IoT associated with Water Quality Estimation

Water Quality from freshwater sources is of considerable importance for the reason that these water resources are generally used for multiple purposes such as drinking, residential, industrial, agricultural or power generation purposes. Any Natural calamity or Disaster effects these to a very large extent.

Water Quality is a result of several inter-related parameters with a local temporal variation which is influenced by water flow rate during the year. Water Quality Index(WQI) is a 100-point scale that complete.

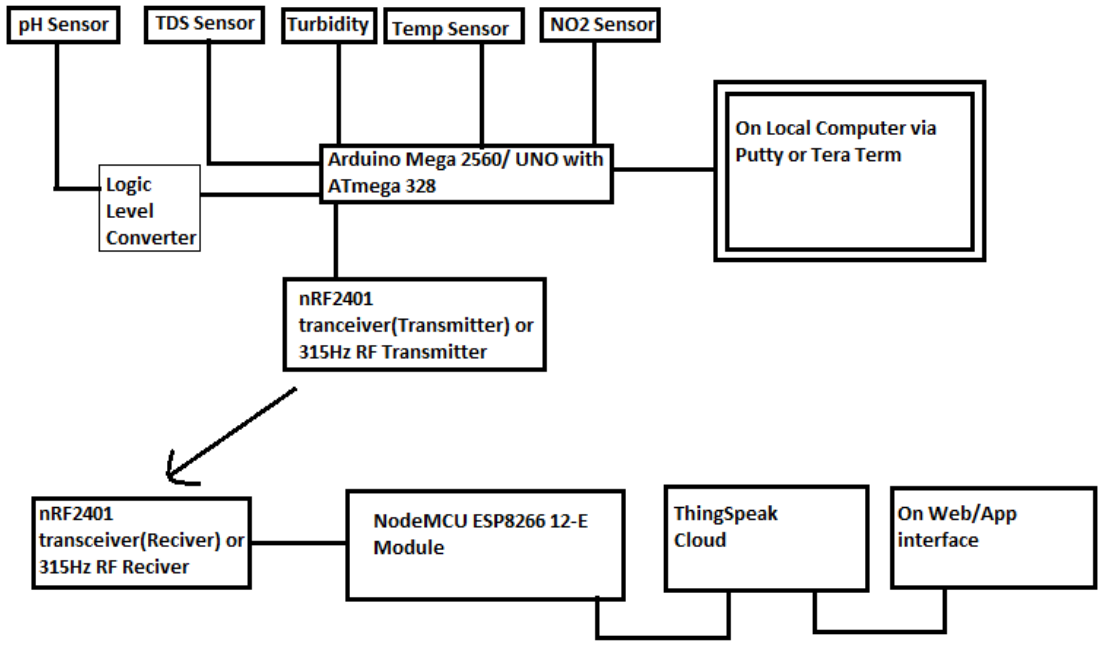
The formula for calculating Water Quality Index is

WQI = Summation of (Weightage(as an individual) * Water Quality Index)

The Weightage in the calculation of the Water Quality Index for some of the measurable factors associated with water quality has been mentioned below:

- pH -> 0.11
- Temperature change -> 0.1
- Nitrates -> 0.1
- Turbidity -> 0.08
- Total Dissolved Solid -> 0.07

CIRCUIT DIAGRAM:



Functioning of Hardware:

Once we set up the hardware as per the circuit diagram, as shown above, the Sensors would start responding to the commands as mentioned in the code. The pH, turbidity, and TDS(Total Dissolved Solids) sensor would send analog data which needs to be converted to there respective voltage equivalent and then to there respective value in units. After all this computation a copy of data would be stored on the local server using Putty or Teraterm for the dataset for the Regression-based Machine learning Algorithm in .csv format.

This data would then be transmitted to NodeMCU ESP8266 12-E Module from the Arduino Mega2560/UNO as a 1-D Array via a pair of NRF2401 Transceiver or RF Transmitter and receiver set. Once received this data would then be sent over to ThingSpeak from where Web/App Developers can access this information using API.

Software Abstract

Technologies Used: Django, python,postgred-SQL, ThingSpeak/ Firebase API(as per convenience),Google Maps API, web sockets, Chart.js ,Django-rest-framework, javascript, ajax, Django-background-tasks, HTML, CSS, etc

Hosting: Preferably Heroku or Python Anywhere or local-host

SUMMARY:

- We believe that we would be able to obtain precise readings through our sensors. However, data obtained without analysis is of no use.
- To properly analyze the data, we propose to create a web application to serve the purpose. The web application will be a management system which will be able to keep track of data sent from various location. We wish to classify the water quality at that point as good, moderate and unhealthy.
- IOT will send the fields PH, nitrite concentration, turbidity through thingspeak. We will collect data continuously through endpoints provided by thingspeak. Salinity will be calculated through the formula given below:
  pH = (2559.7 + 4.5S)/T – 0.5523- 0.01391 S where S is salanity
- The aggregate water quality will be calculated through the formula:

  WQI = Summation of (Weightage(as an individual) * Water Quality Index)

- Through aggregate water quality index, we'll classify the water as good, moderate and unhealthy.
- On a map, we'll color code location with green, yellow and red as per water quality.
- Through the machine learning algorithm, we'll predict the quality of the water quality index at a particular point of time in the future. The method is summarized below. There will be two fields that will be used for prediction: time and water quality index.
- Chart.js will be used to provide a graphical description

How will real-time data be obtained and parallel-processing of data be done?

- Django background task is a python library which is a custom implementation of cron jobs and allows asynchronous data processing. It allows various functions for processing to be executed asynchronously. As soon as we trigger the data collection process for each location through our web app, a function starts executing infinitely until stopped. Through this function, data will be collected through thinspeak API
- As soon as data will be collected for each location, sockets will be triggered by the backend. Data will be updated in front end without the need to refresh the page.

How will predictions be done?

- As soon as the prediction method is called for a particular location is triggered, they'll be processed through ML. A graphical time vs water quality index representation will be shown.

Why we have proposed a web-application instead of a mobile app?

- We believe that a lot of data is involved in the web page. The user needs to monitor data from a different location and a desktop application is always preferable in such a scenario for better user experience. However, we'll create APIs for web applications and

try to create a basic android application as per time constraints. Preferably flutter or MIT App Inventor will be used for that purpose.

Machine Learning Abstract(prediction part)

Prediction is based purely on regression,to be precise univariate linear regression.We are predicting water quality index wrt time.

Our algorithm goes as follows :-

1. Training example

   They are of form (t^i,qlt^i).

   t=time ; qlt=water quality index at time t

2. Prediction

   Func used:def predict(t,a,b):

   #Usage of cost function

3.Cost derivative

Errors are data-dependent

Func used:def cost_derivative(t,qlt):

#Usage of cost derivative

4.Learning Rate

Coefficients are updated wrt errors calculated.

Learning Rate is ratio of current error to error that previous coefficients caused.

5.Max Iteration type

Model is trained for a specific number of iterations.Suppose you have m training examples. Then,no. of iterations would be equal to m.