**Water Quality Prediction Using Hybrid Machine Learning**

**Abstract:-**

One of the key functions of global water resource management authorities is river water quality (WQ) assessment. A water quality index (WQI) is developed for water assessments considering numerous quality-related variables. WQI assessments typically take a long time and are prone to errors during sub-indices generation. This can be tackled through the latest machine learning (ML) techniques renowned for superior accuracy. In this study, water samples were taken from the wells in the study area (North Pakistan) to develop WQI prediction models. Four standalone algorithms, i.e., random trees (RT), random forest (RF), M5P, and reduced error pruning tree (REPT), were used in this study. In addition, 12 hybrid data-mining algorithms (a combination of standalone, bagging (BA), cross-validation parameter selection (CVPS), and randomizable filtered classification (RFC)) were also used. Using the 10-fold cross-validation technique, the data were separated into two groups (70:30) for algorithm creation. Ten random input permutations were created using Pearson correlation coefficients to identify the best possible combination of datasets for improving the algorithm prediction. The variables with very low correlations performed poorly, whereas hybrid algorithms increased the prediction capability of numerous standalone algorithms.

**Introduction:-**

Water pollution is one of the critical challenges of the modern world where the goals such as the United Nations Sustainable Development Goals (UN-SDGs) and a smart and sustainable planet are being pursued. All societies, ecologies, and productions require abundant clean water supplies for farming, The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale De Meo. drinking, sanitation, and energy production. The global water crisis is among the serious threats the human race faces these days. Accordingly, the quantity and quality of groundwater are significant global concerns . Many diseases occur due to polluted water, like cholera, diarrhea, typhoid, amebiasis, hepatitis, gastroenteritis, giardiasis, campylobacteriosis, scabies, and worm infections. Almost 1.6 million people died due to diarrhea in 2017 alone. Water pollutants impact its conditions, which impact human health and marine life.

Inadequate sewage networks, uncontrolled and improperly planned urbanization, and dumping of industrial trash, pesticides, and fertilizers contribute to water pollution. Such pollution is more evident in local rivers or water channels closer to urban developments. With both non-point and point sources, river pollution is becoming a more significant problem and presents a tough challenge to global water management authorities. Such pollution seriously deteriorates water quality (WQ). WQ degradation substantially impacts aquatic life and the availability of clean water for drinking and agricultural purposes. The pollution challenge is harder to tackle in developing countries which frequently go through times of economic fluctuations. Further each development action can have severe environmental consequences. For example, with an increase in the population and demand for more resources, the requirement for more agricultural production pressures soils’ organic fertility, increasing the demand for artificial fertilizers to enhance yield. Accordingly, surplus fertilizers are frequently dumped into rivers and waterways that pollute ground and underground water sources. This increases the need for WQ assessment and surveillance. WQ surveillance and evaluation are critical for environmental, climate, and human health protection. This can be achieved through timely, efficient, and long-term water management plans. The WQ is assessed through the water quality index (WQI). WQI helps guide policymakers’ actions and decisions. However, calculating WQI is not a simple process due to the involvement of multiple sub-indices and equations. WQI is a non-dimensional index derived from defined WQ variables. It uses variables such as pH (potential of hydrogen), DO (dissolved oxygen), TSS (total suspended solids), BOD (biological oxygen demand), AN (ammoniacalnitrogen), COD (chemical oxygen demand), and others. The associated matrices enable a definite evaluation of WQ. Measurements of variables such as Ca2+, Mg2+, NO3, and others are commonly used to estimate groundwater quality indicators (GQIs) . Several aspects of water, including physical, chemical, biological, and radiological, are included in the assessment of WQ. In addition, WQI is a frequently used technique for assessing the effectiveness or failure of WQ management measures. Some examples of WQIs include the Canadian WQI (CQI), United States National Sanitation Foundation WQI (NSFWQI), Interim National Water Quality Standards for Malaysia (INWQS), British Columbia WQI (BCWQI), Oregon WQI (OWQI), Florida Stream WQI (FWQI), and others. WQI is calculated through multiple methods and algorithms around the globe.

**Existing System:-**

developed a regression tree (RT) algorithm and a support vector regression (SVR) algorithm for predicting wastewater quality indicators and discovered that the SVR model provided the best results. Kayaalp et al. developed a hybrid SVR model using monthly WQ param eter data with the fifireflfly algorithm (FFA) to forecast WQI. The algorithm showed a signifificant increase in predic tion performance compared to the standalone SVR model. Kamyab-Talesh et al.looked into the optimization of the SVM algorithm to investigate the factors having the highest impact on the WQI. The authors observed that nitrate is the most crucial parameter for WQI prediction. Wang et al. analyzed three ML algorithms, SVR, SVR-GA (genetic algo rithm), and SVR-PSO (particle swarm optimization), to pre dict WQI and compared their performance. Since decision tree-based algorithms (i.e., M5P, RF, RT, REPT, and others) lack hidden units and modeling clarity, they can produce superior modeling results than ANFIS and ANN.

**Disadvantages:-**

Not well in prediction accuracy.

Its will not supported with dynamic changes.

**Proposed system:-**

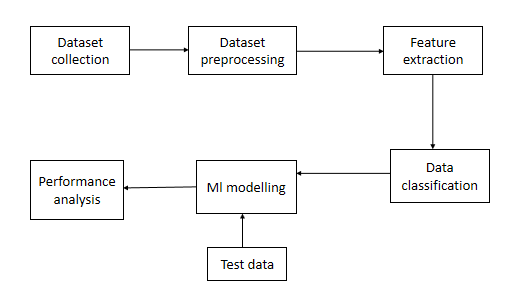
that the data collection was initially performed, and followingly different WQ parameters were calculated from the water samples. The data was then distributed into testing and validation datasets. From the testing datasets, the best input combination was identified. Finally, multiple algorithms were applied to the best varieties, and an algorithm assessment was conducted for the best possible algorithm selection to predict WQI.

the algorithms RF, naive bayeis, logistic regression (LR) and decision tree(DT) have the highest prediction power. All algorithms were validated as the predicted WQI was compared with measured WQI for each model at each testing dataset

**Advanatages:-**

* Better efficiency and accuracy
* ML algorithms is a powerful data modelling tool that is able to capture and represent complex input/output relationships.
* Able to scope with large dataset.
* Accurate prediction.

**System Architecture:-**

****

**System Requirements:-**

**Software and Hardware Requirements:**

**Hardware:**

 OS – Windows 7, 8 and 10 (32 and 64 bit)

RAM – 4GB

**Software:**

Python / Anaconda Navigator

**Integrated module level:**

The meaning of Integration testing is quite straightforward- Integrate/combine the unit tested module one by one and test the behavior as a combined unit.

The main function or goal of this testing is to test the interfaces between the units/modules. We normally do Integration testing after “Unit testing”. Once all the individual units are created and tested, we start combining those “Unit Tested” modules and start doing the integrated testing.

The main function or goal of this testing is to test the interfaces between the units/modules. The individual modules are first tested in isolation. Once the modules are unit tested, they are integrated one by one, till all the modules are integrated, to check the combinational behavior, and validate whether the requirements are implemented correctly or not.

Here we should understand that Integration testing does not happen at the end of the cycle, rather it is conducted simultaneously with the development. So in most of the times, all the modules are not actually available to test and here is what the challenge comes to test something which does not exist!

**Advantages:**

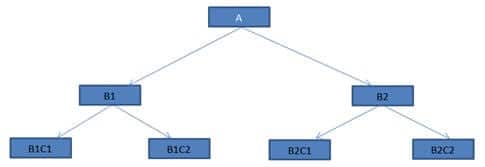
* This testing makes sure that the integrated modules/components work properly.
* Integration testing can be started once the modules to be tested are available. It does not require the other module to be completed for testing to be done, as Stubs and Drivers can be used for the same.
* It detects the errors related to the interface.

**Test Integration Approaches**

There are fundamentally 2 approaches for doing test integration:

1.Bottom-up approach

2.Top-down approach.



**Bottom-up approach**

Bottom-up testing, as the name suggests starts from the lowest or the innermost unit of the application, and gradually moves up. The Integration testing starts from the lowest module and gradually progresses towards the upper modules of the application. This integration continues till all the modules are integrated and the entire application is tested as a single unit.

**Top-down approach**

This technique starts from the topmost module and gradually progress towards the lower modules. Only the top module is unit tested in isolation. After this, the lower modules are integrated one by one. The process is repeated until all the modules are integrated and tested.

**Algoritnms:**

1. **Logistic regression**
2. **Randomforest algorithm**
3. **Decision tree**
4. **Svm**
5. **Knn**
6. **Xgboost**
7. **Gradient boosting**

**2.Randomforest algorithm :**

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision *trees*, resulting in a *forest of trees*, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

**HOW RANDOM FOREST WORKS**

The following are the basic steps involved in performing the random forest algorithm

1. Pick N random records from the dataset.
2. Build a decision tree based on these N records.
3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
4. For classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

**4.Support vector machine:**

Support Vector Machines (SVMs in short) are machine learning algorithms that are used for classification and regression purposes. SVMs are one of the powerful machine learning algorithms for classification, regression and outlier detection purposes. An SVM classifier builds a model that assigns new data points to one of the given categories. Thus, it can be viewed as a non-probabilistic binary linear classifier.

SVMs can be used for linear classification purposes. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using the kernel trick. It enable us to implicitly map the inputs into high dimensional feature spaces.

**Hyperplane:**

A hyperplane is a decision boundary which separates between given set of data points having different class labels. The SVM classifier separates data points using a hyperplane with the maximum amount of margin. This hyperplane is known as the maximum margin hyperplane and the linear classifier it defines is known as the maximum margin classifier.

### Support Vectors:

Support vectors are the sample data points, which are closest to the hyperplane. These data points will define the separating line or hyperplane better by calculating margins.

**Margin**

A margin is a separation gap between the two lines on the closest data points. It is calculated as the perpendicular distance from the line to support vectors or closest data points. In SVMs, we try to maximize this separation gap so that we get maximum margin.

**DOMAIN OVERVIEW:**

**MACHINE LEARNING**

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The machine receives data as input, use an algorithm to formulate answers.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation.

Machine learning is also used for a variety of task like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

**Machine Learning vs. Traditional Programming**

Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

**DATA RULES**

**COMPUTER**

**OUTPUT**

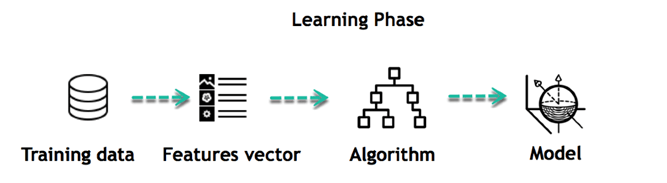
**Machine Learning**

## How does Machine learning work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the **learning**and **inference**. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the **data**. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a **feature vector.** You can think of a feature vector as a subset of data that is used to tackle a problem.

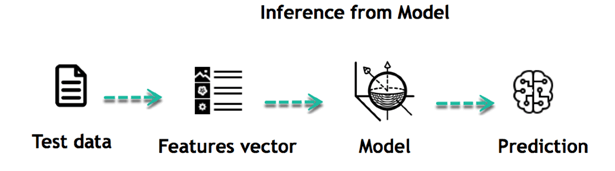
The machine uses some fancy algorithms to simplify the reality and transform this discovery into a **model**. Therefore, the learning stage is used to describe the data and summarize it into a model.



For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant: This is the model

#### Inferring

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

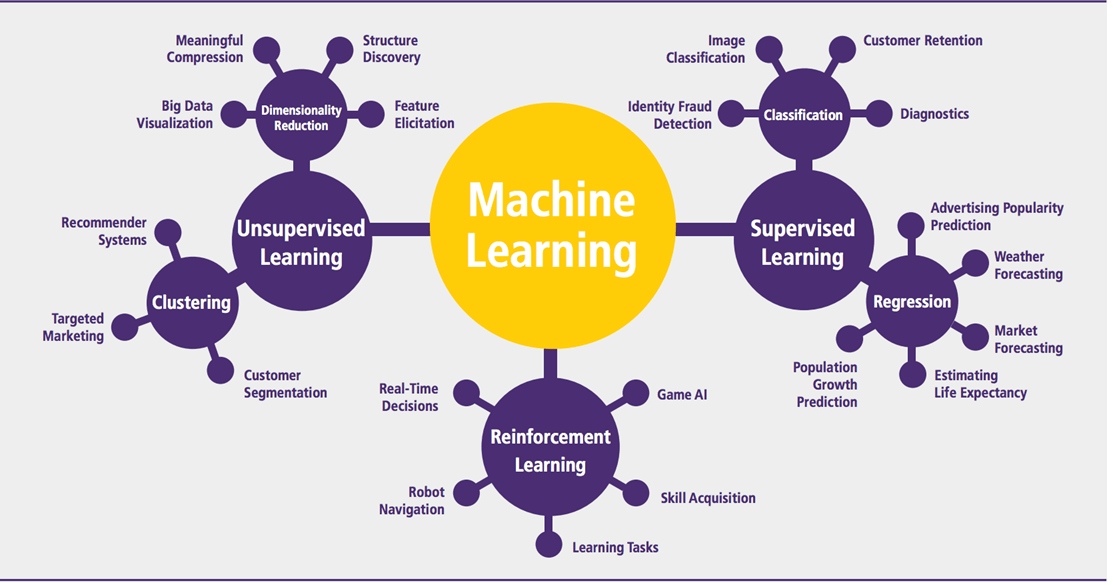


The life of Machine Learning programs is straightforward and can be summarized in the following points:

1. Define a question
2. Collect data
3. Visualize data
4. Train algorithm
5. Test the Algorithm
6. Collect feedback
7. Refine the algorithm
8. Loop 4-7 until the results are satisfying
9. Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

## Machine learning Algorithms and where they are used?



Machine learning can be grouped into two broad learning tasks: Supervised and Unsupervised. There are many other algorithms

#### Supervised learning

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans.

You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

|  |  |  |
| --- | --- | --- |
| **Algorithm Name** | **Description** | **Type** |
| **Linear regression** | Finds a way to correlate each feature to the output to help predict future values. | Regression |
| **Logistic regression** | Extension of linear regression that's used for classification tasks. The output variable 3is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors) | Classification |
| **Decision tree** | Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made | Regression Classification |
| **Naive Bayes** | The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event. | Regression Classification |
| **Support vector machine** | Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non-linear solver. | Regression (not very common) Classification |
| **Random forest** | The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction. | Regression Classification |
| **AdaBoost** | Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome | Regression Classification |
| **Gradient-boosting trees** | Gradient-boosting trees is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it. | Regression Classification |

* Classification task
* Regression task

#### Classification

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female.

The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

#### **Regression**

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

#### Unsupervised learning

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns)

You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Description** | **Type** |
| **K-means clustering** | Puts data into some groups (k) that each contains data with similar characteristics (as determined by the model, not in advance by humans) | Clustering |
| **Gaussian mixture model** | A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters | Clustering |
| **Hierarchical clustering** | Splits clusters along a hierarchical tree to form a classification system.  Can be used for Cluster loyalty-card customer | Clustering |
| **Recommender system** | Help to define the relevant data for making a recommendation. | Clustering |
| **PCA/T-SNE** | Mostly used to decrease the dimensionality of the data. The algorithms reduce the number of features to 3 or 4 vectors with the highest variances. | Dimension Reduction |

**Application of Machine learning**

**Augmentation**:

* Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions. The primary user is to reduce errors due to human bias.

**Automation**:

* Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.

**Finance Industry**

* Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.

**Government organization**

* The government makes use of ML to manage public safety and utilities. Take the example of China with the massive face recognition. The government uses Artificial intelligence to prevent jaywalker.

**Healthcare industry**

* Healthcare was one of the first industry to use machine learning with image detection.

**Marketing**

* Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, marketing department relies on AI to optimize the customer relationship and marketing campaign.

**Example of application of Machine Learning in Supply Chain**

Machine learning gives terrific results for visual pattern recognition, opening up many potential applications in physical inspection and maintenance across the entire supply chain network.

Unsupervised learning can quickly search for comparable patterns in the diverse dataset. In turn, the machine can perform quality inspection throughout the logistics hub, shipment with damage and wear.

For instance, IBM's Watson platform can determine shipping container damage. Watson combines visual and systems-based data to track, report and make recommendations in real-time.

In past year stock manager relies extensively on the primary method to evaluate and forecast the inventory. When combining big data and machine learning, better forecasting techniques have been implemented (an improvement of 20 to 30 % over traditional forecasting tools). In term of sales, it means an increase of 2 to 3 % due to the potential reduction in inventory costs.

**Example of Machine Learning Google Car**

For example, everybody knows the Google car. The car is full of lasers on the roof which are telling it where it is regarding the surrounding area. It has radar in the front, which is informing the car of the speed and motion of all the cars around it. It uses all of that data to figure out not only how to drive the car but also to figure out and predict what potential drivers around the car are going to do. What's impressive is that the car is processing almost a gigabyte a second of data.

Deep Learning

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks. The machine uses different layers to learn from the data. The depth of the model is represented by the number of layers in the model. Deep learning is the new state of the art in term of AI. In deep learning, the learning phase is done through a neural network.

**Reinforcement Learning**

Reinforcement learningis a subfield of machine learning in which systems are trained by receiving virtual "rewards" or "punishments," essentially learning by trial and error. Google's DeepMind has used reinforcement learning to beat a human champion in the Go games. Reinforcement learning is also used in video games to improve the gaming experience by providing smarter bot.

One of the most famous algorithms are:

* Q-learning
* Deep Q network
* State-Action-Reward-State-Action (SARSA)
* Deep Deterministic Policy Gradient (DDPG)

**Applications/ Examples of deep learning applications**

**AI in Finance:**The financial technology sector has already started using AI to save time, reduce costs, and add value. Deep learning is changing the lending industry by using more robust credit scoring. Credit decision-makers can use AI for robust credit lending applications to achieve faster, more accurate risk assessment, using machine intelligence to factor in the character and capacity of applicants.Underwrite is a Fintech company providing an AI solution for credit makers company. underwrite.ai uses AI to detect which applicant is more likely to pay back a loan. Their approach radically outperforms traditional methods.

**AI in HR:**Under Armour, a sportswear company revolutionizes hiring and modernizes the candidate experience with the help of AI. In fact, Under Armour Reduces hiring time for its retail stores by 35%. Under Armour faced a growing popularity interest back in 2012. They had, on average, 30000 resumes a month. Reading all of those applications and begin to start the screening and interview process was taking too long. The lengthy process to get people hired and on-boarded impacted Under Armour's ability to have their retail stores fully staffed, ramped and ready to operate.

At that time, Under Armour had all of the 'must have' HR technology in place such as transactional solutions for sourcing, applying, tracking and onboarding but those tools weren't useful enough. Under armour choose **HireVue**, an AI provider for HR solution, for both on-demand and live interviews. The results were bluffing; they managed to decrease by 35% the time to fill. In return, the hired higher quality staffs.

**AI in Marketing:**AI is a valuable tool for customer service management and personalization challenges. Improved speech recognition in call-center management and call routing as a result of the application of AI techniques allows a more seamless experience for customers.

For example, deep-learning analysis of audio allows systems to assess a customer's emotional tone. If the customer is responding poorly to the AI chatbot, the system can be rerouted the conversation to real, human operators that take over the issue.

Apart from the three examples above, AI is widely used in other sectors/industries.

**Artificial Intelligence**

ML

**Machine Learning**

**Deep Learning**

|  |  |  |
| --- | --- | --- |
|  | **Machine Learning** | **Deep Learning** |
| **Data Dependencies** | Excellent performances on a small/medium dataset | Excellent performance on a big dataset |
| **Hardware dependencies** | Work on a low-end machine. | Requires powerful machine, preferably with GPU: DL performs a significant amount of matrix multiplication |
| **Feature engineering** | Need to understand the features that represent the data | No need to understand the best feature that represents the data |
| **Execution time** | From few minutes to hours | Up to weeks. Neural Network needs to compute a significant number of weights |
| **Interpretability** | Some algorithms are easy to interpret (logistic, decision tree), some are almost impossible (SVM, XGBoost) | Difficult to impossible |

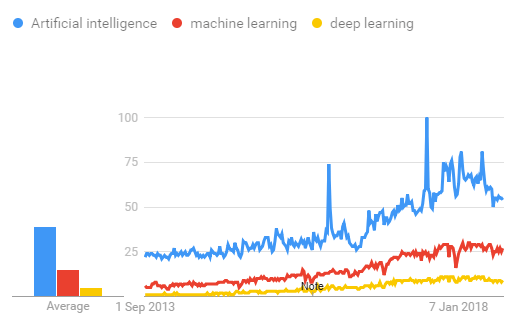
## Difference between Machine Learning and Deep Learning

## When to use ML or DL?

In the table below, we summarize the difference between machine learning and deep learning.

|  |  |  |
| --- | --- | --- |
|  | **Machine learning** | **Deep learning** |
| **Training dataset** | Small | Large |
| **Choose features** | Yes | No |
| **Number of algorithms** | Many | Few |
| **Training time** | Short | Long |

With machine learning, you need fewer data to train the algorithm than deep learning. Deep learning requires an extensive and diverse set of data to identify the underlying structure. Besides, machine learning provides a faster-trained model. Most advanced deep learning architecture can take days to a week to train. The advantage of deep learning over machine learning is it is highly accurate. You do not need to understand what features are the best representation of the data; the neural network learned how to select critical features. In machine learning, you need to choose for yourself what features to include in the model.



## TensorFlow

the most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations.

To give a concrete example, Google users can experience a faster and more refined the search with AI. If the user types a keyword a the search bar, Google provides a recommendation about what could be the next word.

Google wants to use machine learning to take advantage of their massive datasets to give users the best experience. Three different groups use machine learning:

* Researchers
* Data scientists
* Programmers.

They can all use the same toolset to collaborate with each other and improve their efficiency.

Google does not just have any data; they have the world's most massive computer, so TensorFlow was built to scale. TensorFlow is a library developed by the Google Brain Team to accelerate machine learning and deep neural network research.

It was built to run on multiple CPUs or GPUs and even mobile operating systems, and it has several wrappers in several languages like Python, C++ or Java.

In this tutorial, you will learn

**TensorFlow Architecture**

Tensor flow architecture works in three parts:

* Pre processing the data
* Build the model
* Train and estimate the model

It is called Tensor flow because it takes input as a multi-dimensional array, also known as **tensors**. You can construct a sort of **flowchart** of operations (called a Graph) that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output.

This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

**Where can Tensor flow run?**

TensorFlow can hardware, and software requirements can be classified into

Development Phase: This is when you train the mode. Training is usually done on your Desktop or laptop.

Run Phase or Inference Phase: Once training is done Tensorflow can be run on many different platforms. You can run it on

* Desktop running Windows, macOS or Linux
* Cloud as a web service
* Mobile devices like iOS and Android

You can train it on multiple machines then you can run it on a different machine, once you have the trained model.

The model can be trained and used on GPUs as well as CPUs. GPUs were initially designed for video games. In late 2010, Stanford researchers found that GPU was also very good at matrix operations and algebra so that it makes them very fast for doing these kinds of calculations. Deep learning relies on a lot of matrix multiplication. TensorFlow is very fast at computing the matrix multiplication because it is written in C++. Although it is implemented in C++, TensorFlow can be accessed and controlled by other languages mainly, Python.

Finally, a significant feature of Tensor Flow is the Tensor Board. The Tensor Board enables to monitor graphically and visually what TensorFlow is doing.

**List of Prominent Algorithms supported by TensorFlow**

* Linear regression: tf. estimator .Linear Regressor
* Classification :tf. Estimator .Linear Classifier
* Deep learning classification: tf. estimator. DNN Classifier
* Booster tree regression: tf.estimator.BoostedTreesRegressor
* Boosted tree classification: tf.estimator.BoostedTreesClassifier

**MODULES**

1. **DATA COLLECTION**
2. **DATA PRE-PROCESSING**
3. **FEATURE EXTRATION**
4. **EVALUATION MODEL**
5. **STREAMLIT**

**DATA COLLECTION**

Data collection is a process in which information is gathered from many sources which is later used to develop the machine learning models. The data should be stored in a way that makes sense for problem. In this step the data set is converted into the understandable format which can be fed into machine learning models.

Data used in this paper is a set of data with features . This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called *labelled data*.

**DATA PRE-PROCESSING**

Organize your selected data by formatting, cleaning and sampling from it.

Three common data pre-processing steps are:

* Formatting: The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.
* Cleaning: Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonymized or removed from the data entirely.
* Sampling: There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset.

**FEATURE EXTRATION**

Next thing is to do Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally, our models are trained using Classifier algorithm. We use classify module on Natural Language Toolkit library on Python. We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data. The chosen classifiers were Random forest. These algorithms are very popular in text classification tasks.

**EVALUATION MODEL**

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over fitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. To avoid over fitting, both methods use a test set (not seen by the model) to evaluate model performance.

Performance of each classification model is estimated base on its averaged. The result will be in the visualized form. Representation of classified data in the form of graphs.

**Accuracy** is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

**Feasibility study**

Feasibility study in the sense it's a practical  approach of implementing the proposed model of system . Here for a machine learning projects .we generally collect the input   from online websites and filter the input data and visualize them in graphical format and then the data is divided for training and testing . That training is testing data is given to the   algorithms to predict the data .

1. First, we take dataset.

2. Filter dataset according to requirements and create a new dataset which has attribute according to analysis to be done

3. Perform Pre-Processing on the dataset

4. Split the data into training and testing

5. Train the model with training data then analyze testing dataset over classification algorithm

6. Finally you will get results as accuracy metrics.

**5.STREAMLIT:**

Streamlit is an open-source (free) Python library, which provides a fast way to build interactive web apps. It is a relatively new package but has been growing tremendously. It is designed and built especially for machine learning engineers or other data science professionals. Once you are done with your analysis in Python, you can quickly turn those scripts into web apps/tools to share with others.

As long as you can code in Python, it should be straightforward for you to write Streamlit apps. Imagine the app as the canvas. We can write Python scripts from top to bottom to add all kinds of elements, including text, charts, widgets, tables, etc.

Streamlit is the easiest way especially for people with no front-end knowledge to put their code into a web application:

* No front-end (html, js, css) experience or knowledge is required.
* You don't need to spend days or months to create a web app, you can create a really beautiful machine learning or data science app in only a few hours or even minutes.
* It is compatible with the majority of Python libraries (e.g. pandas, matplotlib, seaborn, plotly, Keras, PyTorch, SymPy(latex)).
* Less code is needed to create amazing web apps.
* Data caching simplifies and speeds up computation pipelines.

**DATA FLOW DIAGRAM**

**LEVEL 0**

LEVEL 1

LEVEL 2

**UML DIAGRAMS**

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

* actors
* business processes
* (logical) components
* activities
* programming language statements
* database schemas, and
* Reusable software components.

UML combines best techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML aims to be a standard modeling language which can model concurrent and distributed systems.

**Sequence Diagram:**

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

* **Activity Diagrams-:** 
  + Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency.In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

**Usecase diagram:**

* UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.
* UML was created by Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.
* OMG is continuously putting effort to make a truly industry standard.
* UML stands for **U**nified **M**odeling **L**anguage.
* UML is a pictorial language used to make software blue prints

**Class diagram**

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

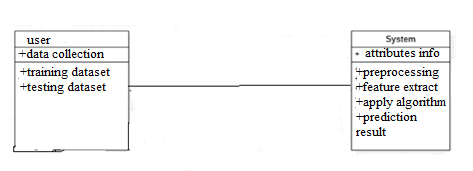
In the diagram, classes are represented with boxes that contain three compartments:

The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.

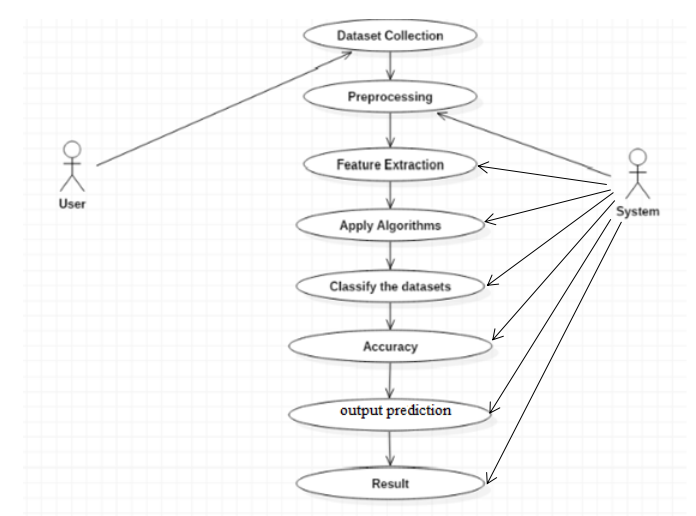
The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.

The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

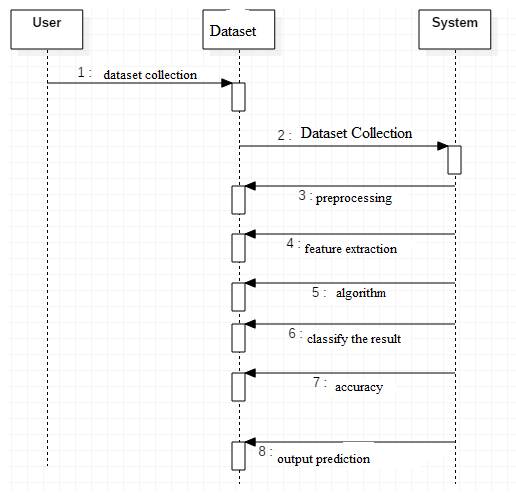
**Use Case Diagram**



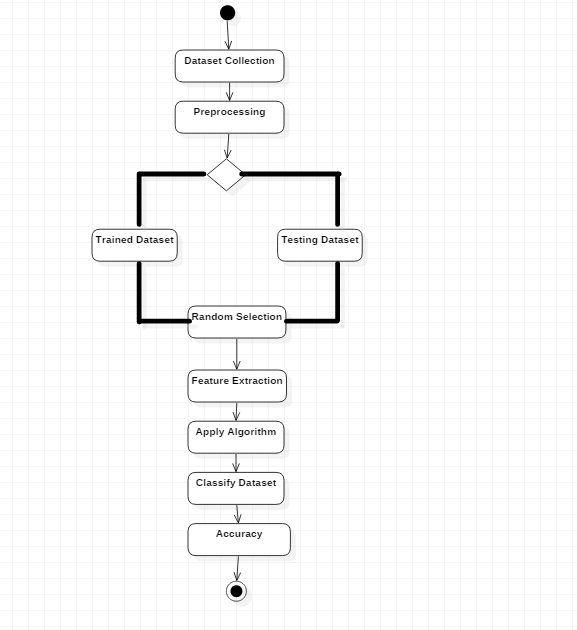
**CLASS DIAGRAM**



SEQUENCE DIAGRAM



**ACTIVITY DIAGRAM**



**PYTHON OVERVIEW**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* **Python is Interpreted:** Python is processed at runtime by the interpreter.You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive:** You can actually sit at a Python prompt and interactwith the interpreter directly to write your programs.
* **Python is Object-Oriented:** Python supports Object-Oriented style ortechnique of programming that encapsulates code within objects.
* **Python is a Beginner's Language:** Python is a great language for thebeginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**History of Python**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, Unix shell, and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

**Python Features**

Python's features include:

**Easy-to-learn:** Python has few keywords, simple structure, and a clearlydefined syntax. This allows the student to pick up the language quickly.

**Easy-to-read:** Python code is more clearly defined and visible to the eyes.

**Easy-to-maintain:** Python's source code is fairly easy-to-maintain.

**A broad standard library:** Python's bulk of the library is very portable andcross-platform compatible on UNIX, Windows, and Macintosh.

**Interactive Mode:** Python has support for an interactive mode which allowsinteractive testing and debugging of snippets of code.

**Portable:** Python can run on a wide variety of hardware platforms and has thesame interface on all platforms.

**Extendable:** You can add low-level modules to the Python interpreter. Thesemodules enable programmers to add to or customize their tools to be more efficient.

**Databases:** Python provides interfaces to all major commercial databases.

**GUI Programming:** Python supports GUI applications that can be created andported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

**Scalable:** Python provides a better structure and support for large programsthan shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below:

* IT supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

**ANACONDA NAVIGATOR**

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, mac OS and Linux.

## Why use Navigator?

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages, and use multiple environments to separate these different versions.

The command line program conda is both a package manager and an environment manager, to help data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages and update them, all inside Navigator.

## **WHAT APPLICATIONS CAN I ACCESS USING NAVIGATOR**?

The following applications are available by default in Navigator:

* Jupyter Lab
* Jupyter Notebook
* QT Console
* Spyder
* VS Code
* Glue viz
* Orange 3 App
* Rodeo
* RStudio

Advanced conda users can also build your own Navigator applications

## How can I run code with Navigator?

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code.

You can also use Jupyter Notebooks the same way. Jupyter Notebooks are an increasingly popular system that combine your code, descriptive text, output, images and interactive interfaces into a single notebook file that is edited, viewed and used in a web browser.

## What’s new in 1.9?

* Add support for **Offline Mode** for all environment related actions.
* Add support for custom configuration of main windows links.
* Numerous bug fixes and performance enhancements.

**TESTING**

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Software Testing can also be stated as the process of validating and verifying that a software program/application/product:

* Meets the business and technical requirements that guided its design and Development.
* Works as expected and can be implemented with the same characteristics.

**TESTING METHODS**

* **Functional Testing**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

* Functions: Identified functions must be exercised.
* Output: Identified classes of software outputs must be exercised.
* Systems/Procedures: system should work properly

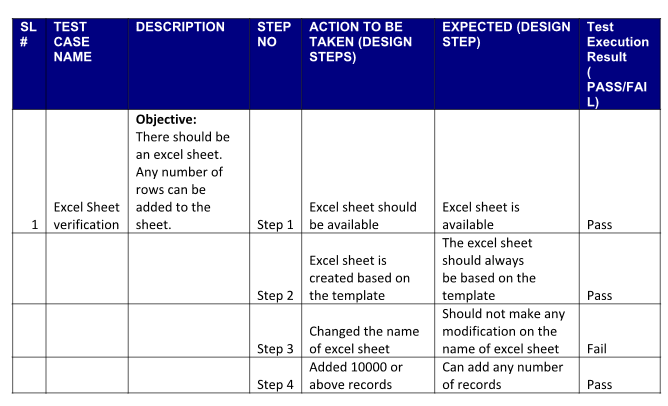
**Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

Test Case for Excel Sheet Verification:

Here in machine learning we are dealing with dataset which is in excel sheet format so if any test case we need means we need to check excel file. Later on classification will work on the respective columns of dataset .

Test Case 1 :



**Results:** (code Screeenshots)

**CONCLUSION:**

Naive Bayes, Decision tree, K-nearest neighbor, Support Vector Machines, and Random Forest are the five well-known data mining classification techniques used in this study to classify water quality as excellent, acceptable, slightly polluted, polluted, and heavily polluted. The Overall Index of Pollution served as the foundation for the models used in each classifier.

The eight possible ranges of water quality parameters were used to create the synthetic data set: temperature, dissolved oxygen (DO), pH, conductivity, biochemical oxygen demand (BOD), nitrates (NO3), feces, and total coli (TC) forms Both domestic and international standards were met by these ranges. The actual data set was derived from the available literature for a number of locations in Tamil Nadu. In the learning stage, the boundaries of every classifier were calibrated to show up at the best boundary settings for

learning a specific water quality class in the informational indexes. During the testing phase, metrics like accuracy, sensitivity, specificity, precision, recall, and F1score were used to assess each predictive model's validity against unseen data. The Radom Forest classifier out of the five available options produces the best results. The level of RF performance was also obtained by the DT classifier**.**

**REFERENCE:**

[1] M. Awais, B. Aslam, A. Maqsoom, U. Khalil, F. Ullah, S. Azam, and

M. Imran, ‘‘Assessing nitrate contamination risks in groundwater: A

machine learning approach,’’ Appl. Sci., vol. 11, no. 21, p. 10034,

Oct. 2021.

[2] T. H. Tulchinsky and E. A. Varavikova, ‘‘Communicable diseases,’’ New

Public Health, San Diego, CA, USA, Tech. Rep., 2014, p. 149.

[3] S. Khalid, M. Shahid, I. Bibi, T. Sarwar, A. H. Shah, and N. K. Niazi,

‘‘A review of environmental contamination and health risk assessment of

wastewater use for crop irrigation with a focus on low and high-income

countries,’’ Int. J. Environ. Res. Public Health, vol. 15, no. 5, p. 895,

May 2018.

[4] E. Chu and J. Karr, ‘‘Environmental impact: Concept, consequences, mea-

surement,’’ Reference Module Life Sci., Elsevier, 2017.

[5] P. M. Kopittke, N. W. Menzies, P. Wang, B. A. McKenna, and E. Lombi,

‘‘Soil and the intensification of agriculture for global food security,’’

Environ. Int., vol. 132, Nov. 2019, Art. no. 105078.

[6] M. Hameed, S. S. Sharqi, Z. M. Yaseen, H. A. Afan, A. Hussain, and

A. Elshafie, ‘‘Application of artificial intelligence (AI) techniques in water

quality index prediction: A case study in tropical region, Malaysia,’’ Neural

Comput. Appl., vol. 28, pp. 893–905, Dec. 2017.

[7] T. Bournaris, J. Papathanasiou, B. Manos, N. Kazakis, and K. Voudouris,

‘‘Support of irrigation water use and eco-friendly decision process in

agricultural production planning,’’ Oper. Res., vol. 15, no. 2, pp. 289–306,

Jul. 2015.

[8] F. Rufino, G. Busico, E. Cuoco, T. H. Darrah, and D. Tedesco, ‘‘Evaluating

the suitability of urban groundwater resources for drinking water and

irrigation purposes: An integrated approach in the Agro-Aversano area of

Southern Italy,’’ Environ. Monitor. Assessment, vol. 191, no. 12, pp. 1–17,

Dec. 2019.

[9] M. Vadiati, A. Asghari-Moghaddam, M. Nakhaei, J. Adamowski, and

A. H. Akbarzadeh, ‘‘A fuzzy-logic based decision-making approach

for identification of groundwater quality based on groundwater quality

indices,’’ J. Environ. Manage., vol. 184, pp. 255–270, Dec. 2016.

[10] A. Shalby, M. Elshemy, and B. A. Zeidan, ‘‘Assessment of climate change

impacts on water quality parameters of lake Burullus, Egypt,’’ Environ. Sci.

Pollut. Res., vol. 27, no. 26, pp. 32157–32178, Sep. 2020.

[11] D. Sharma and A. Kansal, ‘‘Water quality analysis of river Yamuna using

water quality index in the national capital territory, India (2000–2009),’’

Appl. Water Sci., vol. 1, nos. 3–4, pp. 147–157, Dec. 2011.

[12] D. T. Bui, K. Khosravi, J. Tiefenbacher, H. Nguyen, and

N. Kazakis, ‘‘Improving prediction of water quality indices using

novel hybrid machine-learning algorithms,’’ Sci. Total Environ., vol. 721,

Jun. 2020, Art. no. 137612.

[13] Z. M. Yaseen, M. M. Ramal, L. Diop, O. Jaafar, V. Demir, and O. Kisi,

‘‘Hybrid adaptive neuro-fuzzy models for water quality index estimation,’’

Water Resour. Manage., vol. 32, pp. 2227–2245, May 2018.

[14] C. Iticescu, L. P. Georgescu, G. Murariu, C. Topa, M. Timofti, V. Pintilie,

and M. Arseni, ‘‘Lower Danube water quality quantified through WQI and

multivariate analysis,’’ Water, vol. 11, no. 6, p. 1305, Jun. 2019.

[15] W. C. Leong, A. Bahadori, J. Zhang, and Z. Ahmad, ‘‘Prediction of water

quality index (WQI) using support vector machine (SVM) and least square-

support vector machine (LS-SVM),’’ Int. J. River Basin Manage., vol. 19,

no. 2, pp. 149–156, Apr. 2021.

[16] I. H. Sarker, ‘‘Machine learning: Algorithms, real-world applications and

research directions,’’ Social Netw. Comput. Sci., vol. 2, no. 3, pp. 1–21,

May 2021.

[17] M. J. Alizadeh, M. R. Kavianpour, M. Danesh, J. Adolf, S. Shamshirband,

and K.-W. Chau, ‘‘Effect of river flow on the quality of estuarine and

coastal waters using machine learning models,’’ Eng. Appl. Comput. Fluid

Mech., vol. 12, no. 1, pp. 810–823, Jan. 2018.

[18] K. Kargar, S. Samadianfard, J. Parsa, N. Nabipour, S. Shamshirband,

A. Mosavi, and K.-W. Chau, ‘‘Estimating longitudinal dispersion coef-

ficient in natural streams using empirical models and machine learning

algorithms,’’ Eng. Appl. Comput. Fluid Mech., vol. 14, no. 1, pp. 311–322,

Jan. 2020.

[19] T. M. Tung and Z. M. Yaseen, ‘‘A survey on river water quality modelling

using artificial intelligence models: 2000–2020,’’ J. Hydrol., vol. 585,

Jun. 2020, Art. no. 124670.

[20] K. Khosravi, L. Mao, O. Kisi, Z. M. Yaseen, and S. Shahid, ‘‘Quantifying

hourly suspended sediment load using data mining models: Case study of a

glacierized Andean catchment in Chile,’’ J. Hydrol., vol. 567, pp. 165–179,

Dec. 2018.