**A Major Project Report On**

# WATERNET: A NETWORK FOR MONITORING AND ACCESSING WATER QUALITY FOR DRINKING AND IRRIGATION PURPOSES

### Submitted in Partial Fulfillment of the Academic Requirement for the Award Degree of

**BACHELOR OF TECHNOLOGY**

### in

**Computer Science and Engineering**

Submitted By:

**ETIKALA SRIKANTH 20R01A0580**

**VADDE UDAYSAI 20R01A05B5**

**VELICHETI REVANTHSAI 20R01A05B8**

### Under The Esteemed Guidance of Ms. Ashwini S

**(Assistant Professor, CSE Department)**



CMR INSTITUTE OF TECHNOLOGY

**(UGC AUTONOMOUS)**

### Approved by AICTE, Affiliated to JNTUH, Accredited by NBA and NAAC Kandlakoya(V), Medchal Dist - 501 401

[**www.cmritonline.ac.in**](http://www.cmritonline.ac.in/)

**2023-2024**

# CMR INSTITUTE OF TECHNOLOGY

**UGC AUTONOMOUS**

***(Approved by AICTE, Permanently Affiliated to JNTUH, Hyderabad, Accredited by NBA and NAAC with ‘A’ Grade)***

**Kandlakoya (V), Medchal Road, Hyderabad – 501 401**

# CERTIFICATE

This is to certify that the project report titled **WATERNET: A Network for Monitoring and Assessing Water Quality for Drinking and Irrigation Purposes** is being submitted by

**ETIKALA SRIKANTH 20R01A0580**

**VADDE UDAYSAI 20R01A05B5**

**VELICHETI REVANTHSAI 20R01A05B8**

To JNTUH, Hyderabad, in partial fulfillment of the requirement for award of the degree of B.Tech in CSE and is a record of a bonafide work carried out under our guidance and supervision. The results in this project have been verified and are found to be satisfactory. The results embodied in this work have not been submitted to have any other University for award of any other degree or diploma.

|  |  |  |
| --- | --- | --- |
| **Signature of Guide** | **Signature of Coordinator** | **Signature of HOD** |
| **Ms. S. Ashwini** | **Ms. S. Kalyani** | **Mr. A. Prakash** |
| **Assistant professor** | **Assistant professor** | **Head of the Department** |

**Signature of EXTERNAL EXAMINER**

# DECLARATION

We, the below students of Department of Computer Science and Engineering, CMR Institute of Technology declares that the titled " **WATERNET: A Network for Monitoring and Assessing Water Quality for Drinking and Irrigation Purposes** " is a bonafide work carried out by us. Further, we declare that this has not formed the basis of award of any Degree, Diploma, Associate-ship or other similar degree or diploma and has not been submitted anywhere else.

## NAMES ROLL NO.

**ETIKALA SRIKANTH 20R01A0580**

**VADDE UDAYSAI 20R01A05B5**

**VELICHETI REVANTHSAI 20R01A05B8**

# ACKNOWLEDGEMENT

We are extremely grateful to **Dr. M. Janga Reddy**, **Director**, **Dr. B. Satyanarayana**, **Principal** and **Mr. A. Prakash**, **Head of Department**, Dept of Computer Science and Engineering, CMR Institute of Technology for their inspiration and valuable guidance during entire duration.

We are extremely thankful to our internal guide **Ms. S. Ashwini**, Dept of Computer Science and Engineering, CMR Institute of Technology for her constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this Project successfully in time.

Finally, we are very much thankful to our parents and relatives who guided directly or indirectly for every step towards success.

**ETIKALA SRIKANTH** **20R01A0580**

**VADDE UDAYSAI 20R01A05B5**

**VELICHETI REVANTH SAI 20R01A05B8**

# ABSTRACT

Water is a fundamental requirement for human, animal, and plant survival. Despite its importance and quality, water is not always suitable for drinking, domestic and/or industrial use. Numerous factors such as industrialization, mining, pollution, and natural occurrences impact the quality of water, as they introduce or alter various parameters present therein, thus, affecting its suitability for human consumption or general use. The World Health Organization has guidelines which stipulate the threshold levels of various parameters present in water samples intended for consumption or irrigation. The Water Quality Index (WQI) and Irrigation WQI (IWQI) are metrics used to express the level of these parameters to determine the overall water quality. Collecting water samples from different sources, measuring the various parameters present, and bench- marking these measurements against pre-set standards, while adhering to various guidelines during transportation and measurement can be extremely daunting. To this end this study proposes a network architecture to collect data on water parameters in real-time and use Machine Learning (ML) tools to automatically determine suitability of water samples for drinking and irrigation purposes. The developed monitoring network is based on LoRa and takes the land topology into consideration. Results of simulations done in Radio Mobile revealed a partial mesh network topology as the most adequate. Due to the absence of large and open datasets on drinking and irrigation water, datasets usable for training ML models were developed. Three ML models - Random Forest (RF), Logistic Regression (LR) and Support Vector Machine (SVM) were considered for the water classification process and results obtained showed that LR performed best for drinking water, while SVM was better suited for irrigation water. Recursive feature elimination was then combined with the three ML models to reveal which of the water parameters had the greatest influence on the classification accuracies of the respective model.

# INDEX

|  |  |
| --- | --- |
| **CONTENT** | **PAGE NO** |
| **1. INTRODUCTION** | **1** |
| 1.1 ABOUT PROJECT | **1** |
| **2. SYSTEM ANALYSIS** | **2** |
| 2.1 EXISTING SYSTEM | **2** |
| 2.2 DISADVANTAGES OF EXISTING SYSTEM | **3** |
| 2.3 PROPOSED SYSTEM | **4** |
| 2.4 ADVANTAGES OF PROPOSED SYSTEM | **4** |
| **3. SYSTEM STUDY** | **5** |
| 3.1 FEASIBILITY STUDY | **5** |
| **4. HARDWARE AND SOFTWARE REQUIREMENTS** | **7** |
| 4.1 HARDWARE REQUIREMENTS | **7** |
| 4.2 SOFTWARE REQUIREMENTS | **7** |
| **5. ARCHITECTURE** | **8** |
| **6. MODULES** | **9** |
| 6.1 SERVICE PROVIDER | **9** |
| 6.2 VIEW AND AUTHORIZE | **9** |
| 6.3 REMOTE USER | **9** |
| **7. DIAGRAMS** | **10** |
| 7.1 DATA FLOW DIAGRAM | **10** |
| 7.2 USE CASE DIAGRAM | **11** |
| 7.3 CLASS DIAGRAM | **12** |
| 7.4 SEQUENCE DIAGRAM | **13** |
| 7.5 FLOW CHART DIAGRAM | **14** |
| **8. IMPLEMENTATION** | **16** |
| 8.1 ALGORITHMS | **16** |
| 8.2 SOURCE CODE | **20** |
| **9. SCREEN SHOTS** | **32** |
| 9.1 REMOTE USER | **32** |
| 9.2 SERVICE PROVIDER | **34** |
| **10. TESTING** | **39** |
| **11. CONCLUSION** | **41** |
| **12. REFERENCES** | **42** |

**LIST Of FIGURES**

|  |  |  |
| --- | --- | --- |
| FIGURE NO | PARTICULARS | PAGE NO |
| 5.1 | Architecture Diagram | 8 |
| 7.1 | Data Flow Diagram | 10 |
| 7.2 | Use Case Diagram | 11 |
| 7.3 | Class Diagram | 12 |
| 7.4 | Sequence Diagram | 13 |
| 7.5.1 | Flow Chart Diagram- Remote User | 14 |
| 7.5.2 | Flow Chart Diagram- Service Provider | 15 |
| 9.1.1 | Remote User Login Page | 32 |
| 9.1.2 | Remote User Registration Page | 32 |
| 9.1.3 | Remote User Water Quality Prediction Form | 33 |
| 9.1.4 | Remote User Profile Page | 33 |
| 9.2.1 | Service Provider Login Page | 34 |
| 9.2.2 | Accuracy According to Algorithms | 34 |
| 9.2.3 | Bar Chart for Accuracy Visualization | 35 |
| 9.2.4 | Line Chart for Accuracy Visualization | 35 |
| 9.2.5 | Pie Chart for Accuracy Visualization | 36 |
| 9.2.6 | View Predicted Water Quality Detection type | 36 |
| 9.2.7 | Water Quality Detection Type Ratio Details | 37 |
| 9.2.8 | Water Quality Detection Ratio Results in Line Chart | 37 |
| 9.2.9 | Water Quality Detection Ratio Results in Pie Chart | 38 |
| 9.2.10 | List of Remote Users | 38 |

# INTRODUCTION

## 1.1 ABOUT PROJECT

Access to water is a critical component of human lives and is now considered a basic human right. Access to clean water is also one of the 17 Sustainable Development Goals (SDG) set up by the United Nations in 2015 to achieve a better future for all [1].Specifically, the sixth goal, which is to ensure and sustain the availability of water and sanitation to all [2]. Potable water can also be linked to the third SDG goal \_ good health and well-being, as contaminated water can be a transmission medium for diseases such as cholera, typhoid, and diarrhoea, which are jointly the highest cause of mortality (especially children) in developing nations of Africa and Asia [3]. Water is also important in agriculture and food production. Recent statistics shows that about 10% of the world population is malnourished, with developing countries being hit the hardest, with starvation resulting in about 45% of infant mortality [5]. Ensuring global food security is thus of utmost importance. Food security has been recognized as a critical requirement, hence its inclusion as one of the SDG (goal 2), with specific focus on ending hunger, by promoting sustainable agriculture and improving food distribution. Food production and agriculture in general rely heavily on water, both for irrigation and for animal consumption. It is thus pertinent to ensure the availability and sustainable management of water \_t for agricultural use.

There are several sources of water for both drinking and irrigation use, including rivers, streams, rain, and groundwater (accessed through wells and boreholes). The nature and characteristics of a source of water are often critical factors that influence the constituents of water samples obtained therein. Beyond natural factors, chemical wastes from human activities such as mining, crude oil extraction, and industrial wastes, most often end up in streams, rivers, and other sources of water, changing the nature and properties of these waters. These waters then end up in homes or farms, where they are used for domestic purposes, drank, fed to livestock, or used to water crops. Consuming this type of water can have dire health consequences or result in death. It is therefore paramount that a proper process be put in place to ensure end to- end monitoring of the water right from the source to its last point of use. At each monitoring point, samples of water need to be collected to assess the quality or ``fitness for use'' for human (and animal) consumption, irrigation and domestic (or industrial)uses.

# SYSTEM ANALYSIS

## EXISTING SYSTEM

In a network for measuring and monitoring water parameters in a metal producing city in Brazil was developed. Twelve water monitoring stations were setup to measure several physico-chemical water parameters, including pH, dissolved solids, Zinc, Lead etc. Finally, obtained results were analysed using principal component analysis. In a similar manner, [ developed a system to monitor water quality in Limpopo River Basin in Mozambique and set up 23 monitoring stations to measure physico- chemical and microbiological parameters, and ultimately assess the quality of water in the river basin. To address the challenges of optimal placement of gauges and sampling frequencies, which are often faced when developing water monitoring systems, the authors in [14] developed an economically viable model that combined genetic algorithm with 1-D water quality simulation. Though the work was only simulated by using genetic algorithm, the authors were able to solve the NP hard problem of optimally placing monitoring stations.

Monitoring water parameters often entails periodically sampling a body of water to capture relevant metrics. These metrics might include physico-chemical and microbiological measurements, such as potential of hydrogen (pH), temperature, sodium levels etc. In a water monitoring network, measured parameters need to be transferred to a base station where relevant decision(s) would be taken. Due to the sparse nature of transmitted data, light weight communication protocols capable of transmitting relatively small data over long distance are required for water monitoring networks. From literature, Low Power Wide Area Network (LPWAN) technologies have been favoured for such applications. An extensive discussion on LPWAN technologies was done in [19]. The work compared a few sub-GHz solutions including Sig- Fox, LoRa, Ingenu and Telensa, with respect to their range,

transmission rate, and channel count. Ingenu was reported to have the longest range in city settings at 15 km, followed by SigFox at 10 km (in cities) and 50 km (in rural areas); then LoRa at 5 km (in cities), and 15 km in rural settings.

Regarding the assessment of communication technologies, there has been a long-drawn debate over the efficacy of software simulations versus real-world testing. Though this debate still rages, several researchers have shown that simulation results are often at par with real-world tests. For instance, using LoRa, the authors in [20] compared simulation results with real world test for intervehicle communication. They used NS3 as a simulation platform and an Arduino UNO C Dragino LoRa module for the real-world tests, while Propagation loss, coverage Packet Inter-reception (PIR), Packet

Delivery Ratio (PDR) and Received Signal Strength Indicator (RSSI) level were used as benchmark metrics. They concluded that the results of the simulator were consistent with those of the real-world tests. In a similarwork,

Hassan [21] also compared the efficacy of simulation results (from Radio Mobile simulator) with real- world tests (using micro controllers C LoRa modules) when using LoRa as a bridge for Wi-Fi. Unlike [20], [21] did not give a side-byside comparison of simulated vs. real-world results for each metric considered but concluded that the simulator performed well. [22] set up seven pairs of XBee modules and compared communication performance using both the 800/900MHz

and 2.4GHz frequencies. They concluded that simulation results from the Radio Mobile simulator corroborated with those of real-world tests.

## DISADVANTAGES OF EXISTING SYSTEM

* + - An existing methodology doesn’t implement DATA PRE-PROCESSING & LABELLING method.
    - The system not implemented Calculating WQI for Irrigation Water for prediction in the datasets.

## PROPOSED SYSTEM

The water monitoring network proposed in this work is to be deployed in the City of Cape Town in Western Cape, South Africa, with the intention of monitoring water parameters in water storage dams and/or water treatment plants across the city. Data gathered by the monitoring network are then passed through Machine Learning (ML) models to determine their suitability for consumption or irrigation purposes.

1. Build a network for real-time collection and monitoring of water quality across water storage dams in the city of Cape Town. This network takes into consideration the unique geographical features of Cape Town, such as mountains and elevations that might obstruct radio frequency propagation.
2. Curate ample sized datasets on drinking and irrigation water that can be used to train (and test) machine learning models to automatically determine the `fitness for use'' of a sample of water for drinking and/or irrigation purposes.
3. Build models that determine the most critical parameters that influence the accuracy of machine learning models in analyzing water for drinking or irrigation.

## ADVANTAGES OF PROPOSED SYSTEM

* + - The purpose of WaterNet is to gather data on water parameters from dams across the city. These parameters are then used to assess the quality of water with regards ``fitness for use'' for drinking and irrigation purposes.
    - In this work, rather than relying on instrumental and physico-chemical analysis carried out in laboratories to assess water parameters, we propose the use of machine learning (ML) models, which take the numerous water parameters into consideration and automatically determine if a sample of water is potable or fit for agricultural use.

# SYSTEM STUDY

## FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

ECONOMICAL FEASIBILITY TECHNICAL FEASIBILITY SOCIAL FEASIBILITY

### Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

# HARDWARE AND SOFTWARE REQUIREMENTS

## HARDWARE REQUIREMENTS

Processor - Intel core i3 RAM - 4 GB (min)

Hard Disk - 20 GB

## SOFTWARE REQUIREMENTS

Operating system : Windows 11 Coding Language : Python.

Front-End : Python.

Back-End : Django-ORM

Designing : HTML, CSS, JavaScript.

Data Base : MySQL(WAMPServer).

# ARCHITECTURE

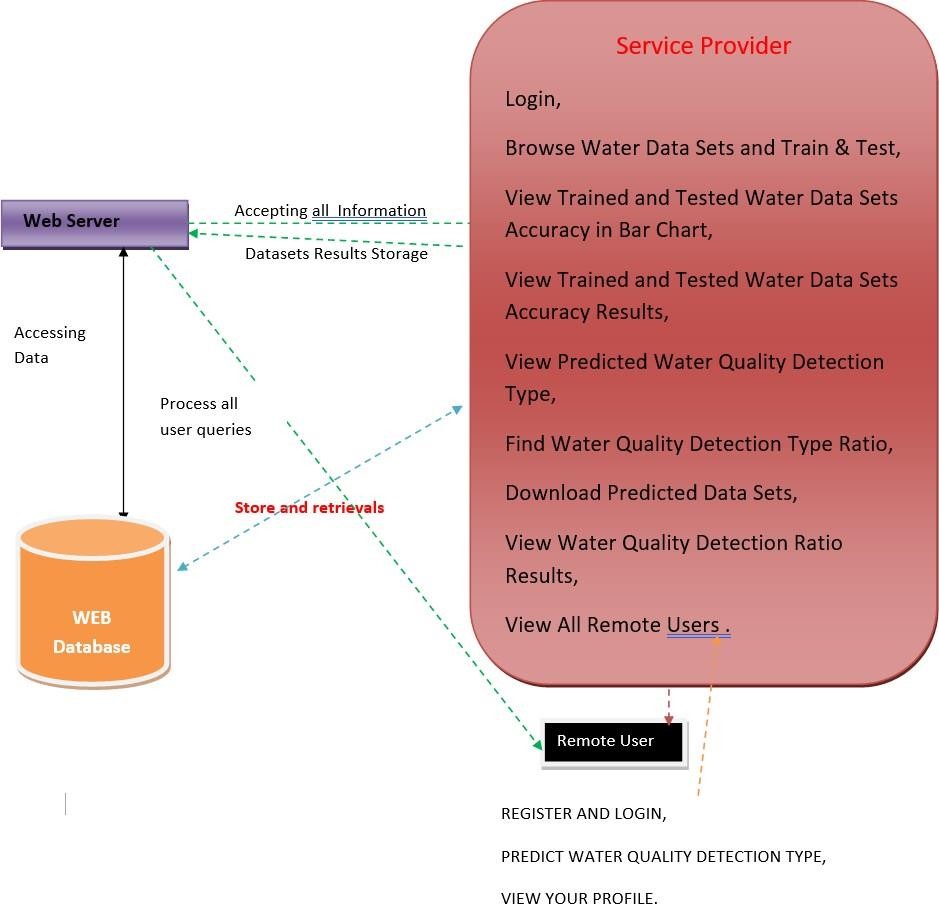


Fig 5.1 Architecture Diagram

# MODULES

## SERVICE PROVIDER

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Login, Browse Water Data Sets and Train & Test, View Trained and Tested Water Data Sets Accuracy in Bar Chart, View Trained and Tested Water Data Sets Accuracy Results, View Predicted Water Quality Detection Type, Find Water Quality Detection Type Ratio, Download Predicted Data Sets, View Water Quality Detection Ratio Results, View All Remote Users.

## VIEW AND AUTHORIZE

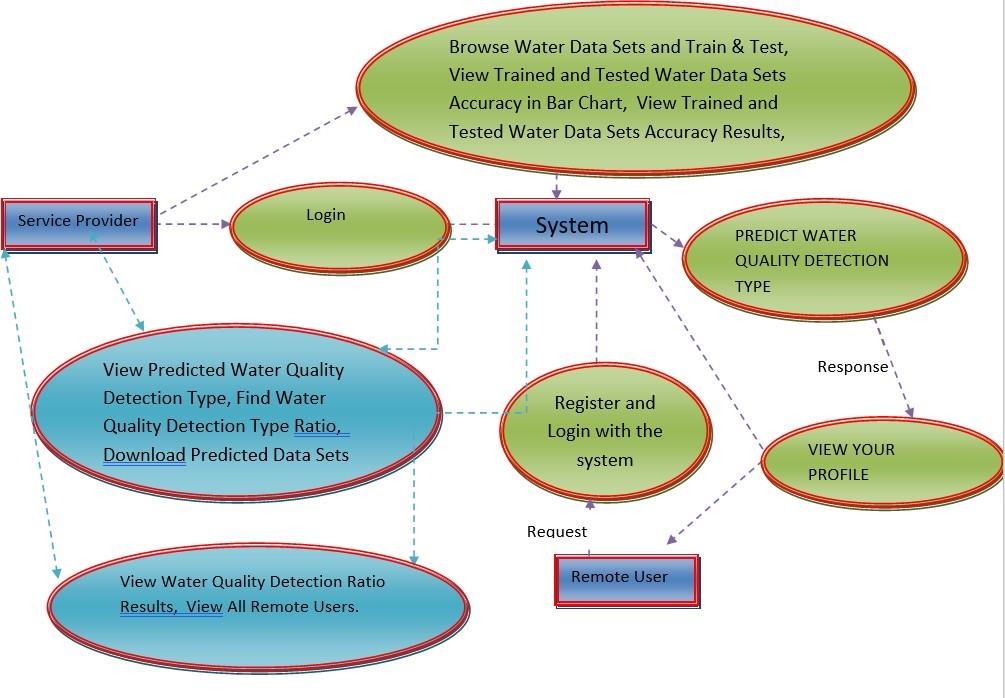
In this module, the admin can view the list of users who all registered. In this, the admin can view the user’s details such as, user name, email, address and admin authorizes the users.

## REMOTE USER

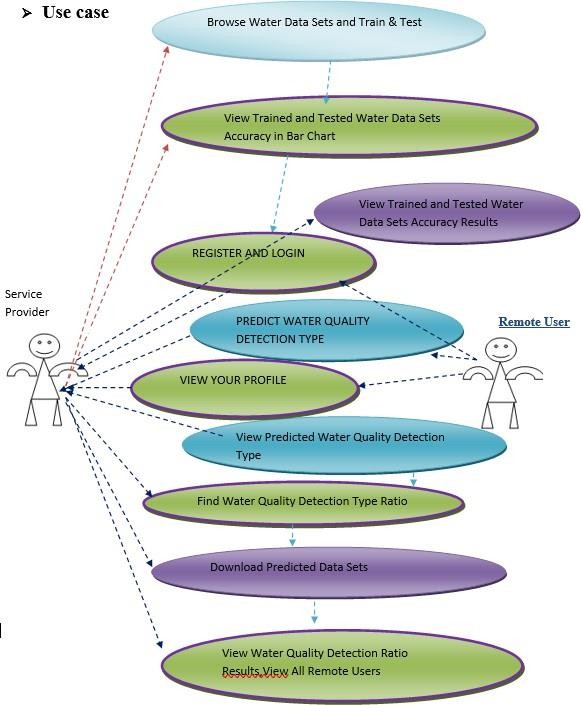
In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN, PREDICT WATER QUALITY DETECTION TYPE, VIEW YOUR PROFILE.

# DIAGRAMS

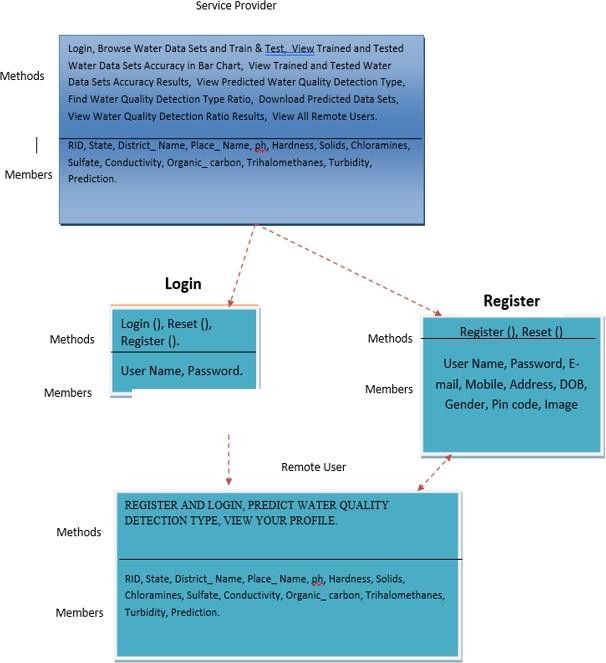
## DATA FLOW DIAGRAM



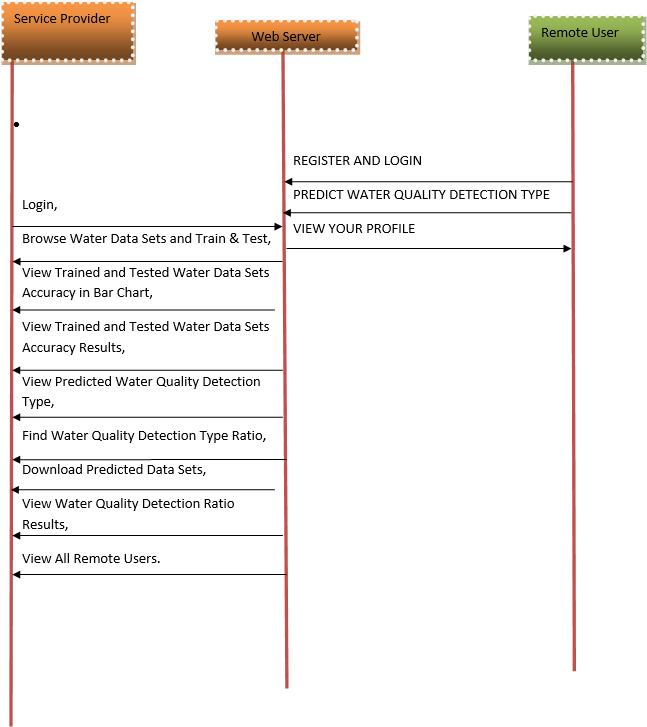
* 1. **USE CASE DIAGRAM**



## CLASS DIAGRAM

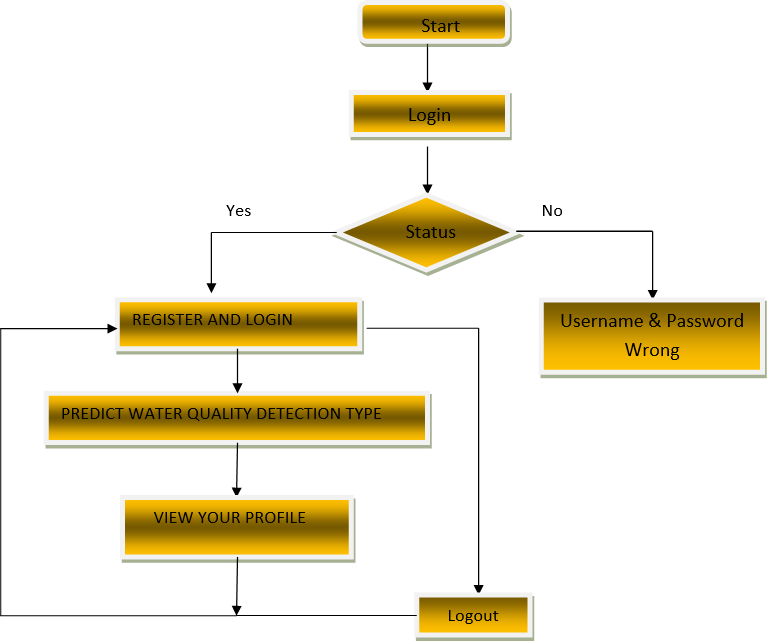


* 1. **SEQUENCE DIAGRAM**

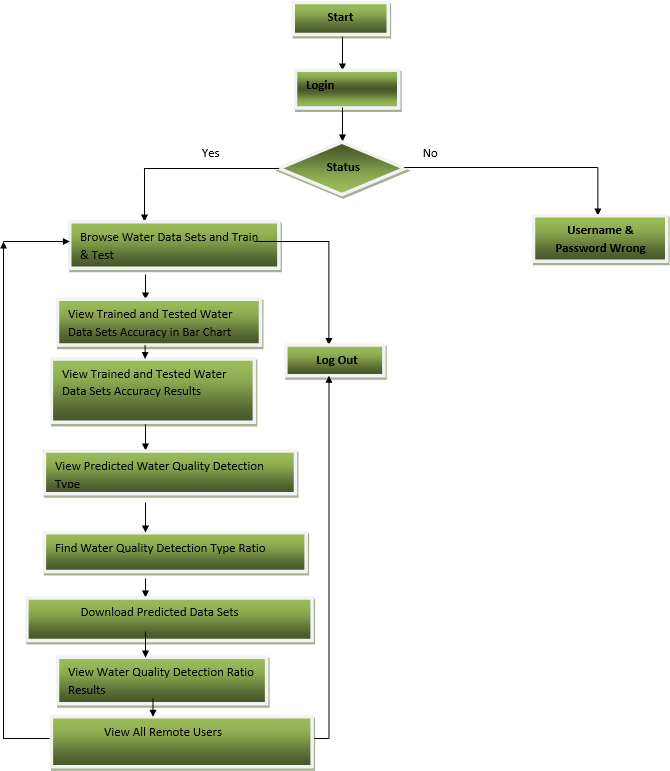


## FLOW CHART DIAGRAM

### Remote User



* + 1. **Service Provider**



# IMPLEMENTATION

## ALGORITHMS

### Decision tree classifiers

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision-making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such

is as follows:

Step 1. If all the objects in S belong to the same class, for example Ci, the decision tree for S consists of a leaf labeled with this class

Si has outcome Oi for T. T becomes the root of the decision tree and for each outcome Oi we build a subsidiary decision tree by invoking the same procedure recursively on the set Si.

### Gradient boosting

**Gradient boosting** is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms forest. A gradient- boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

### K-Nearest Neighbors (KNN)

Simple, but a very powerful classification algorithm Classifies based on a similarity measure.

Non-parametric Lazy learning



Whenever we have a new data to classify, we find its K-nearest neighbors from the training data.

Example

Training dataset consists of k-closest examples in feature space.

Feature space means, space with categorization variables (non-metric variables) Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset.

### Logistic regression Classifiers

*Logistic regression analysis* studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name *logistic regression* is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name *multinomial logistic regression* is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis.

### Naïve Bayes

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature .

Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias).

While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain amodel easy to interpret and deploy, they do not understand the interest of such a technique.

Thus, we introduce in a new presentation the results of the learning process. The classifier is easier to understand, and its deployment is also made easier. In the first part of this tutorial, we present some theoretical aspects of the naive bayes classifier. Then, we implement the approach on a dataset with Tanagra. We compare the obtained results (the parameters of the model) to those obtained with other linear approaches such as the logistic regression, the linear discriminant analysis and the linear SVM. We note that the results are highly consistent. This largely explains the good performance of the method in comparison to others. In the second part, we use various tools on the same dataset (**Weka 3.6.0**, **R 2.9.2**, **Knime 2.1.1**, **Orange 2.0b** and **RapidMiner 4.6.0)**. We try above all to understand the obtained results.

### Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random

decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

The first algorithm for random decision forests was created in 1995 by Tin Kam Ho[1] using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc.).The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho[1] and later independently by Amit and Geman[13] in order to construct a collection of decision trees with controlled variance.

Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

### SVM

In classification tasks a discriminant machine learning technique aims at finding, based on an *independent and identically distributed* (*iid*) training dataset, a discriminant function that can correctly predict labels for newly acquired instances. Unlike generative machine learning approaches, which require computations of conditional probability distributions, a discriminant classification function takes a data point *x* and assigns it to one of the different classes that are a part of the classification task. Less powerful than generative approaches, which are mostly used when prediction involves outlier detection, discriminant approaches require fewer computational resources and less training data, especially for a multidimensional feature space and when only posterior probabilities are needed. From a geometric perspective, learning a classifier is equivalent to finding the equation for a multidimensional surface that best separates the different classes in the feature space.

SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane paramet~~er~~ in contrast to *genetic algorithms* (*GAs*) or *perceptrons*, both of which are widely used for classification in machine learning. For perceptrons, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the

feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized. The aim of GAs and perceptrons is only to minimize error during training, which



## SOURCE CODE

### Manage.py

#!/usr/bin/env python

"""Django's command-line utility for administrative tasks.""" import os

import sys def main():

"""Run administrative tasks.""" os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'waternet.settings') try:

from django.core.management import execute\_from\_command\_line except ImportError as exc:

raise ImportError(

"Couldn't import Django. Are you sure it's installed and "

"available on your PYTHONPATH environment variable? Did you " "forget to activate a virtual environment?"

) from exc execute\_from\_command\_line(sys.argv)

if name == ' main ':

main()

if name == ' main ': main()

### Remote User

**Views.py**

from django.db.models import Count from django.db.models import Q

from django.shortcuts import render, redirect, get\_object\_or\_404 import datetime

import openpyxl

import re import string

import pandas as pd

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report from sklearn.metrics import accuracy\_score

from sklearn.metrics import f1\_score

from sklearn.tree import DecisionTreeClassifier from sklearn.ensemble import VotingClassifier

# Create your views here.

from Remote\_User.models import ClientRegister\_Model,assessing\_water\_quality,detection\_ratio,detection\_accuracy

def login(request):

if request.method == "POST" and 'submit1' in request.POST: username = request.POST.get('username')

password = request.POST.get('password') try:

enter = ClientRegister\_Model.objects.get(username=username,password=password) request.session["userid"] = enter.id

return redirect('ViewYourProfile') except:

pass

return render(request,'RUser/login.html') def Register1(request):

if request.method == "POST":

username = request.POST.get('username') email = request.POST.get('email') password = request.POST.get('password') phoneno = request.POST.get('phoneno') country = request.POST.get('country') state = request.POST.get('state')

city = request.POST.get('city') address = request.POST.get('address')

gender = request.POST.get('gender') ClientRegister\_Model.objects.create(username=username, email=email, password=password,

phoneno=phoneno,

country=country, state=state, city=city, address=address, gender=gender) obj = "Registered Successfully"

return render(request, 'RUser/Register1.html', {'object': obj}) else:

return render(request,'RUser/Register1.html')

def ViewYourProfile(request): userid = request.session['userid']

obj = ClientRegister\_Model.objects.get(id= userid)

return render(request,'RUser/ViewYourProfile.html',{'object':obj})

def Prediction\_Water\_Quality\_Detection(request): if request.method == "POST":

if request.method == "POST":

RID= request.POST.get('RID') State= request.POST.get('State')

District\_Name= request.POST.get('District\_Name') Place\_Name= request.POST.get('Place\_Name') ph= request.POST.get('ph')

Hardness= request.POST.get('Hardness') Solids= request.POST.get('Solids') Chloramines= request.POST.get('Chloramines') Sulfate= request.POST.get('Sulfate') Conductivity= request.POST.get('Conductivity')

Organic\_carbon= request.POST.get('Organic\_carbon') Trihalomethanes= request.POST.get('Trihalomethanes') Turbidity= request.POST.get('Turbidity')

data = pd.read\_csv("water\_datasets.csv") def apply\_results(results):

if (results == 0):

return 0 # Irrigation Water elif (results == 1):

return 1 # Drinking Water

data['Label'] = data['Potability'].apply(apply\_results) x = data['Place\_Name']

y = data['Label']

cv = CountVectorizer() x = cv.fit\_transform(x) models = []

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.20)

X\_train.shape, X\_test.shape, y\_train.shape print("Naive Bayes")

from sklearn.naive\_bayes import MultinomialNB

NB = MultinomialNB() NB.fit(X\_train, y\_train) predict\_nb = NB.predict(X\_test)

naivebayes = accuracy\_score(y\_test, predict\_nb) \* 100 print(naivebayes)

print(confusion\_matrix(y\_test, predict\_nb)) print(classification\_report(y\_test, predict\_nb)) models.append(('naive\_bayes', NB))

# SVM Model print("SVM")

from sklearn import svm

lin\_clf = svm.LinearSVC() lin\_clf.fit(X\_train, y\_train) predict\_svm = lin\_clf.predict(X\_test)

svm\_acc = accuracy\_score(y\_test, predict\_svm) \* 100 print(svm\_acc)

print("CLASSIFICATION REPORT")

print(classification\_report(y\_test, predict\_svm)) print("CONFUSION MATRIX")

print(confusion\_matrix(y\_test, predict\_svm)) models.append(('svm', lin\_clf))

print("Logistic Regression")

from sklearn.linear\_model import LogisticRegression

reg = LogisticRegression(random\_state=0, solver='lbfgs').fit(X\_train, y\_train) y\_pred = reg.predict(X\_test)

print("ACCURACY")

print(accuracy\_score(y\_test, y\_pred) \* 100) print("CLASSIFICATION REPORT")

print(classification\_report(y\_test, y\_pred)) print("CONFUSION MATRIX")

print(confusion\_matrix(y\_test, y\_pred)) models.append(('logistic', reg))

print("Decision Tree Classifier") dtc = DecisionTreeClassifier() dtc.fit(X\_train, y\_train) dtcpredict = dtc.predict(X\_test) print("ACCURACY")

print(accuracy\_score(y\_test, dtcpredict) \* 100)

print("CLASSIFICATION REPORT")

print(classification\_report(y\_test, dtcpredict)) print("CONFUSION MATRIX")

print(confusion\_matrix(y\_test, dtcpredict)) models.append(('DecisionTreeClassifier', dtc))

print("KNeighborsClassifier")

from sklearn.neighbors import KNeighborsClassifier kn = KNeighborsClassifier()

kn.fit(X\_train, y\_train) knpredict = kn.predict(X\_test) print("ACCURACY")

print(accuracy\_score(y\_test, knpredict) \* 100) print("CLASSIFICATION REPORT")

print(classification\_report(y\_test, knpredict)) print("CONFUSION MATRIX")

print(confusion\_matrix(y\_test, knpredict)) models.append(('KNeighborsClassifier', kn))

print("SGD Classifier")

from sklearn.linear\_model import SGDClassifier

sgd\_clf = SGDClassifier(loss='hinge', penalty='l2', random\_state=0) sgd\_clf.fit(X\_train, y\_train)

sgdpredict = sgd\_clf.predict(X\_test) print("ACCURACY")

print(accuracy\_score(y\_test, sgdpredict) \* 100) print("CLASSIFICATION REPORT")

print(classification\_report(y\_test, sgdpredict)) print("CONFUSION MATRIX")

print(confusion\_matrix(y\_test, sgdpredict)) models.append(('SGDClassifier', sgd\_clf))

classifier = VotingClassifier(models) classifier.fit(X\_train, y\_train)

y\_pred = classifier.predict(X\_test)

Place\_Name1 = [Place\_Name]

vector1 = cv.transform(Place\_Name1).toarray() predict\_text = classifier.predict(vector1)

pred = str(predict\_text).replace("[", "") pred1 = pred.replace("]", "")

prediction = int(pred1) if prediction == 0:

val = 'IRRIGATION WATER'

elif prediction == 1:

val = 'DRINKING WATER'

print(prediction) print(val)

assessing\_water\_quality.objects.create( RID=RID,

State=State, District\_Name=District\_Name, Place\_Name=Place\_Name, ph=ph,

Hardness=Hardness, Solids=Solids, Chloramines=Chloramines, Sulfate=Sulfate, Conductivity=Conductivity, Organic\_carbon=Organic\_carbon, Trihalomethanes=Trihalomethanes, Turbidity=Turbidity, Prediction=val)

return render(request, 'RUser/Prediction\_Water\_Quality\_Detection.html',{'objs': val}) return render(request, 'RUser/Prediction\_Water\_Quality\_Detection.html')

### Models.py

from django.db import models

# Create your models here.

from django.db.models import CASCADE

class ClientRegister\_Model(models.Model): username = models.CharField(max\_length=30) email = models.EmailField(max\_length=30) password = models.CharField(max\_length=10) phoneno = models.CharField(max\_length=10) country = models.CharField(max\_length=30) state = models.CharField(max\_length=30)

city = models.CharField(max\_length=30) address = models.CharField(max\_length=3000) gender = models.CharField(max\_length=300)

class assessing\_water\_quality(models.Model):

RID= models.CharField(max\_length=3000) State= models.CharField(max\_length=3000)

District\_Name= models.CharField(max\_length=3000) Place\_Name= models.CharField(max\_length=3000) ph= models.CharField(max\_length=3000)

Hardness= models.CharField(max\_length=3000) Solids= models.CharField(max\_length=3000) Chloramines= models.CharField(max\_length=3000)

Sulfate= models.CharField(max\_length=3000) Conductivity= models.CharField(max\_length=3000) Organic\_carbon= models.CharField(max\_length=3000) Trihalomethanes= models.CharField(max\_length=3000) Turbidity= models.CharField(max\_length=3000) Prediction= models.CharField(max\_length=3000)

class detection\_accuracy(models.Model):

names = models.CharField(max\_length=300) ratio = models.CharField(max\_length=300)

class detection\_ratio(models.Model):

names = models.CharField(max\_length=300) ratio = models.CharField(max\_length=300)

### Service Provider View.py

from django.db.models import Count, Avg from django.shortcuts import render, redirect from django.db.models import Count

from django.db.models import Q import datetime

import xlwt

from django.http import HttpResponse

import nltk import re import string

from nltk.corpus import stopwords

from sklearn.feature\_extraction.text import CountVectorizer from nltk.stem.wordnet import WordNetLemmatizer

import pandas as pd

from wordcloud import WordCloud, STOPWORDS

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report from sklearn.metrics import accuracy\_score

from sklearn.metrics import f1\_score

from sklearn.tree import DecisionTreeClassifier from sklearn.ensemble import VotingClassifier # Create your views here.

from Remote\_User.models import ClientRegister\_Model,assessing\_water\_quality,detection\_ratio,detection\_accuracy

def serviceproviderlogin(request): if request.method == "POST":

admin = request.POST.get('username') password = request.POST.get('password')

if admin == "Admin" and password =="Admin": return redirect('View\_Remote\_Users')

return render(request,'SProvider/serviceproviderlogin.html')

def Find\_Predicted\_Water\_Quality\_Detection\_Type\_Ratio(request): detection\_ratio.objects.all().delete()

ratio = ""

kword = 'IRRIGATION WATER'

print(kword)

obj = assessing\_water\_quality.objects.all().filter(Q(Prediction=kword)) obj1 = assessing\_water\_quality.objects.all()

count = obj.count(); count1 = obj1.count();

ratio = (count / count1) \* 100 if ratio != 0:

detection\_ratio.objects.create(names=kword, ratio=ratio)

ratio1 = ""

kword1 = 'DRINKING WATER'

print(kword1)

obj1 = assessing\_water\_quality.objects.all().filter(Q(Prediction=kword1)) obj11 = assessing\_water\_quality.objects.all()

count1 = obj1.count(); count11 = obj11.count();

ratio1 = (count1 / count11) \* 100 if ratio1 != 0:

detection\_ratio.objects.create(names=kword1, ratio=ratio1)

obj = detection\_ratio.objects.all()

return render(request, 'SProvider/Find\_Predicted\_Water\_Quality\_Detection\_Type\_Ratio.html',

{'objs': obj})

def View\_Remote\_Users(request): obj=ClientRegister\_Model.objects.all()

return render(request,'SProvider/View\_Remote\_Users.html',{'objects':obj})

def ViewTrendings(request): topic =

assessing\_water\_quality.objects.values('topics').annotate(dcount=Count('topics')).order\_by('-dcount') return render(request,'SProvider/ViewTrendings.html',{'objects':topic})

def charts(request,chart\_type):

chart1 = detection\_ratio.objects.values('names').annotate(dcount=Avg('ratio'))

return render(request,"SProvider/charts.html", {'form':chart1, 'chart\_type':chart\_type})

def charts1(request,chart\_type):

chart1 = detection\_accuracy.objects.values('names').annotate(dcount=Avg('ratio'))

return render(request,"SProvider/charts1.html", {'form':chart1, 'chart\_type':chart\_type})

def View\_Predicted\_Water\_Quality\_Detection\_Type(request): obj =assessing\_water\_quality.objects.all()

return render(request, 'SProvider/View\_Predicted\_Water\_Quality\_Detection\_Type.html',

{'list\_objects': obj})

def likeschart(request,like\_chart):

charts =detection\_accuracy.objects.values('names').annotate(dcount=Avg('ratio'))

return render(request,"SProvider/likeschart.html", {'form':charts, 'like\_chart':like\_chart})

def Download\_Predicted\_DataSets(request):

response = HttpResponse(content\_type='application/ms-excel') # decide file name

response['Content-Disposition'] = 'attachment; filename="Predicted\_Data.xls"' # creating workbook

wb = xlwt.Workbook(encoding='utf-8') # adding sheet

ws = wb.add\_sheet("sheet1") # Sheet header, first row row\_num = 0

font\_style = xlwt.XFStyle() # headers are bold font\_style.font.bold = True

# writer = csv.writer(response)

obj = assessing\_water\_quality.objects.all() data = obj # dummy method to fetch data. for my\_row in data:

row\_num = row\_num + 1

ws.write(row\_num, 0, my\_row.RID, font\_style) ws.write(row\_num, 1, my\_row.State, font\_style) ws.write(row\_num, 2, my\_row.District\_Name, font\_style) ws.write(row\_num, 3, my\_row.Place\_Name, font\_style) ws.write(row\_num, 4, my\_row.ph, font\_style) ws.write(row\_num, 5, my\_row.Hardness, font\_style) ws.write(row\_num, 6, my\_row.Solids, font\_style) ws.write(row\_num, 7, my\_row.Chloramines, font\_style) ws.write(row\_num, 8, my\_row.Sulfate, font\_style) ws.write(row\_num, 9, my\_row.Conductivity, font\_style) ws.write(row\_num, 10, my\_row.Organic\_carbon, font\_style) ws.write(row\_num, 11, my\_row.Trihalomethanes, font\_style) ws.write(row\_num, 12, my\_row.Turbidity, font\_style) ws.write(row\_num, 13, my\_row.Prediction, font\_style)

wb.save(response) return response

def Train\_Test\_DataSets(request): detection\_accuracy.objects.all().delete()

data = pd.read\_csv("water\_datasets.csv")

def apply\_results(results): if (results == 0):

return 0 # Irrigation Water elif (results == 1):

return 1 #Drinking Water

data['Label'] = data['Potability'].apply(apply\_results) x = data['Place\_Name']

y = data['Label']

cv = CountVectorizer()

x = cv.fit\_transform(x) models = []

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.20)

X\_train.shape, X\_test.shape, y\_train.shape

print("Naive Bayes")

from sklearn.naive\_bayes import MultinomialNB NB = MultinomialNB()

NB.fit(X\_train, y\_train) predict\_nb = NB.predict(X\_test)

naivebayes = accuracy\_score(y\_test, predict\_nb) \* 100 print(naivebayes)

print(confusion\_matrix(y\_test, predict\_nb)) print(classification\_report(y\_test, predict\_nb)) models.append(('naive\_bayes', NB)) detection\_accuracy.objects.create(names="Naive Bayes", ratio=naivebayes)

# SVM Model print("SVM")

from sklearn import svm lin\_clf = svm.LinearSVC() lin\_clf.fit(X\_train, y\_train)

predict\_svm = lin\_clf.predict(X\_test)

svm\_acc = accuracy\_score(y\_test, predict\_svm) \* 100 print(svm\_acc)

print("CLASSIFICATION REPORT")

print(classification\_report(y\_test, predict\_svm)) print("CONFUSION MATRIX")

print(confusion\_matrix(y\_test, predict\_svm)) models.append(('svm', lin\_clf)) detection\_accuracy.objects.create(names="SVM", ratio=svm\_acc)

print("Logistic Regression")

from sklearn.linear\_model import LogisticRegression

reg = LogisticRegression(random\_state=0, solver='lbfgs').fit(X\_train, y\_train) y\_pred = reg.predict(X\_test)

print("ACCURACY")

print(accuracy\_score(y\_test, y\_pred) \* 100) print("CLASSIFICATION REPORT")

print(classification\_report(y\_test, y\_pred)) print("CONFUSION MATRIX")

print(confusion\_matrix(y\_test, y\_pred)) models.append(('logistic', reg))

detection\_accuracy.objects.create(names="Logistic Regression", ratio=accuracy\_score(y\_test, y\_pred) \* 100)

print("Decision Tree Classifier") dtc = DecisionTreeClassifier() dtc.fit(X\_train, y\_train) dtcpredict = dtc.predict(X\_test) print("ACCURACY")

print(accuracy\_score(y\_test, dtcpredict) \* 100) print("CLASSIFICATION REPORT")

print(classification\_report(y\_test, dtcpredict)) print("CONFUSION MATRIX")

print(confusion\_matrix(y\_test, dtcpredict)) models.append(('DecisionTreeClassifier', dtc))

detection\_accuracy.objects.create(names="Decision Tree Classifier", ratio=accuracy\_score(y\_test, dtcpredict) \* 100)

print("KNeighborsClassifier")

from sklearn.neighbors import KNeighborsClassifier kn = KNeighborsClassifier()

kn.fit(X\_train, y\_train) knpredict = kn.predict(X\_test) print("ACCURACY")

print(accuracy\_score(y\_test, knpredict) \* 100) print("CLASSIFICATION REPORT")

print(classification\_report(y\_test, knpredict)) print("CONFUSION MATRIX")

print(confusion\_matrix(y\_test, knpredict)) models.append(('KNeighborsClassifier', kn))

detection\_accuracy.objects.create(names="KNeighborsClassifier", ratio=accuracy\_score(y\_test, knpredict) \* 100)

print("SGD Classifier")

from sklearn.linear\_model import SGDClassifier

sgd\_clf = SGDClassifier(loss='hinge', penalty='l2', random\_state=0) sgd\_clf.fit(X\_train, y\_train)

sgdpredict = sgd\_clf.predict(X\_test) print("ACCURACY")

print(accuracy\_score(y\_test, sgdpredict) \* 100) print("CLASSIFICATION REPORT")

print(classification\_report(y\_test, sgdpredict)) print("CONFUSION MATRIX")

print(confusion\_matrix(y\_test, sgdpredict)) models.append(('SGDClassifier', sgd\_clf))

detection\_accuracy.objects.create(names="SGD Classifier", ratio=accuracy\_score(y\_test, sgdpredict) \* 100)

print("Random Forest Classifier")

from sklearn.ensemble import RandomForestClassifier rf\_clf = RandomForestClassifier()

rf\_clf.fit(X\_train, y\_train) rfpredict = rf\_clf.predict(X\_test) print("ACCURACY")

print(accuracy\_score(y\_test, rfpredict) \* 100) print("CLASSIFICATION REPORT")

print(classification\_report(y\_test, rfpredict)) print("CONFUSION MATRIX")

print(confusion\_matrix(y\_test, rfpredict)) models.append(('RandomForestClassifier', rf\_clf)) detection\_accuracy.objects.create(names="Random Forest Classifier",

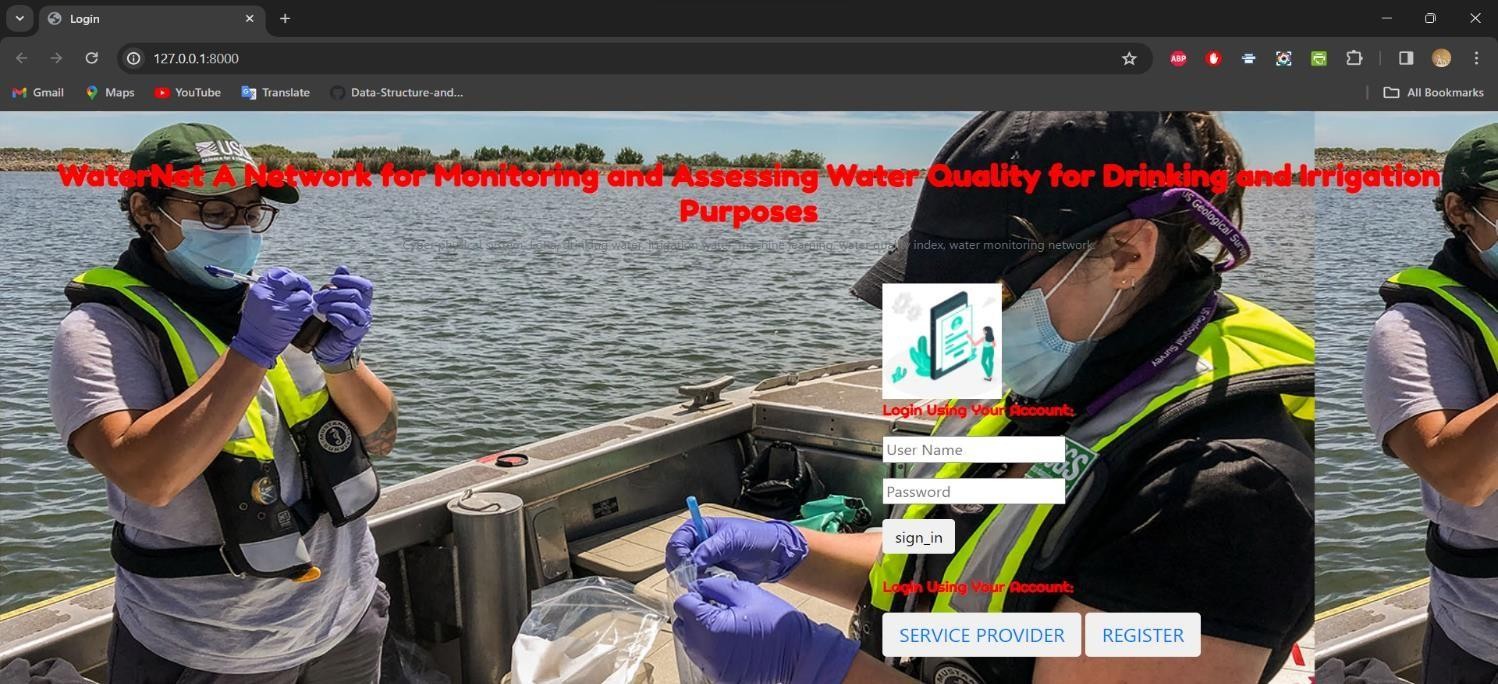
ratio=accuracy\_score(y\_test, rfpredict) \* 100)

obj = detection\_accuracy.objects.all()

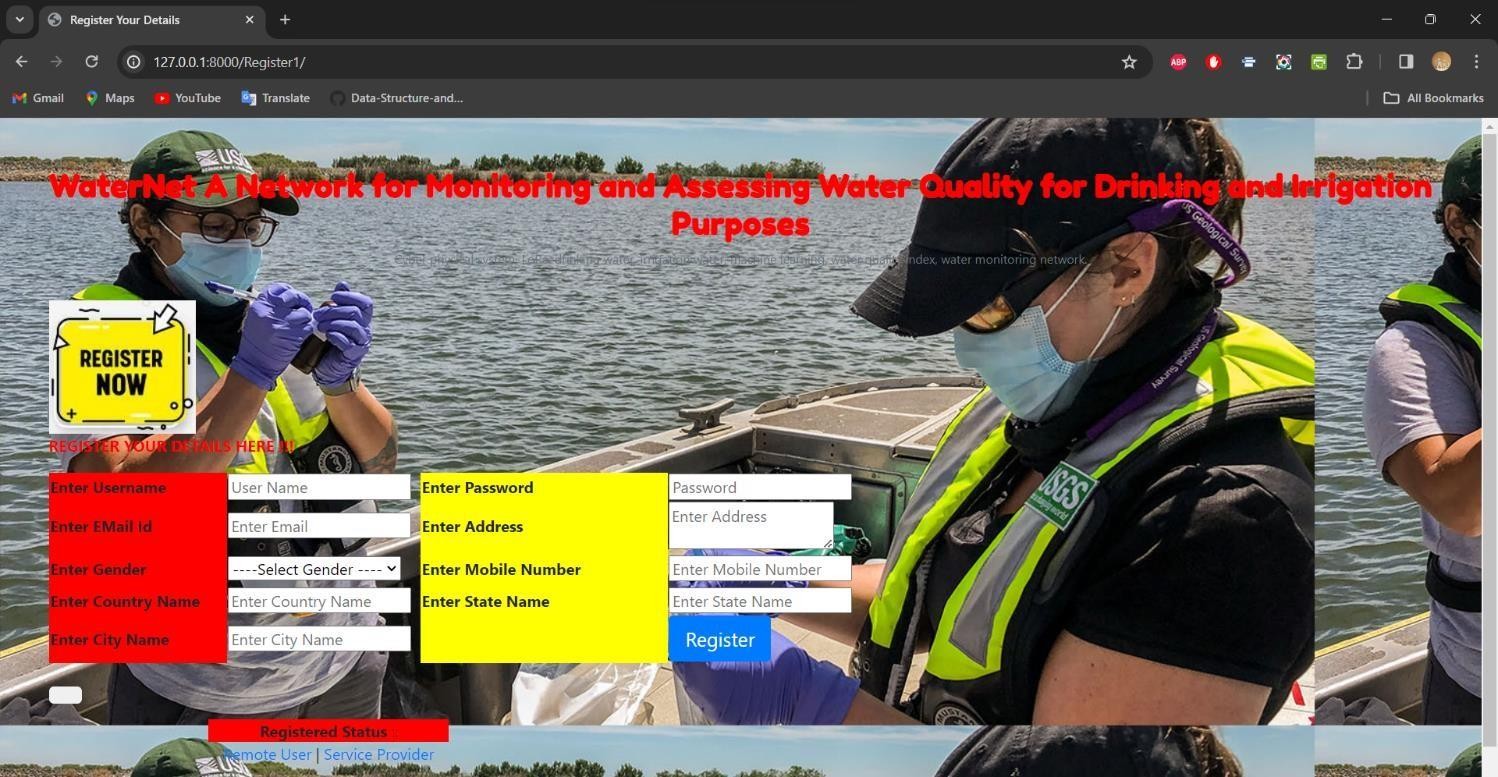
return render(request,'SProvider/Train\_Test\_DataSets.html', {'objs': obj})

* 1. **REMOTE USER**

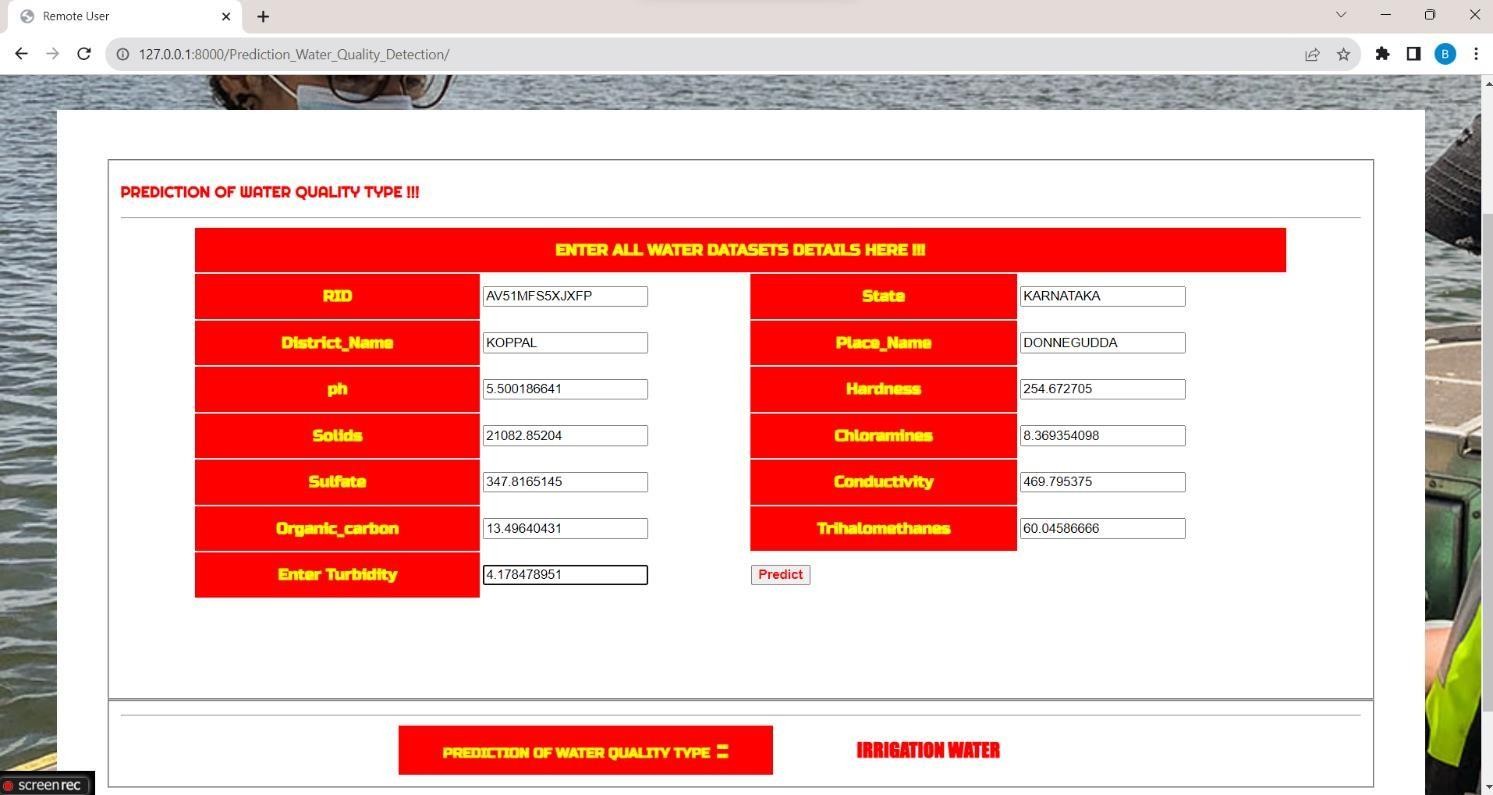
# 9 . SCREEN SHOTS



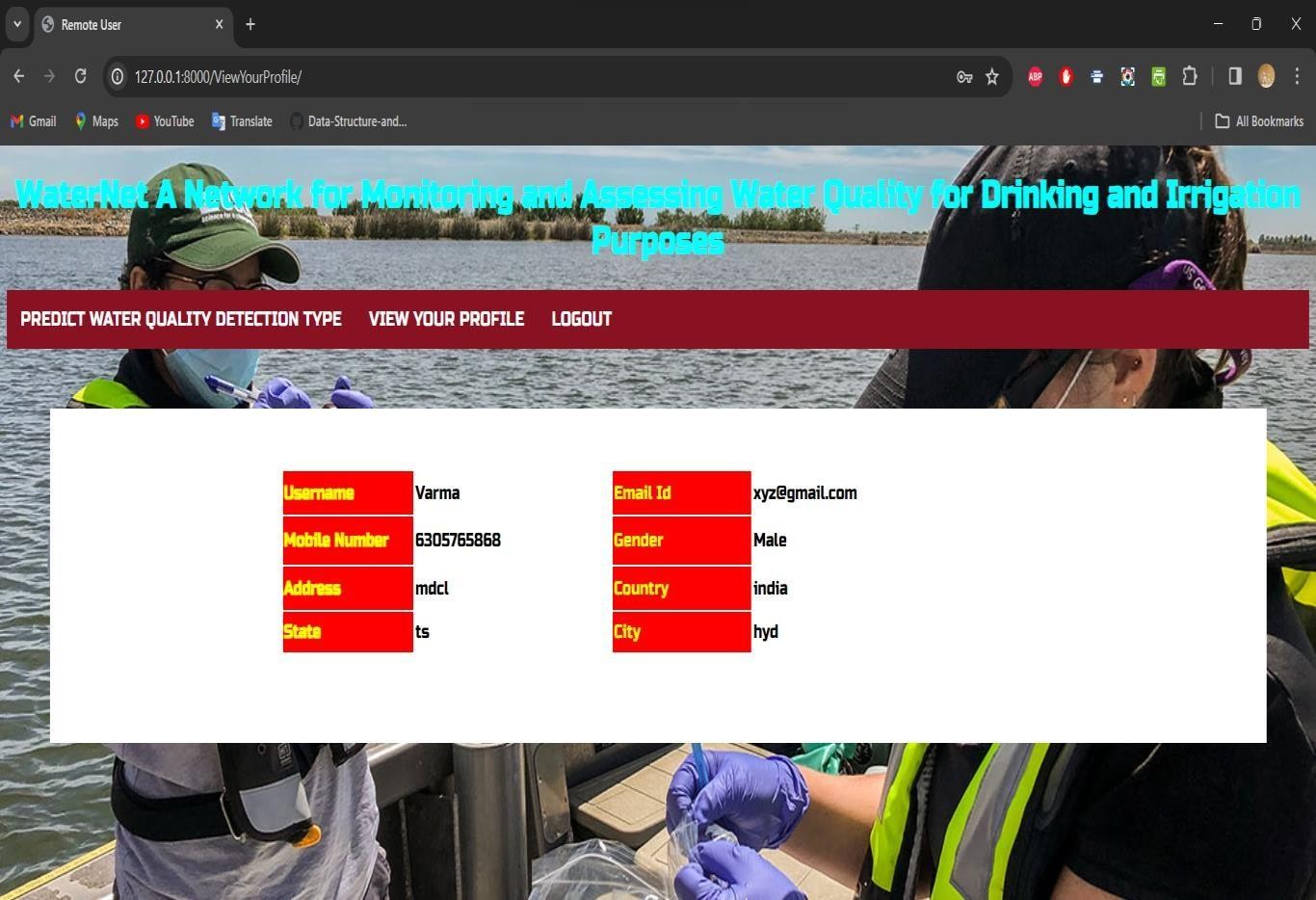
**Fig 9.1.1 Remote User Login Page**



**Fig 9.1.2 Remote User Registration Page**



**Fig 9.1.3 Remote User Water Quality Prediction Form**

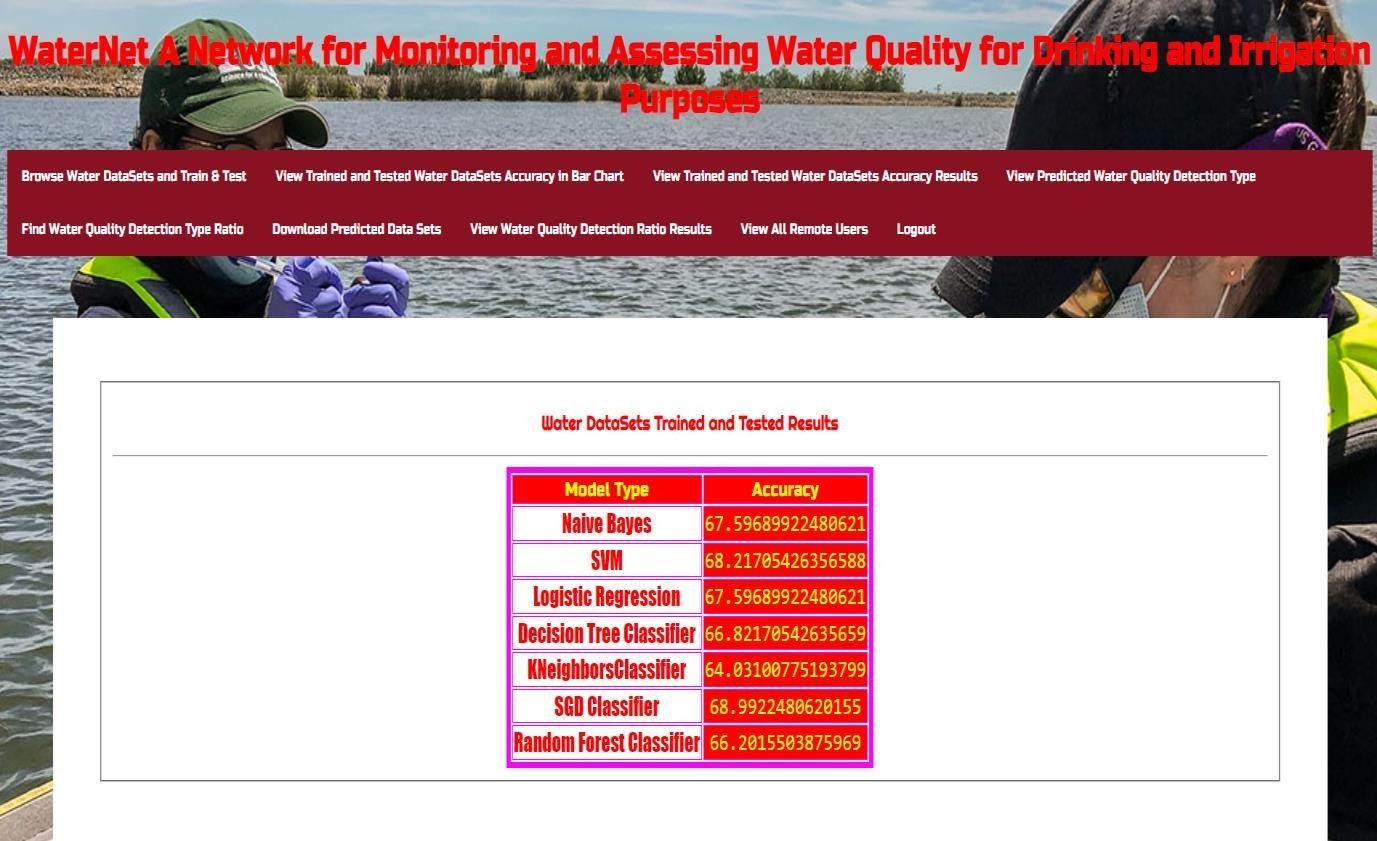


**Fig 9.1.4 Remote User Profile Page**

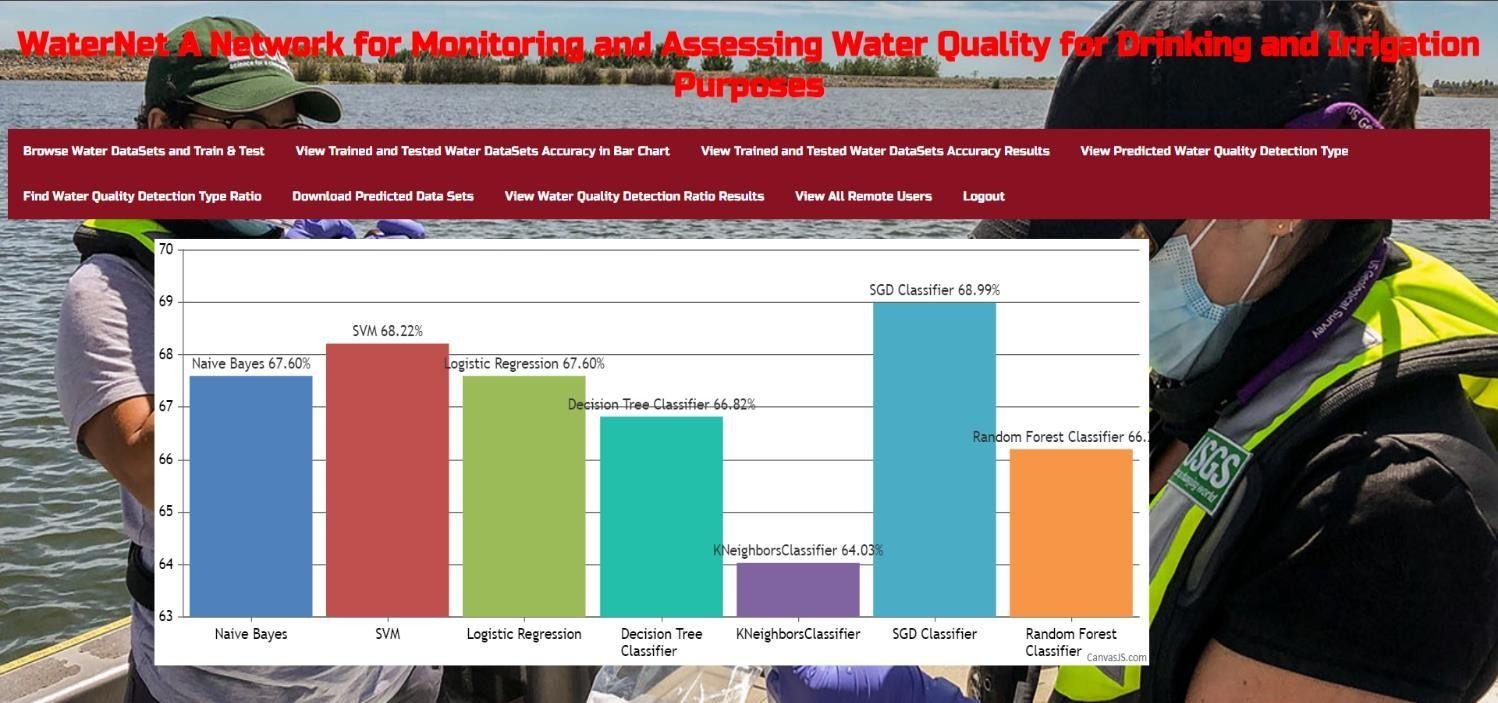
## SERVICE PROVIDER



**Fig 9.2.1 Service Provider Login Page**



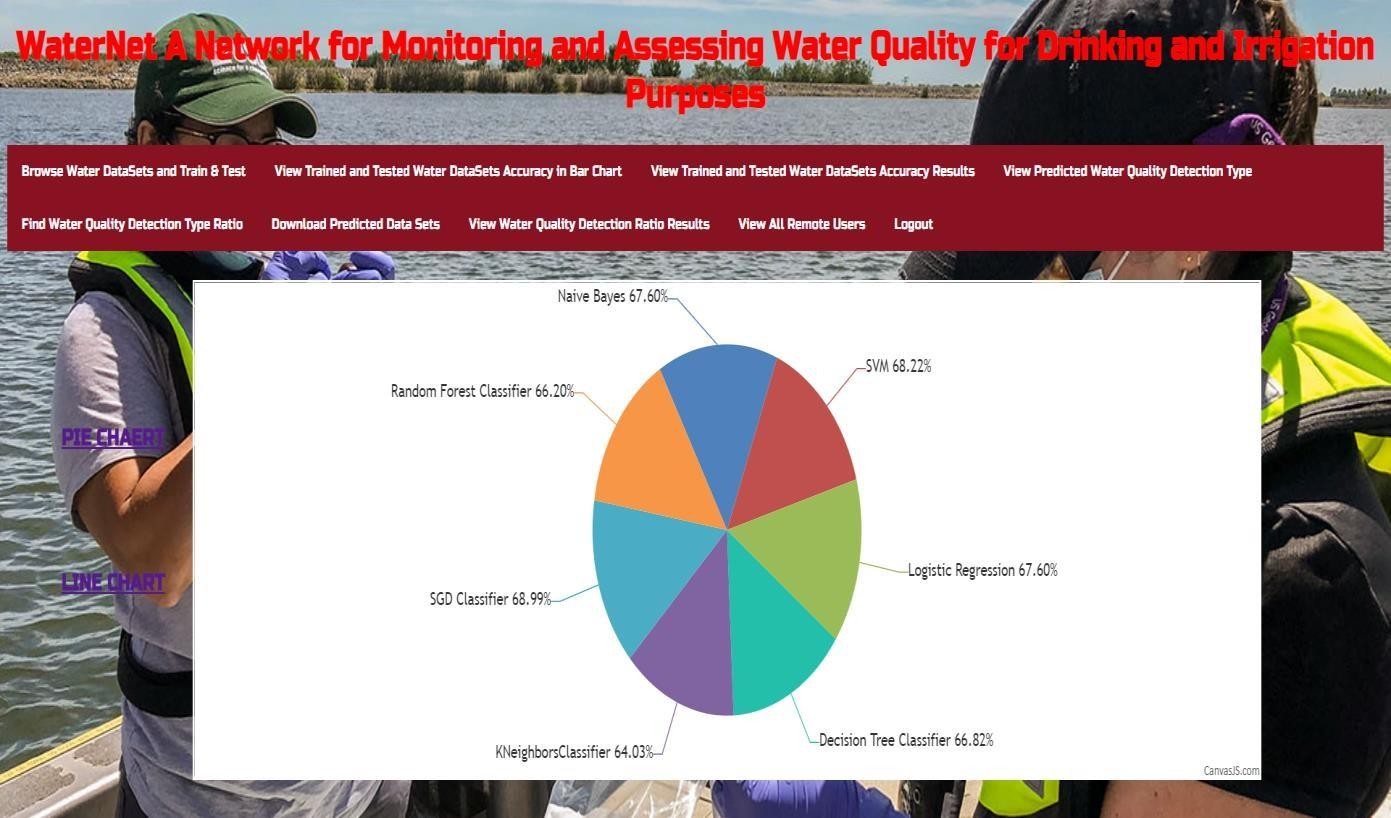
**Fig 9.2.2 Accuracy According to Algorithms**



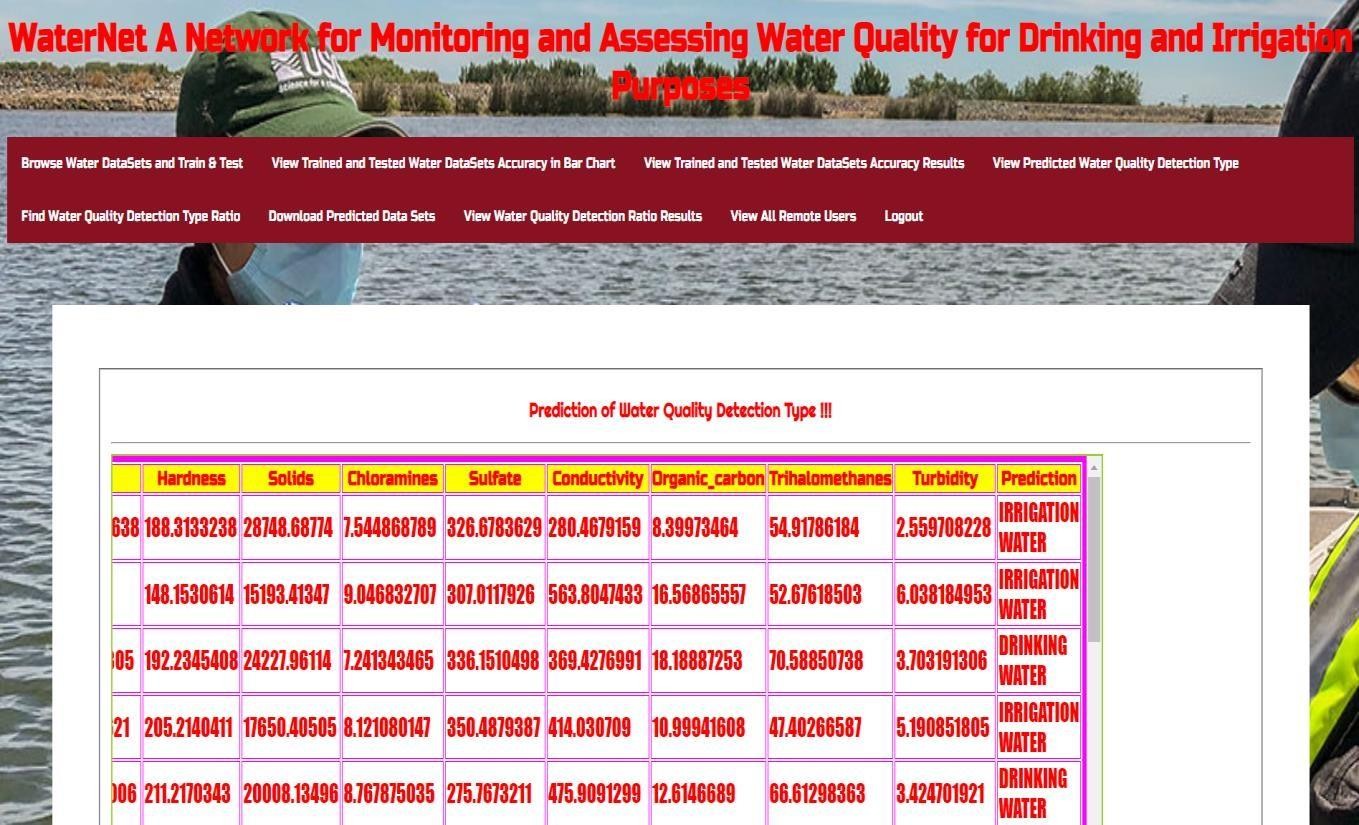
**Fig 9.2.3 Bar Chart for Accuracy Visualization**



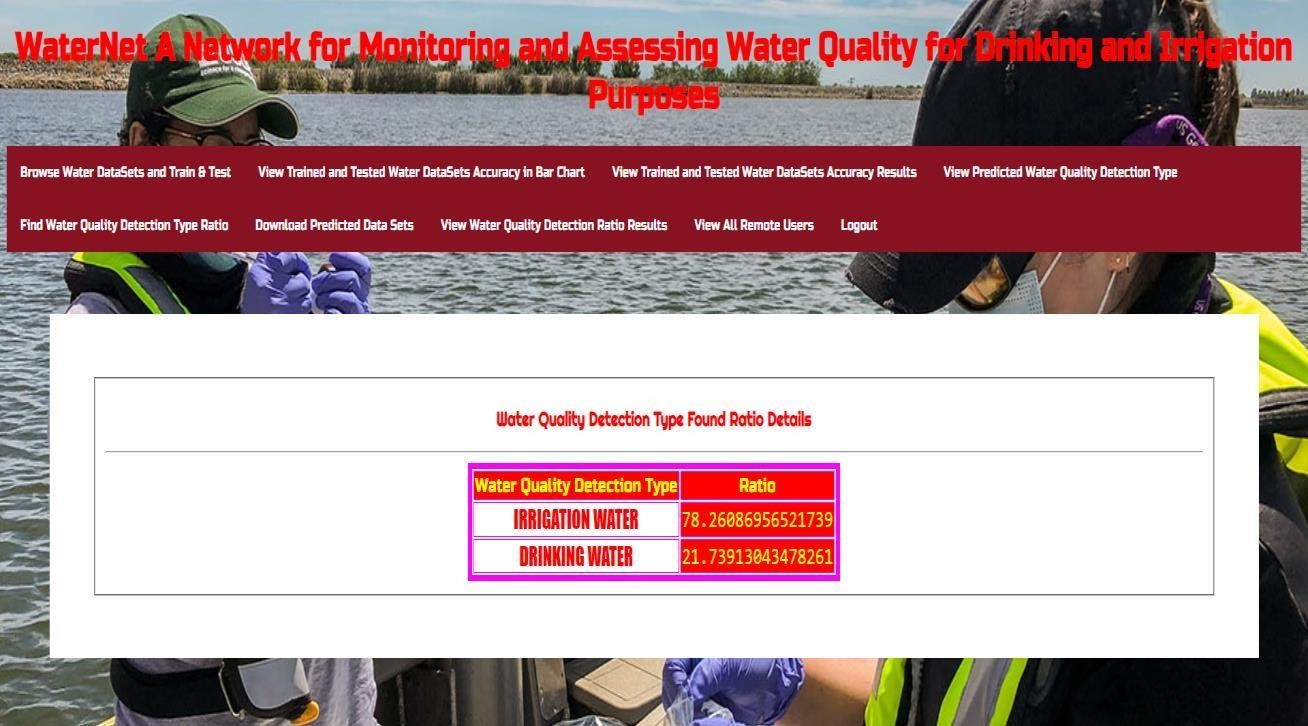
**Fig 9.2.4 Line Chart for Accuracy Visualization**



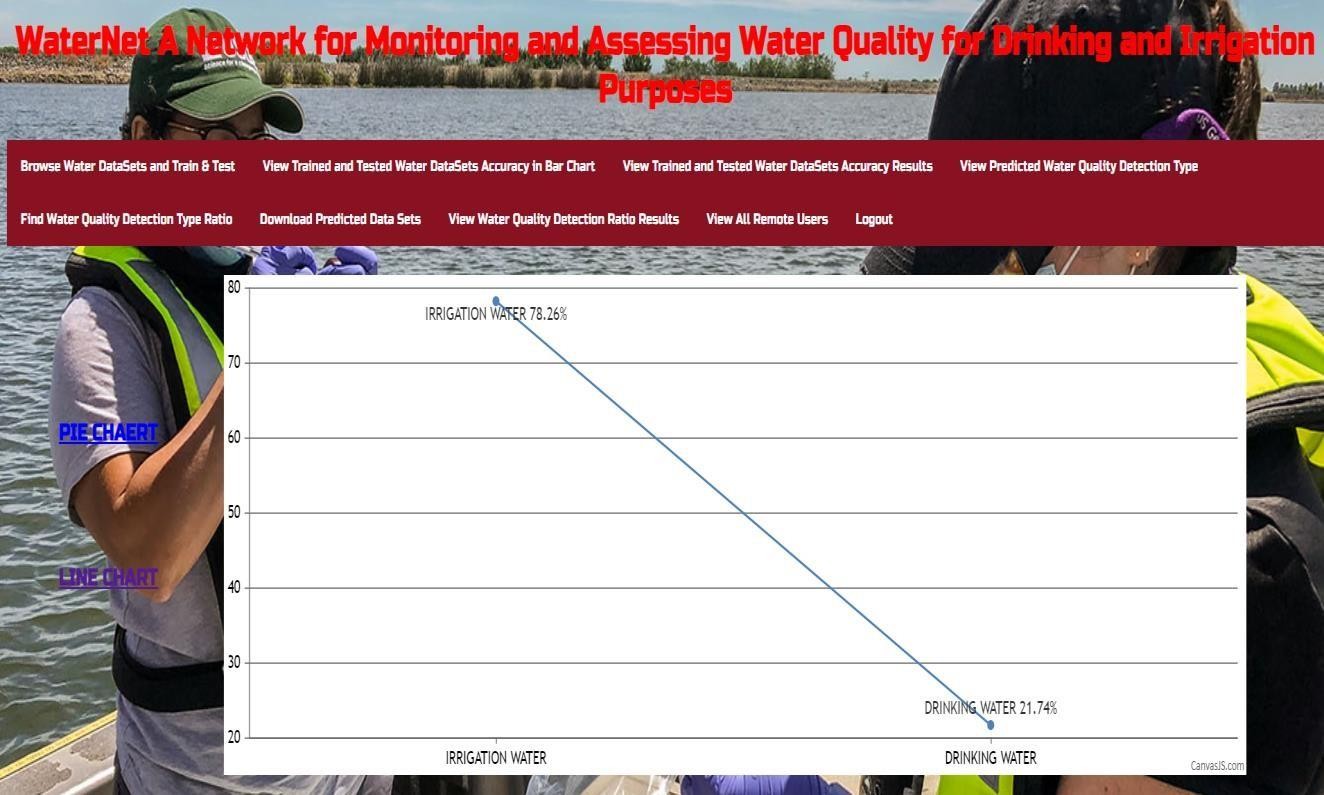
**Fig 9.2.5 Pie Chart for Accuracy Visualization**



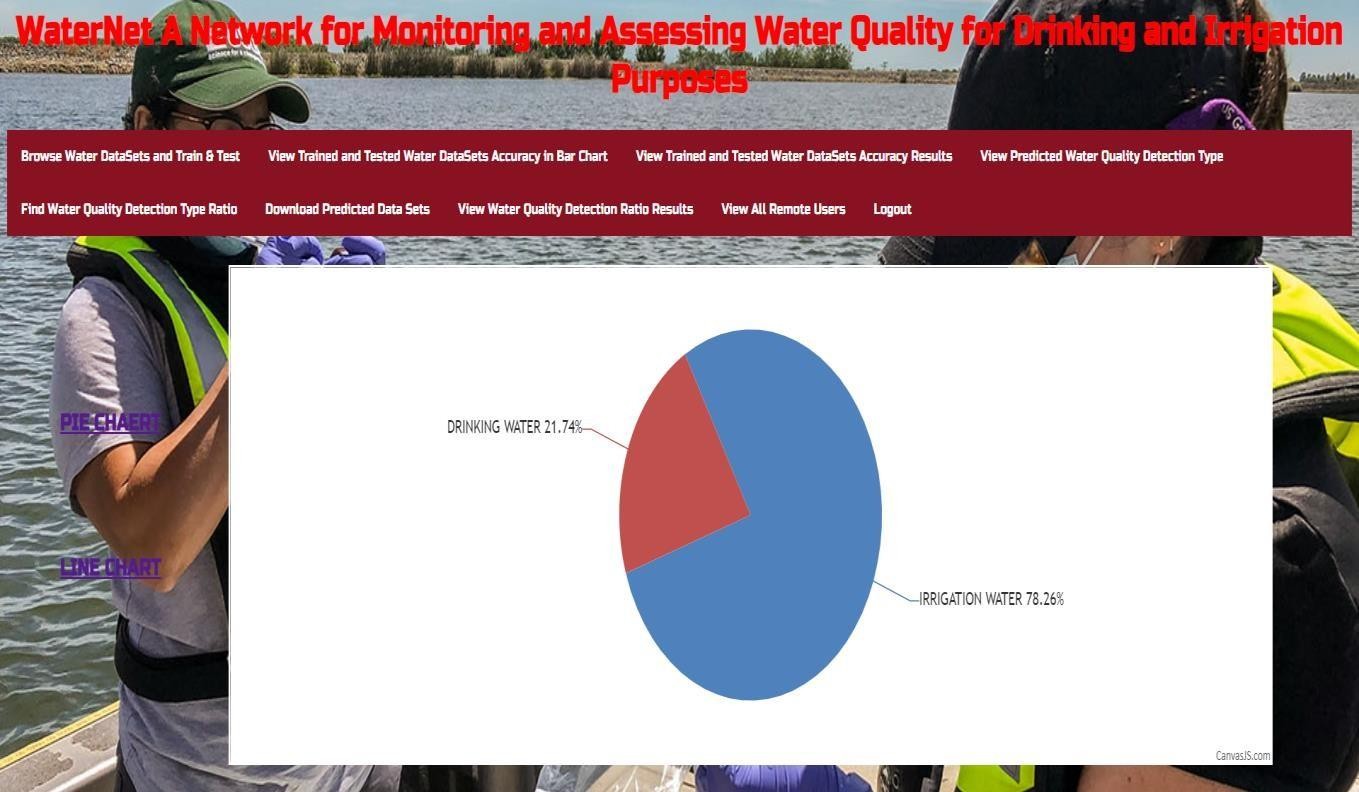
**Fig 9.2.6 View Predicted water quality detection type**



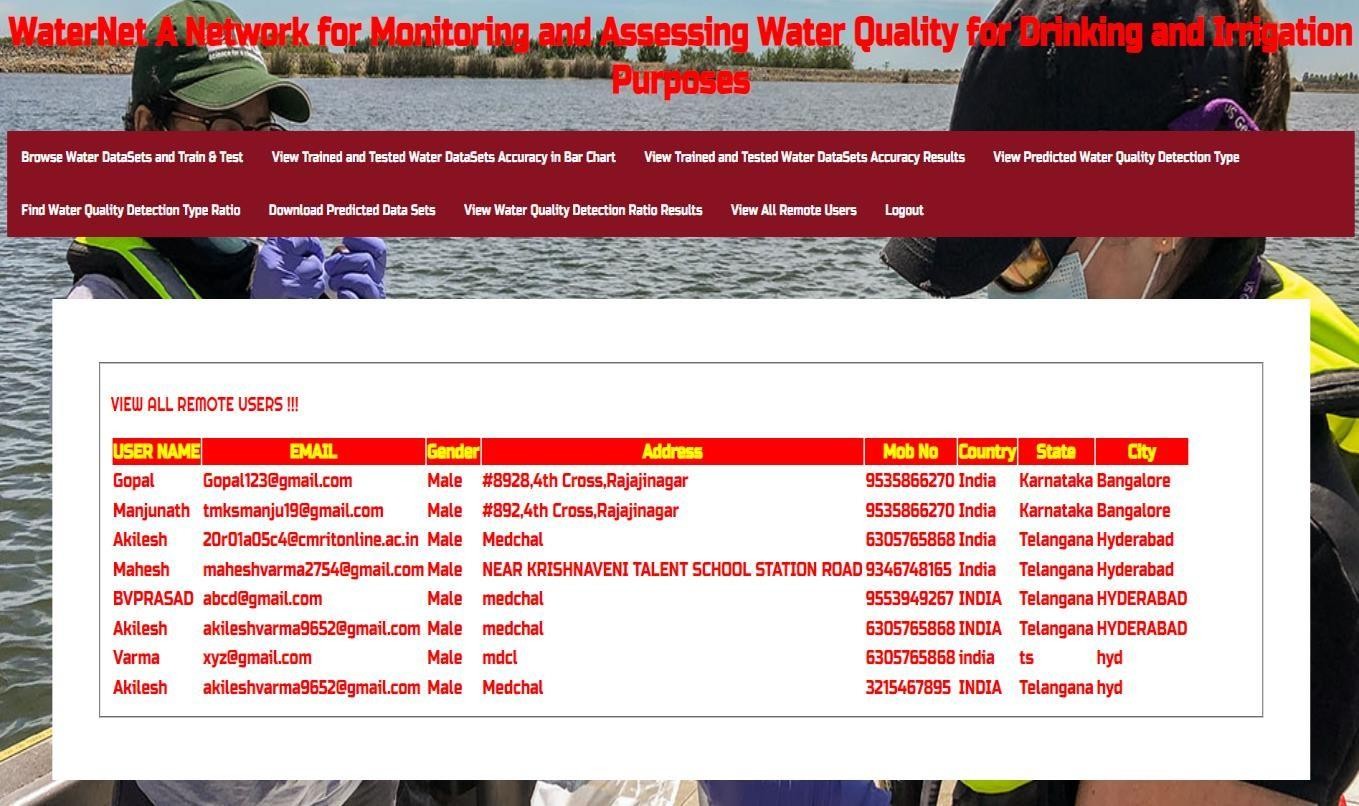
**Fig 9.2.7 Water Quality Detection Type Ratio Details**



**Fig 9.2.8 Water Quality Detection Ratio Results in Line Chart**



**Fig 9.2.9 Water Quality Detection Ratio Results in Pie Chart**



**Fig 9.2.10 List of Remote Users**

# 10.TESTING

## SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

## TYPES OF TESTS

### Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### Functional test

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:

Valid Input : identified classes of valid input must be accepted. Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised. Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

### System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

### White Box Testing

White Box Testing is a test in which the software tester has knowledge of the inner workings, structure, and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.





Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a test in which the software under test is treated, as a black box. you   responds to outputs without considering how the software works.

# 11.CONCLUSION

This work focused on two major concept, firstly, the proposal of a real-time water monitoring network for gathering data on water parameters from water bodies. Secondly, the application of machine learning (ML) models as means of assessing water quality. The developed water monitoring network is based on Lo Ra, a low power long range protocol for data transmission, and was developed using the City of Cape Town as case study. Results of the simulation done in Radio Mobile, revealed a partialmesh network topology as the most adequate network to cover the city. Data gathered from this monitoring network would ideally be aggregated on a Cloud server, where ML models can then be applied to assess the water's fitness of use for drinking or irrigation purposes. Due to the absence of relevant datasets, two suitable datasets were built in this work and used to training and testing three ML models considered, which are Random Forest (RF), Logistic Regression (LR) and Support VectorMachine (SVM). Results of the test showed that LR performed best for drinking water, as it gave the highest classification accuracy and lowest false positive and negative values, while SVM was better suited for irrigation water. Finally, a model for identifying the most influential water parameter(s) w.r.t classification accuracies of the ML models was then explored using recursive feature elimination (RFE). Obtained results showed that pH, and total hardness were the least influential parameters in drinking water, while SSP was the least for irrigation water.

Though the authors acknowledge the possible application of deep learning models, these were not used in this work. In future works, deep learning models such as the various variants of neural networks could be considered as expansion to this work. Furthermore, water quality indices were manually calculated and used to assess the ``fitness for use'' of water, future works could explore the application of unsupervised ML models as alternatives to manually calculated water quality indices. In the same vein, rather than using RFE, other approaches such as multi criteria decision making could also be considered to identify influential parameters. Finally, incorporating usage prediction models and microbial monitoring into the water network as well as tracking sources of water contaminates could also be avenues to further this work.

# 12.REFERENCES

B. X. Lee, F. Kjaerulf, S. Turner, L. Cohen, P. D. Donnelly, R. Muggah,

R. Davis, A. Realini, B. Kieselbach, L. S. MacGregor, I.Waller, R. Gordon,

M. Moloney-Kitts, G. Lee, and J. Gilligan, ``Transforming our world: Implementing the 2030 agenda through sustainable development goal indicators,'' *J. Public Health Policy*, vol. 37, no. S1, pp. 13\_31, Sep. 2016.

1. *Integrated Approaches for Sustainable Development Goals Planning: The Case of Goal 6 on Water and Sanitation*, U. ESCAP, Bangkok, Thailand,2017.
2. WHO. Water. *Protection of the Human Environment*. Accessed:

Jan. 24, 2022. [Online]. Available: [www.afro.who.int/health-topics/water](http://www.afro.who.int/health-topics/water)

1. L. Ho, A. Alonso, M. A. E. Forio, M. Vanclooster, and P. L. M. Goethals,

``Water research in support of the sustainable development goal 6: A case study in Belgium,'' *J. Cleaner Prod.*, vol. 277, Dec. 2020, Art. no. 124082.

1. *Global Nutrition Report 2016: From Promise to Impact: Ending Malnutrition by 2030*, International Food Policy Research Institute, Washington,

DC, USA, 2016, doi: 10.2499/9780896295841.

1. N. Akhtar, M. I. S. Ishak, M. I. Ahmad, K. Umar, M. S. Md Yusuff,

M. T. Anees, A. Qadir, and Y. K. A. Almanasir, ``Modi\_cation of the

water quality index (WQI) process for simple calculation using the multicriteria decision-making (MCDM) method: A review,'' *Water*, vol. 13,

no. 7, p. 905, Mar. 2021.

1. World Health Organization. (1993). *Guidelines for Drinking-Water Quality*. World Health Organization. Accessed: Jan. 12, 2022. [Online]. Available: <http://apps.who.int/iris/bitstream/handle/> 10665/44584/9789241548151-eng.pdf
2. *Standard Methods for the Examination of Water and Wastewater*, Federation WE, APH Association, American Public Health Association (APHA), Washington, DC, USA, 2005.
3. L. S. Clesceri, A. E. Greenberg, and A. D. Eaton, ``Standard methods for the examination of water and wastewater,'' Amer. Public Health Assoc. (APHA), Washington, DC, USA. Tech. Rep.21, 2005.
4. M. F. Howladar, M. A. Al Numanbakth, and M. O. Faruque, ``An application of water quality index (WQI) and multivariate statistics to evaluate the water quality around Maddhapara granite mining industrial area, Dinajpur, Bangladesh,'' *Environ. Syst. Res.*, vol. 6, no. 1, pp. 1\_8,

Jan. 2018[11] A. R. Finotti, R. Finkler, N. Susin, and V. E. Schneider, ``Use of water quality index as a tool for urban water resources management,'' *Int. J.*

*Sustain. Develop. Planning*, vol. 10, no. 6, pp. 781\_794, Dec. 2015.

1. A. R. Finotti, N. Susin, R. Finkler, M. D. Silva, and V. E. Schneider,

``Development of a monitoring network of water resources in urban areas as a support for municipal environmental management,'' *WIT Trans. Ecol. Environ.*, vol. 182, pp. 133\_143, May 2014.

1. M. Chilundo, P. Kelderman, and J. H. O'keeffe, ``Design of a water quality monitoring network for the limpopo river basin in Mozambique,'' *Phys.*

*Chem. Earth, A/B/C*, vol. 33, nos. 8\_13, pp. 655\_665, Jan. 2008.

1. M. Karamouz, M. Karimi, and R. Kerachian, ``Design of water quality monitoring network for river systems,'' in *Critical Transitions in Water*

*and Environmental Resources Management*. London, U.K.: IWA, 2004, pp. 1\_9.

1. J. Foschi, A. Turolla, and M. Antonelli, ``Soft sensor predictor of E. Coli concentration based on conventional monitoring parameters forwastewater disinfection control,'' *Water Res.*, vol. 191, Mar. 2021, Art. no. 116806.
2. Libelium.com. *IoT Solution for Water Management*. Accessed:

Jan. 27, 2022. [Online]. Available: https://[www.libelium.com/](http://www.libelium.com/) iot-solutions/smart-water/

1. K. Ma, A. Bagula, C. Nyirenda, and O. Ajayi, ``An IoT-based fog computing model,'' *Sensors*, vol. 19, no. 12, p. 2783, Jun. 2019.
2. I. Odun-Ayo, O. Ajayi, and A. Falade, ``Cloud computing and quality of service: Issues and developments,'' in *Proc. Int. Multi-Conf. Eng. Comput. Scientists (IMECS 2018)*, Hong kong, Mar. 2018, pp. 14\_16.
3. U. Raza, P. Kulkarni, and M. Sooriyabandara, ``Low power wide area networks: An overview,'' *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2,

pp. 855\_873, 2nd Quart., 2017.

1. F. M. Ortiz, T. T. de Almeida, A. E. Ferreira, and L. H. M. K. Costa,

``Experimental vs. simulation analysis of LoRa for vehicular communications,''

*Comput. Commun.*, vol. 160, pp. 299\_310, Jul. 2020.

1. H. A. Aden and K. R. Karlsson, ``Evaluating LoRa physical as a radio link technology for use in a remote-controlled electric switch system for a network bridge radio-node,'' M.S. thesis, Dept. Elect. Eng., School Elect. Eng. Comput. Sci., KTH Royal Inst. Technol., Stockholm, Sweden, 2018.
2. M. Zennaro, A. Bagula, D. Gascon, and A. B. Noveleta, ``Long distance wireless sensor networks: Simulation vs reality,'' in *Proc. 4th ACM Workshop Netw. Syst. Developing Regions (NSDR)*, 2010, pp. 1\_2.
3. M. G. Uddin, S. Nash, and A. I. Olbert, ``A review of water quality index models and their use for assessing surface water quality,'' *Ecolog. Indicators*, vol. 122, Mar. 2021, Art. no. 107218.
4. T. Banda and M. Kumarasamy, ``Development of a universal water quality index (UWQI) for South African river catchments,'' *Water*, vol. 12, no. 6,

p. 1534, May 2020.

1. P. Shrivastava and R. Kumar, ``Soil salinity: A serious environmental issue and plant growth promoting bacteria as one of the tools for its alleviation,'' *Saudi J. Biol. Sci.*, vol. 22, no. 2, pp. 123\_131, Mar. 2015.