**AI (ML DL) April15**

**Project**

**On**

**Urban Water Quality measurement**

By Team 55

1) Lavanya Jogi

2) Prathyusha Imandi

3) Maheshbabu Mamidisetti

4) Jagadesh kumar Mallavolu

**Table of Contents :**

1. INTRODUTION
   1. Overview
   2. Purpose
2. LITERATURE SURVEY
   1. Existing Problem
   2. Proposed Solution
3. THEORITICAL ANALYSIS
   1. Block Diagram
   2. Hardware/Software desinging
4. EXPERIMENTAL INVESTIGATIONS
5. FLOWCHART
6. RESULT
7. ADVANTAGES AND DISADVANTAGES
8. APPLICATIONS
9. CONCLUSION
10. FUTURE SCOPE
11. BIBILOGRAPHY
12. APPENDIX

12.1 Source Code

1. **Introduction** 
   1. **Overview:**

Prediction of water quality which can ensure the water supply and prevent water pollution is essential for a successful water transfer project. In recent years, with the development of artificial intelligence, the backpropagation (BP) neural network has been increasingly applied for the prediction and forecasting field. However, the BP neural network frame cannot satisfy the demand of higher accuracy. In this study, we extracted monitoring data from the water transfer channel of both the water resource and the intake area as training samples and selected some distinct indices as input factors to establish a BP neural network whose connection weight values between network layers and the threshold of each layer had already been optimized by an improved artificial bee colony (IABC) algorithm. Compared with the traditional BP and ABC-BP neural network model, it was shown that the IABC-BP neural network has a greater ability for forecasting and could achieve much better accuracy, nearly 25% more precise than the BP neural network. The new model is particularly practical for the water quality prediction of a water diversion project and could be readily applied in this field.

* 1. **Purpose:**

Nowadays, machine learning algorithms have proven themselves as a universal tool for different types of tasks, giving advanced possibilities for dealing with analysed data, including such types of tasks as data imputation, unsupervised clusterization, classification and regression. They are commonly used in many research areas; however, they are yet less common among environmental engineering workers, though such tools may provide an extremely efficient alternative to the traditional analytical approaches. (Wilcox, Woon and Aung 2013) The purpose of the research behind this thesis was in presenting of examples of how such advanced tools may be used on a particular data set meant for increasing water quality in european region. In the following chapters one will go through the presentation of the machine learning, it’s origins and possibilities in general, explanation of the data and models used during the research, results of the application of algorithms, discussion (covering obstacles one can face while working with this kind of models) and conclusion, which will cover the presented material, give advices for engineers and scientists who would like to use this models for their environmental tasks and finally and give some words about the possbile future of the development of these tools in environmental field.

1. **Literature Survey**

**2.1 Existing Problem**

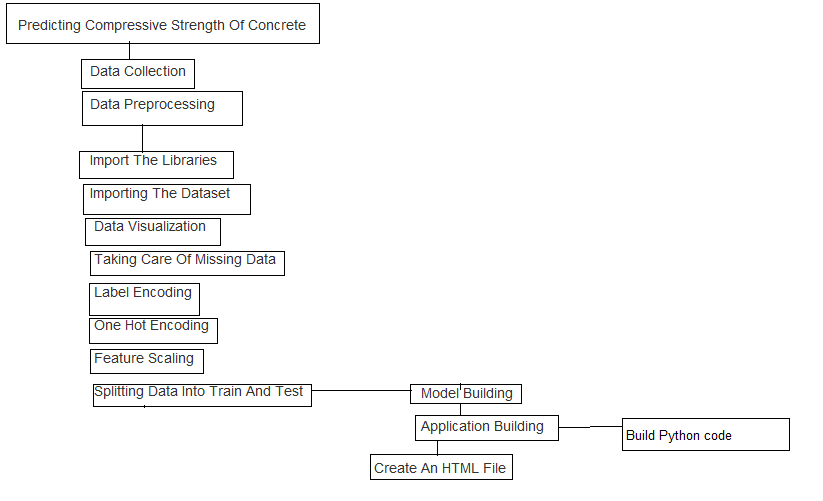
With the rapid development of economy and accelerated urbanization, water pollution has become more and more serious. Urban water quality is of great importance to our daily lives. Prediction of urban water quality help control water pollution and protect human health. To overcome this kind of problem statement, we developed a Machine learning model to predict the water quality.

**2.2 Proposed Solution**

Therefore, understanding the problems and trends of water pollution is of great significance for the prevention and control of water pollution. We have proposed a system that uses Machine learning algorithms to predict the water quality in Urban & to forecast the predictions.

1. **Theoretical Analysis**

**3.1 Block Diagram**

****

**3.2 Hardware / Software designing**

Python,Python Web Frame Works,Python For Data Analysis,Python For Data Visualization,Exploratory Data Analysis,Data Preprocessing Techniques,Machine Learning,Regression Algorithms,Classification Algorithms

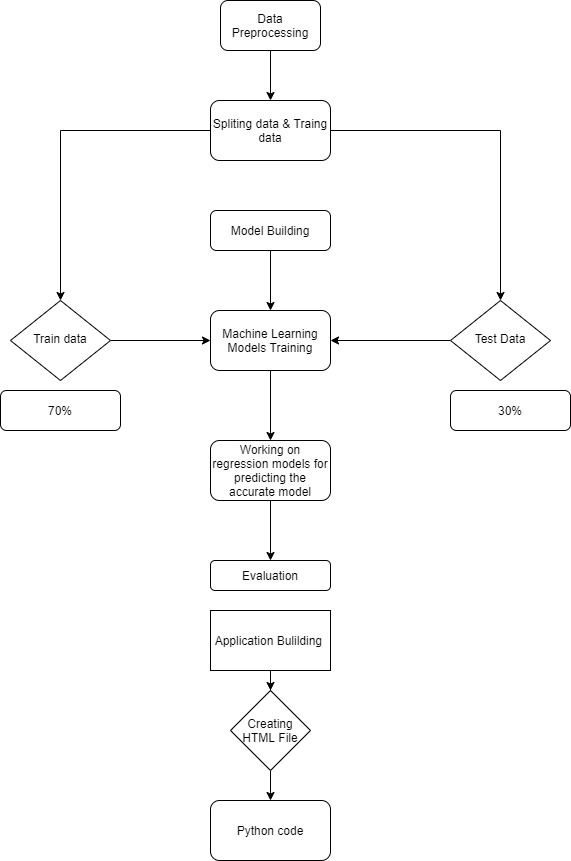
1. **Experimental Investigation**

For predicting the water Quality, we considering seven parameters namely DO, pH, Conductivity, BOD, Nitrates, Total Coliform and year. There is no particular ranges for these parameters. By taking the values of parameters and divide them into percentage and calculate wqi(mg/L)

wqi = wph + wdo + wbdo + wco + wna + wtc.

We get this wqi as predicted values.

1. **Flowchart**



1. **Result**

We have analysed the Quality of Water and used Machine Learning to Predict the quality of Urban Water. We have used Linear Regression and its variations, Lasso, Ridge and Random Forests to make predictions and compared their performance. Random Forest Regressor has highest accuracy and is a good choice for this problem. Random Forest Regressor trains randomly initialized trees with random subsets of data sampled from the training data, this will make our model more robust

1. **Advantages and Disadvantages**

**Advantages:**

Using Machine learning to predict the quality of water will be time and more accuracy in predicting the approximately close value can be done easily. Its more trust worthy and cost effective. It also reduces the man power for doing the experiments to find the quality of water in different unknown situations.

**Disadvantages :**

There is a 3 % chances that the outcome will not predict the approximate value in that situation it can be troublesome.

1. **Applications:**

* Can predict the Quality of Urban Water using the inputs provided.
* Implementable on the website

1. **Conclusion**

* The pH and ORP were identified as major water quality response parameters for the four inorganic pollutants studied. The responses in pure water appeared to be more sensitive, in terms of intensity, than those in tap water. In the pure water, a significant response was observed at the first level (the reference concentration allowed for each pollutant), whereas the response appeared at the second stage (10 times the threshold) in the case of tap water. Moreover, the pure water exhibited a higher strength at all stages.
* The results of the batch and simulated pipeline experiments for tap water showed almost identical tendencies, except for the second level of manganese injection (concentration 0.5 mg/L). However, the reasons for this difference remain unclear.

1. **Future Scope**

This model can predict the outcome with many different inputs within seconds. The model will save a lot of time While predict the quality of water. It costs nothing.

1. **Bibliography**

Batista, G. E. A. P. A., and M. C. Monard. “A Study of K-Nearest Neighbour as an Imputation Method .” Soft Computing Systems: Design, Management and Applications , 2002: pp. 251–260 . Behar, Sharon. Testing the Waters: Chemical and Physical Vital Signs of a River. River Watch Network, 1996. Biau, Gerard. “Analysis of a Random Forests Model.” The Journal of Machine Learning Research 13, no. 1 (2012): 1063-1095. Blackwell, Matthew, James Honaker, and Gary King. “A Unified Approach to Measurement Error and Missing Data: Overview and Applications.” Sociological Methods & Research,

2015: 1-39.

**Data repositories**

Kaggle.com

**Algorithms**

Thesmartbridgeteachable.com

1. **Appendix**

Source Code :

#IMPORTING LIBRARIES

import pandas as pd

import numpy as np

#IMPORTING THE DATA SET

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

data.dtypes

data['Temp']=pd.to\_numeric(data['Temp'],errors='coerce')

data['D.O. (mg/l)']=pd.to\_numeric(data['D.O. (mg/l)'],errors='coerce')

data['PH']=pd.to\_numeric(data['PH'],errors='coerce')

data['B.O.D. (mg/l)']=pd.to\_numeric(data['B.O.D. (mg/l)'],errors='coerce')

data['CONDUCTIVITY (µmhos/cm)']=pd.to\_numeric(data['CONDUCTIVITY (µmhos/cm)'],errors='coerce')

data['NITRATENAN N+ NITRITENANN (mg/l)']=pd.to\_numeric(data['NITRATENAN N+ NITRITENANN (mg/l)'],errors='coerce')

data['TOTAL COLIFORM (MPN/100ml)Mean']=pd.to\_numeric(data['TOTAL COLIFORM (MPN/100ml)Mean'],errors='coerce')

data.dtypes

start=2

end=1779

station=data.iloc [start:end ,0]

location=data.iloc [start:end ,1]

state=data.iloc [start:end ,2]

do= data.iloc [start:end ,4].astype(np.float64)

value=0

ph = data.iloc[ start:end,5]

co = data.iloc [start:end ,6].astype(np.float64)

year=data.iloc[start:end,11]

tc=data.iloc [2:end ,10].astype(np.float64)

bod = data.iloc [start:end ,7].astype(np.float64)

na= data.iloc [start:end ,8].astype(np.float64)

na.dtype

data.head()

data=pd.concat([station,location,state,do,ph,co,bod,na,tc,year],axis=1)

data. columns = ['station','location','state','do','ph','co','bod','na','tc','year']

data.head()

#Calculation of ph

data['npH']=data.ph.apply(lambda x: (100 if (8.5>=x>=7)

else(80 if (8.6>=x>=8.5) or (6.9>=x>=6.8)

else(60 if (8.8>=x>=8.6) or (6.8>=x>=6.7)

else(40 if (9>=x>=8.8) or (6.7>=x>=6.5)

else 0)))))

#Calculation of do

data['ndo']=data.do.apply(lambda x:(100 if (x>=6)

else(80 if (6>=x>=5.1)

else(60 if (5>=x>=4.1)

else(40 if (4>=x>=3)

else 0)))))

#Calculation of tc

data['ntc']=data.tc.apply(lambda x:(100 if (5>=x>=0)

else(80 if (50>=x>=5)

else(60 if (500>=x>=50)

else(40 if (10000>=x>=500)

else 0)))))

#Calculation of bdo

data['nbdo']=data.bod.apply(lambda x:(100 if (3>=x>=0)

else(80 if (6>=x>=3)

else(60 if (80>=x>=6)

else(40 if (125>=x>=80)

else 0)))))

#Calculation of Electrical Conductivity

data['nco']=data.co.apply(lambda x:(100 if (75>=x>=0)

else(80 if (150>=x>=75)

else(60 if (225>=x>=150)

else(40 if (300>=x>=225)

else 0)))))

#Calculation of nitrate

data['nna']=data.na.apply(lambda x:(100 if (20>=x>=0)

else(80 if (50>=x>=20)

else(60 if (100>=x>=50)

else(40 if (200>=x>=100)

else 0)))))

data.head()

data.dtypes

data['wph']=data.npH \* 0.165

data['wdo']=data.ndo \* 0.281

data['wbdo']=data.nbdo \* 0.234

data['wco']=data.nco\* 0.009

data['wna']=data.nna \* 0.028

data['wtc']=data.ntc \* 0.281

data['wqi']=data.wph+data.wdo+data.wbdo+data.wco+data.wna+data.wtc

data.head()

ag = data.groupby('year')['wqi'].mean()

ag

data = ag.reset\_index(level=0,inplace=False)

data

x = data.iloc[:,0:1].values

y = data.iloc[:,1:2].values

x

y

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

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size = 0.2,random\_state = 0)

from sklearn.linear\_model import LinearRegression

reg = LinearRegression()

reg.fit(x\_train,y\_train)

y\_pred = reg.predict(x\_test)

y\_pred

y\_test

import pickle

pickle.dump(reg,open('water\_quality.pkl','wb'))