: Deep learning plus post-processing techniques,” Journal of Hydrology, vol. 589, 2020, ISSN 0022-1694.

[29] B. M. S. Hasan and A. M. Abdulazeez, “A review of Principal Component Analysis Algorithm for Dimensionality Reduction,” JSCDM, 2021.

[30] A. Z. Mustaqim, S. Adi, Y. Pristyanto, and Y. Astuti, “The Effect of Recursive Feature Elimination with Cross-Validation (RFECV) Feature Selection Algorithm toward Classifier Performance on Credit Card Fraud Detection,” IEEE International Conference on Artificial Intelligence and Computer Science Technology (ICAICST), Yogyakarta, Indonesia, 2021, pp. 270-275, doi: 10.1109/ICAICST53116.2021.9497842.

[31] F. Yang, P. Piao, Y. Lai, and L. Pei, “Margin based permutation variable importance: A stable importance measure for random forest,” IEEE International Conference on Intelligent Systems and Knowledge Engineering (ISKE), Nanjing, China, 2017, pp. 1-8, doi: 10.1109/ISKE.2017.8258842.

[32] P. Swetha, A. H. K. P. Rasheed, and V. P. Harigovindan, “Random Forest Regression based Water Quality Prediction for Smart Aquaculture,” IEEE International Conference on Computing and Communication Systems (I3CS), Shillong, India, 2023, pp. 1-5, doi: 10.1109/I3CS58314.2023.10127488.

[33] B. A. de Abreu, A. Berndt, I. S. Campos, C. Meinhardt, J. T. Carvalho, M. Grellert, and S. Bampi, “Fast Logic Optimization Using Decision Trees,” IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Korea, 2021, pp. 1-5, doi: 10.1109/ISCAS51556.2021.9401664.

[34] P. L. Lik Pao and M. A. Ismail, “Loan Eligibility Classification Using Logistic Regression,” IEEE International Conference On Software Engineering and Computer Systems (ICSECS), Penang, Malaysia, 2023, pp. 1-4, doi: 10.1109/ICSECS58457.2023.10256402.

[35] R. Zhang and D. Perkins, “A Quick K-Nearest Neighbor Algorithm with Aggregated Centroids,” IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2021, pp. 1198-1202, doi: 10.1109/ICAICA52286.2021.9498032.

[36] Herman, L. Syafie, D. Indra, A. Djamalilleil, Nirsal, H. Hamrul, S. Anraeni, and L. B. Ilmawan, “Comparison of Artificial Neural Network and Gaussian Naïve Bayes in Recognition of Hand-Writing Number,” IEEE Conference on Computer and Information Technology (EIConCIT), Makassar, Indonesia, 2018, pp. 276-279, doi: 10.1109/EIConCIT.2018.8878651.

[37] A. A. Dzilfadhilah, A. A. Vinaya, and N. Yesica, "Fault Classification of Pump Using Support Vector Machine (SVM) Method," IEEE International Conference on Information Technology and Education (ICIT&E), Malang, Indonesia, 2022, pp. 201-205, doi: 10.1109/ICITE54466.2022.9759882.

[38] M. A. Haque, J. S. Rani Alex, and N. Venkatesan, "Evaluation of Modified Deep Neural Network Architecture Performance for Speech Recognition," IEEE International Conference on Intelligent and Advanced System (ICIAS), Kuala Lumpur, Malaysia, 2018, pp. 1-5, doi: 10.1109/ICIAS.2018.8540636.

[39] "Water quality: Dataset for water quality classification," Kaggle, 2023, https://www.kaggle.com/datasets/mssmartypants/water-quality

[40] "Decision Tree: Classification and Complexity," scikit-learn, 2023, https://scikit-learn.org/stable/modules/tree.html#complexity

[41] "Nearest Neighbors: Classification and Complexity," scikit-learn, 2023, https://scikit-learn.org/stable/modules/neighbors.html

[42] "Support Vector Machines: Classification and Complexity," scikit-learn, 2023, https://scikit-learn.org/stable/modules/svm.html#complexity

[43] "Neural Network Models (Supervised): Classification and Complexity," scikit-learn, 2023, https://scikit-learn.org/stable/modules/neural_networks_supervised.html#complexity

[44] "Drinking water tests and fees," Community Science Institute, 2023, http://www.communityscience.org/certified-water-testing/drinking-water-tests-fees/

[45] "Drinking Water Analysis," Analytical Testing Laboratories, AMTEST Lab, 2023, http://amtestlab.com/prices/drinking_water.asp

[46] "Safe Drinking Water Act Price List," Wisconsin State Lab of Hygiene at University of Wisconsin–Madison, 2023, https://www.slh.wisc.edu/environmental/water/sdwa-price-list/

[47] "Current Prices," UCDAVIS Analytical Lab, 2023, https://anlab.ucdavis.edu/Prices

[48] "Analytical Services and Prices," University of New Hampshire, 2023, https://wrrc.unh.edu/analytical-services-prices

[49] "Laboratories Services and Price List," Oklahoma State University, 2023, https://extension.okstate.edu/programs/soil-testing/laboratory-services-and-price-list.html

Abu Asaduzzaman (Senior Member, IEEE) received the PhD and MS degrees, both in Computer Engineering, from Florida Atlantic University, Florida. Currently, Dr. Asaduzzaman is an Associate Professor of Computer Engineering at Wichita State University, Kansas. He serves as Undergraduate Program Director in his department. His research interests include high performance computing, machine learning, and data analysis. He has authored more than 20 refereed journal articles, more than 80 peer-reviewed conference papers, two book-chapters, and one U.S. Patent out of his research work. He has received research grants from Kansas NSF EPSCoR, Nvidia, and NetApp. Dr. Asaduzzaman has served as a reviewer of NSF programs and IEEE journals. As invited speaker, he has presented his research work in professional forums at various institutions including the Nara Institute of Science and Technology in Japan, the Old Dominion University in Virginia, the International Society for Engineering Research and Development in Thailand, and the IEEE Wichita Professional Section in Kansas.

Md Raihan Uddin (Student Member, IEEE) received his BS degree in Electrical Electronic and Communication Engineering from Bangladesh University of Professionals (BUP) in 2018. Currently he is pursuing his PhD degree in the Electrical and Computer Engineering (ECE) Department at Wichita State University (WSU). Mr. Uddin conducts research activities in the WSU Computer Architecture and Parallel Programming Laboratory. His research area includes heterogeneous and distributed systems, machine learning, and high-performance computing. He has published three peer-reviewed IEEE conference papers, and submitted one journal article and two conference papers out of his research work. He has received the DEAN's Certificate award at BUP. He is a student member of IEEE and an active member of the ECE PhD Club.

Fadi Sibai joined the Gulf University for Science and Technology in Kuwait as Associate Dean, College of Engineering and Architecture, in 2022. Prior to that, he served as acting Dean of the School of Engineering, American International University, as Dean of the College of Computer Engineering and Science, Prince Mohammad Bin Fahd University, and as Program Director in the College of Information Technology at the UAE University. He also taught Engineering at the University of California, and the University of Akron, USA. He received IBM and nVIDIA equipment grants, an IBM Faculty award, and research grants from the Emirates Foundation and the University of Akron. He authored or co-authored about 250 publications and technical reports. He also served in various capacities on program and organizing committees of over 20 international conferences. Dr. Sibai received the PhD and MS degrees from Texas A&M University, all in Electrical and Computer Engineering.