E-TONGUE: A SMART TOOL TO PREDICT THE SAFE CONSUMPTION OF GROUND WATER

Final (Draft) Report

Gunarathna PMSSB IT16168800

Nibraz MM IT16165694

Thenuja S IT16170780

Alahakoon AMPB IT16175280

B.Sc. (Hons) Degree in Information Technology
Specialized in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

September 2020

E-TONGUE: A SMART TOOL TO PREDICT THE SAFE CONSUMPTION OF GROUND WATER

Gunarathna PMSSB IT16168800

Nibraz MM IT16165694

Thenuja S IT16170780

Alahakoon AMPB IT16175280

The dissertation was submitted in partial fulfillment of the requirements for the BSc Special Honors degree in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

September 2020

DECLARATION

We declare that this is our own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or institute higher learning and to the best of our knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. we retain the right to use this content in whole or part in future works (such as article or books).

Student Name	Student ID	Signature
Gunarathna PMSSB	IT16168800	
Alahakoon AMPB	IT16175280	
Thenuja S	IT16170780	
Nibraz MM	IT16165694	

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

…………………………………….. ……………………..

Signature of the Supervisor: Date:

ABSTRACT

Human body uses water in every cells, organs and tissues to help regulate body functions and temperature. It is vital that the water used by the body organs are of good quality. In Sri Lanka, reportedly 59.4% of population depends on water from natural sources which grabs the attention to make sure that these people are receiving and dealing with a safe water with a good quality for the usage. Although government is taking necessary action to provide a better quality of water, there has been always a need for a better educational session to educate people about the importance of maintaining water quality, importance of using a better-quality water and necessary precautions to be taken to avoid any health hazards. Taking this issue into consideration, it is mainly recognized that a smart solution must be implemented in order to solve the identification of water quality problem. E-Tongue: a smart device to predict safe consumption of ground water is an attempt to assist any kind of users to identify the water quality of a groundwater sample in real time by analyzing the water quality parameters to predict the Water Quality Index. This task is achieved by designing a hardware device that embeds a set of sensors to read the value of water quality parameters which will be then transferred to cloud environment for an easy access by the mathematical model to process and identify the WQI value. It will then predict the water parameter readings that could be changed in future along with any disease outbreak possibilities. All the outputs will be finally displayed through mobile application.

Keywords: *water quality, Water Quality Index (WQI), hardware device, mobile application.*

ACKNOWLEDGEMENT

First and foremost, we would like to express our sincere gratitude to our supervisor Ms. Chathurangika Kahandawaarachchi for the constant guidance and support which helped us at all times for the successful completion of us undergraduate research.Besides our supervisor, our sincere thanks also go to Ms. Narmadha Gamage, the co-supervisor of this research project for being willing to help whenever it was needed.

Getting through this dissertation required the nurturance of many individuals. Our appreciation goes out to Dr.Dharshana Kasthurirathna for willingly sharing his knowledge related to Machine learning concepts and providing me with necessary knowledge source throughout this journey. This research study being a mixture of technology and irrigation required the guidance and assistance of not only technology experts but also professionals from water resource authorities. The immense support extended by Mr. W.G.T. Indragith, the Manager of the groundwater-Studies, Seniro Hydrologist of National Water Supply and Drainage Board, Ratmalana helped me throughout the project to bridge the knowledge gap in those areas.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations	**Description**
ML	Machine Learning
AI	Artificial Intelligence
IoT	Internet of Things
WHO	World Health Organization
LSTM	Long-Short-Term-Memory
COD	Chemical Oxygen Demand
DO	Dissolved Oxygen
BOD	Biochemical Oxygen Demand
EC	Elective Conductivity
WQI	Water Quality Index
SDLC	Software Development Life Cycle

1. INTRODUCTION

1.1 Introduction

As a country which has a rich history of agriculture, irrigation and architecture, Sri Lanka has faced an abundant ups and downs in keeping up the legacy in constant. Although the country is filled with trees, water sources and landscapes, the lack of proper resources has been a burden which pushed the day to day life to a questionable place. As we humans possess a copious knowledge in the important of food, water and air for the survival, it is our prime duty to make sure that the essentials that are keeping us alive is obtained and consumed in a safe way which does not bring any harm to our own selves and to the others.

The conventional approach for observing the water quality is with the end goal that the water test is taken and sent to the lab to be tried physically by expository strategies. Even though by this strategy the substance, physical, and organic particles of the water can be dissected, it comes with disadvantages. Right off, it is tedious and works escalated. Besides, the expense of this strategy is high because of the activity cost, work cost, and gear cost, and it is hard to detect analytical choices in real-time [P2].

Through the establishment of national water supply and drainage board and water resource authorities, Sri Lankan government has taken strides in fulfilling the needs of the people to make sure the water is supplied to as much as the community as possible. By interacting with the locals regarding their needs and the sources of water they are based on, these centers have become one important source of information to the government as well as resources of innovation to be implemented locally. The National Water Supply and Drainage Board, Ratmalana is one such center which contributed immensely towards this research project.

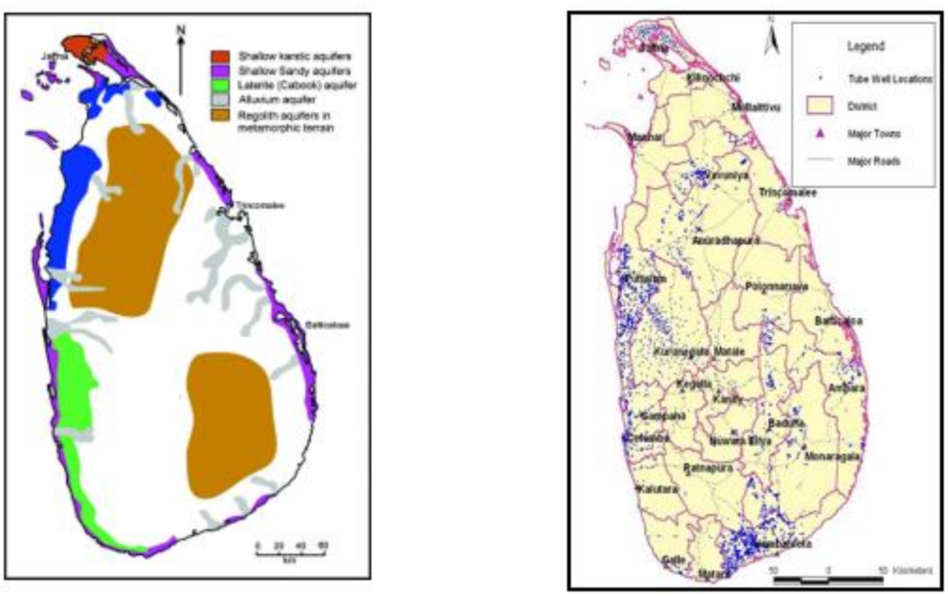


Figure 1: Types of aquifers and distribution of tube well sites in sri lanka

1.2 Background Literature

1.2.1 background about device

This is an enormous water amount and if 1% of water misfortune is spared it will be adequate to give water to 20,000 families [p1]. The related cost that the nation could be spared will be roughly Rs. 3,000 million every year. This sparing is adequate to build 1 km of interstates for every year. Along these lines, if more levels of water misfortunes can be wiped out the advantages are so high. Supply of consumable water is a costly issue and accordingly, the capital speculation required for the arrangement of consumable water surpasses Rs. 175,000 to 300,000 for each family. Consequently, the measure of cash required for water supply is enormous and this has gained the ground of providing water slower than usual. Simultaneously if the water created is utilized for expected purposes, without wastage and ill-advised utilization, the inclusion of the population can be expanded without going for additional speculations [p1].

Our proposed approach, a possess collected Arduino microcontroller is utilized as the central controller of the system. When the code is transferred to the microcontroller. Right now, sensors are utilized to gauge the fundamental water parameters. As it was considered from the past investigates, the most fundamental water parameters should have been checked by the normal clients are water pH level, water turbidity (darkness), water temperature, conductivity, and the calcium level. Along these lines, Sensors' circuits are associated with the microcontroller and the tests of the turbidity, pH, conductivity, and temperature sensors put inside the water. A waterproof temperature sensor is utilized to dodge any harm or electrical stun to the device and the end-user. All sensors read the water quality parameters and send the information to the microcontroller as electrical signs and it will be transmitted to the cloud server through the WI-FI module.

M.K.Khurana and his colleagues proposed a water quality observing device that can examine the nature of water and impart a caution signal to the authorities through Wi-Fi. If the water parameter is certainly not an ideal worth [7]. This platform gives an exact estimation of the water parameters since right now pH sensor is twofold aligned. Be that as it may, this is only capable of showing to the pH level of the water and no other water quality parameters.

N. Vijayakumar and R.Ramya concocted a thought for the constant water quality observing in the IoT (Internet of Things) condition. Their framework comprises a few sensors which can quantify some basic parameters of the water, for example, temperature, pH level, conductivity, turbidity, and the information can be seen on the web utilizing distributed computing. Center Micro controller implement here is raspberry Pi, its detriment is that it is run on LINUX utilizing the console. It requires the clients to include a command each time when they need to realize the sensors perusing [3].

A.S. Rao, built up a water quality observing framework utilizing Arduino Mega 2560 microcontroller and separate sensors to screen the temperature, conductivity, pH, broke up oxygen, light, and oxidation decrease capability of the water. Even though it comprises of complex wiring and requires a PC or an additional Beagle board XM ARM processor for correspondence interface and activity [4].

M Deqing, Z. Ying and C.Shangsong, in [8], utilized the Global System for Mobile Communications to detect the nature of water remotely. In their proposed framework, the basic water quality parameters, in particular, pH level, conductivity, dissolved oxygen, and turbidity are perused from the water through the individual sensors and it is then dissected by the controller and in the event that it is past the standard range, it is sent to relevant parties through an SMS, simultaneously. The information is also put away in a database and it is plotted to a graph for additional analysis. Be that as it may, this product is moderate for large water provider organizations or ventures since it comprises of costly parts.

Fu Qi-Feng et al's. framework contains an alert component that informs the production line representatives when the deliberate boundaries for example ammonia nitrogen and several other parameters are out of range. if the boundaries are not in the ostensible territory, at that point it very well may be sent to a treatment tank which it tends to be dealt with once more. Baihaqi Siregar et al. utilized a straighter forward methodology utilizing Waspmote as a micro controller as a remote association with the sensors. 3G component was utilized for sending information to a cloud which is then shown in diagrams and if any of the qualities are far-off, a SMS warning is transmitted to alarm the client. Here used methodology is major pertinent contrasted and the past undertaking [20].

1.2.2 An overview of Chronic Kidney Disease in Sri Lanka

Kidneys are one of the most vital organs that a human body have and it performs the function of purifying the blood. Toxins and liquid wastes that were found in blood is filtered and drained out from the body by the help of kidneys. There are spectrum of reasons and factors which causes kidney damages and as scary as it sounds, damages to kidneys will be a fatal condition where serious medical condition is required. Among the variety of diseases that are related to kidneys, Chronic Kidney Disease (CKD) is one common, yet more dangerous disease which has been ruining lives of many people around the world.

Conditions like Hypertension, Diabetes mellitus and various forms of glomerulonephritis are some common conditions which has a highest risk of causing CKD among people [6]. However, in 1990 a new CKD with no identified causes has been escalating in Sri Lanka especially in rural areas which was later on commonly called as *chronic kidney disease of unknown etiology* (CKDu). Studies found out that even though CKDu is caused by many distinguished factors, the places which has high presence of CKDu is mainly due to the consumption of contaminated water. Agricultural and metal wastes have been identified as the important cause for the spreading of CKDu in districts like Anuradhapura and Polonnaruwa since the people are mostly depending on ground water sources which has direct seepage from all these above-mentioned contaminations.

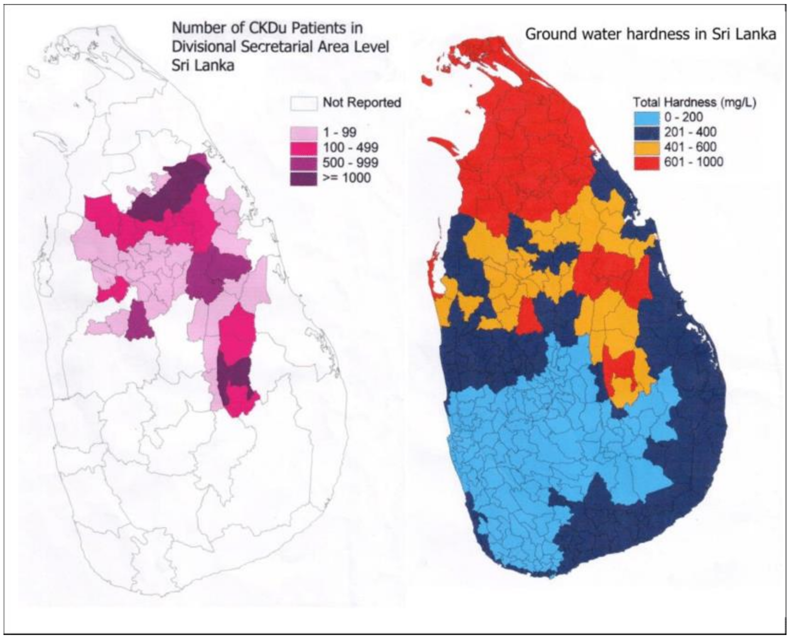


Figure 2: Geographical distribution of patients with CKDu and ground water hardness in Sri Lanka

Alongside the contaminated wastes from various kind of activities, the natural chemical compositions of the places too plays an important role in defining the chemical composition of the ground water [6]. Ions like fluoride, calcium and sodium that were naturally present in rocks could make the water a hard water which has a possibility in causing CKDu. Pesticides such as 2,4-D, 3,5,6-trichloropyridinol, p-nitrophenol, 1-naphthol, 2-naphthol, glyphosate, and AMPA have been detected in the urine of CKDu patients, however whether any of these agents appear to have a causative role in the etio-pathogenesis of CKDu has not been established [7]. A causal relationship of CKDu to acetylcholine esterase (AChE) inhibitor pesticides was explored in a study conducted in affected areas when the disease entity was newly recognized [8].

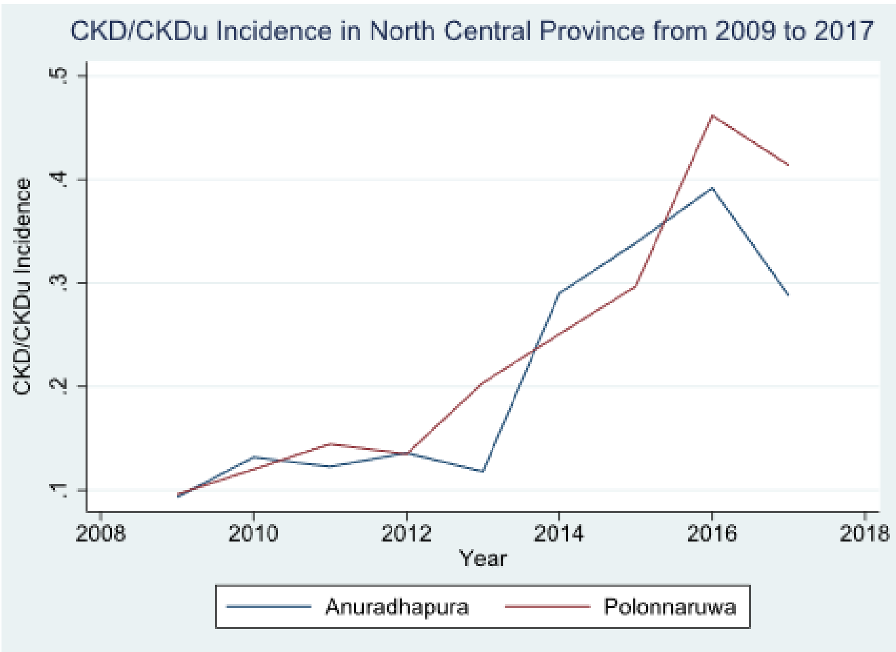


Figure 3: Incidence of CKD/CKDu patients in Anuradhapura and Polonnaruwa districts

Although necessary actions have been taken by the government and health care sectors to reduce the occurrence of CKDu in the country, especially in north central province, there are still room for development and consideration of supply of quality water and educating the people on the reason for the unfortunate events they have been facing due to the usage of contaminated water and hard water.

1.2.3 Forecasting water quality parameters

In Sri Lanka, lots of people depend on public water sources for drinking purposes in their routine life. The National Water Supply and Drainage Board of Sri Lanka has a major responsibility to make sure the quality of the water sources and identify them as safe to consume for drinking or agriculture. People who are in the dry zone area, they have been suffering from the hardness of water that may cause water-borne disease.

The shortage of the drinking water that has been occurred in many dry zone areas according to the Disaster Management report in Sri Lanka. There are 337 000 people across the eight out of the twenty-five districts of Sri Lanka which are affecting the water shortage due to sea water intrusion into ground water and dry spell [4]. Therefore, rapidly increase the consumer level of certain water resources, it also affects the water quality parameter in the future, for that case we need to check the level of water quality parameter in advance. The changing level of the water quality parameters that impact the safeness of the water. In order to avoid those reasons, we need to know the water quality parameter level in preemptively. The Department of Meteorology in Sri Lanka reported seasonal climate changes during March to May is affected the water quality parameters because of the dry spell.

The major problem that was identified in 2003, the intrusion of the seawater into the groundwater system which impacts the water quality parameters, therefore most of the people moving to common water sources which means water plant that is provided by the National Water Supply & Drainage Board in Sri Lanka. That was the huge highlighted issue for people who depends on groundwater. It happened in the dry – zone area which is located nearby seas.

Central Environment Authority (CEA) [5] in Sri Lanka has been conducted numerous water quality monitoring program. They have been tested the water quality parameters in different water bodies which is evaluated by the Canadian Water Quality Index (CWQI) method for monthly basis in selected location. The monitoring parameters are included Electric Conductivity (EC), pH, Turbidity, Dissolved Oxygen (DO), Temperature and Total Dissolved Solid (TDS). These are the parameters that are influenced the quality of the water. The dissolved materials carry heavy metals such as Chromium, Leas, Microbiological contaminant, and Nutrient etc.…

1.2.4 Machine learning models for water quality prediction

According to the Linkoping University, department of applied physics, Prof. Fredrik Winquist was the inventor of the voltammetry based electronic tongue concept [9] which indeed was an inspiration for the development of an electronic tongue based smart device to address an ongoing environmental issue. Even though the concept was used in many industrial purposes like pharmaceutical drug testing and whisky taste testing, a chemical model to analyze the performance of the electrodes of a custom made voltammetric e-tongue for the evaluation of infused tea gives a sole chemical approach to the idea behind the voltammetry based electronic tongue concept [10][11] which then provided the way for more advanced mathematical model implementations for identification of tea samples [12].

According to Shafi U in [13], classical Machine learning algorithms like Support Vector Machines (SVM), Deep Neural Networks (Deep NN) and Neural Networks (NN) were used to measure the water quality with a highest accuracy of 93%. This level of accuracy distinctly indicates the importance of training the selected mathematical model under both controlled and open field conditions. It can also be stated that NN, Deep NN and Machine learning models are highly fitting in training a model which includes complex functions. Furthermore, it is important to note here that out of 30 water quality variables which were defined by World Health Organization (WHO), 25 variables are used in order to achieve this highest accuracy. However, using 25 different types of sensors makes this system economically infeasible due to budgetary restraints. Sakizadeh, M [14] used 16 water quality parameters along with an Artificial Neural Network (ANN) with Bayesian regularizations. This study capitulated correlation coefficients between the observed and the predicted values of 0.94 and 0.77. Even though the reduction of number of used variables didn't impact on a vast difference in the accuracy and mean error, using 16 sensors on respective variables puts the progress of the study into a tight spot.

In a comprehensive overview [15], the study suggests a different and more efficient and scalable approach when it comes to selecting the variables for the WQI calculation. Once a data set has been collected, it is initially passed through a Principal Factor Analysis (PFA) where all the variables present in the dataset will be preprocessed to select the best suiting variables while preserving the overall variance as much as possible. This step addresses a principal issue when it comes to designing the device with using suitable sensors that could be within the planned budget.

This paper also states the importance of parallelly training multiple algorithms, selecting the best algorithm with the highest accuracy and a least mean error to proceed with an effective mathematical model for a reliable WQI output [15][16].

Yuanyuan Wang et al… in 2017 [3] was conducted the research. Mainly it has focused on water prediction and prevention of water pollution besides it was a time series prediction. This developed system has been used as a new method which is based on LSTM (Long- and Short-Term Memory) Neural Network for predicting quality of water inaccuracy in surface water in Taihu Lake.

1.2.5 A comprehensive study of seasonal variation in groundwater quality of Sagar city by Principal Component Analysis.

Hemant Pathak et al… [9] While gathering data on seasonal variation and get an idea about how the seasonal variation impacts the water quality, there is another research article was found [9]. In 2015, fifteen sampling centers were taken for investigation about chemical parameters which were conducted on pre-monsoon, monsoon, and post-monsoon seasons [9]. The result of this research identifying water quality parameters which were impacted by seasonal variation.

1.2.6 Time Series Forecasting of Water Quality of River Godavari

prof B.S.N.Raju et al… In addition, there is another kind of research in time series model for forecasting of water quality of the River Godavari, this research was done to forecast monthly basis a single water quality parameter as Dissolved Oxygen in the water of river Godavari [10]. Time series analysis of past water quality data was learned to predict future values. For each water quality parameter, they were calculated minimum, maximum, mean, standard deviation, and variation for Actual measured, past (2009 to 2012), and future (2012 to 2015). After that, the various values of actual, past, and future were compared with each other then made some conclusions such as there are some variations for past water quality values and the actual values the reason is the damage of the water in the current period. It provides the conclusion that water quality parameters are affected by seasonal variations and trends [10].

1.2.7 Forecasting of River Water Quality Parameters

Mosin I Hasan et al… did research that has been done in the forecasting of parameters of river water quality, through this research, river pollution has been prevented. The water quality parameters such as pH, Temperature, Turbidity, conductivity, dissolved oxygen in the river were predicted and forecasted based on the time series analysis method and ARIMA modeling [11]. Burnett River is a river that is in Australia and its water quality parameters dataset of the year 2015 was given by the Australian government for this research. By using the dataset, the machine learning model was created and forecasted future water quality parameter values and according to that values, the government bodies can take necessary actions in the earliest stage [11].

1.3 Research Gap

A thorough background study suggests that the lives of people who are consuming unsafe water are at risks of getting exposure to major health issues. Even though industries are taking necessary actions to provide safe water, majority of people are leaning towards untreated ground water due to the abundance of ground water resources, financial reasons and to avoid consumption of chlorinated water. It is prime time to educate the people on the quality of the water that they are dealing with. The lack of a smart device to help people find out the ground water quality using a simple water sample to avoid manual and time-consuming laboratory processes is a main goal for the research.

Even though Sri Lanka is in the verge of becoming more advanced in water quality researches and predictions, there have been still no researches done in the ground water quality analysis. According to National Building Research Organization (NBRO), more attention has been paid to surface water since the surface area that are present is high but the ground water sources have been comparatively neglected. Not only the ground water quality, but also the consequences that people face due to consumption of these unsafe water must be highlighted. Also, seasonal variation can change the quality of the water parameters which mainly affects the quality of water. Therefore, a smart device that could be used in real time identification of water quality, disease outbreak possibilities and the water quality parameter changing rate due to seasonal variation could be an effective solution for an ongoing problem.

As clearly explained in the background study and the literature review above, several attempts have been taken in the recent past when it comes to water quality prediction in some developed countries. Although its important to live clean and healthy, it is never too late to introduce the concept of WQI in Sri Lanka to assure the better quality of any kind of water found in sources.

Not only this concept is new to our country but also it is a bit difficult in practically applying it to every water sample that is about to be tested.

Table 1: Comparisons of existing systems

Features	MeraBhu jal[9]	Time series analysis [8]	Water Quality 4Thai [10]	E-Tongue
Sensors	✓	✗	✓	✓
Sensor Calibration	✗	✗	✓	✓
Real-time	✗	✗	✗	✓
Location wise prediction	✓	✓	✓	✓
Seasonal wise prediction	✗	✓	✗	✓
Forecasting water quality parameter	✗	✗	✗	✓
Predict WQI	✗	✗	✗	✓
CKDU prediction	✗	✗	✗	✓

1.4 Research problem

A healthy lifestyle could be possibly ensured once the day to day activities are made clean and clear. With an abundance of ongoing research and explorations regarding water sources, a bit more attention must be paid in analyzing groundwater. Since the techniques that are used in Sri Lanka is manual and time consuming, an innovative solution must be put forward to identify the quality of ground water in real time. Moreover, a prediction of the changing rate of water quality parameters due to seasonal changes too plays a vital role in determining the characteristic of the selected water source in the future. This water quality changing rate prediction could be used to predict the possibility of outbreak of any water borne diseases. This solution could be used mainly by a single user or an organization in order to identify the ground water quality of specific location which will eventually lead the way for a better future.

While doing research in this domain, some interesting research problems have been found out and they have been listed below.

- How to identify the water quality parameters using IoT sensors and connect them into mobile application?
- How to predict the quality of the water and confirm the safe consumption via WQI?
- How to create the smart app to predict the risk of exposing to the CKDu using supervised Learning?
- How Machine Learning technologies are used to predict the water quality parameters in advanced based on the user location?

To overcome the above-mentioned problems, data are gathered from the National Water Supply and Drainage Board in Sri Lanka. Then, machining learning algorithms have been developed to predict water quality parameters and to identify the quality of water sources based on location and predict the risk of the CKDu. Through this research, the awareness has been created among people who are affected by bad quality of the water.

1.5 Research Objectives

1.5.1 Main Objectives

The ultimate intention of this research is to design a smart device that could be used to identify the ground water quality by examining a water sample. A Water Quality Index (WQI) will be an output from the mathematical model to display the range to which the WQI belongs, to the user through a mobile application. In order to make the device, a combination of sensors, an accuracy increasing algorithm along with a mathematical model must be combined to make it function as intended.

1.4.2 Sub Objectives

To develop an IoT device by combining water quality parameter sensors to obtain readings to be stored in cloud.

- Transfer the raw sensor data to the cloud platform and store.
- Implement data stream pipelines to ingest data to mathematical models and achieve results.
- Process the ingested data to select an appropriate algorithm and train the algorithm to select the best Machine Learning Algorithm in order to obtain high accurate WQI.
- Forecasting water quality parameters based on the user location and checking the seasonal variation.
- Predict the risk of exposure to CKDu and water borne diseases.

2. METHODOLOGY

The methodology is used to handle the main and sub-functions of our research approaches that follow the software lifecycle model to implement the system. It portrays a smart way to solve the research problems that are raised by us. The result of this system is a smart intelligent tool to predict the safe consumption of the groundwater by using ML, AI and IoT techniques which is used to come up with the solution.

2.1 System Overview

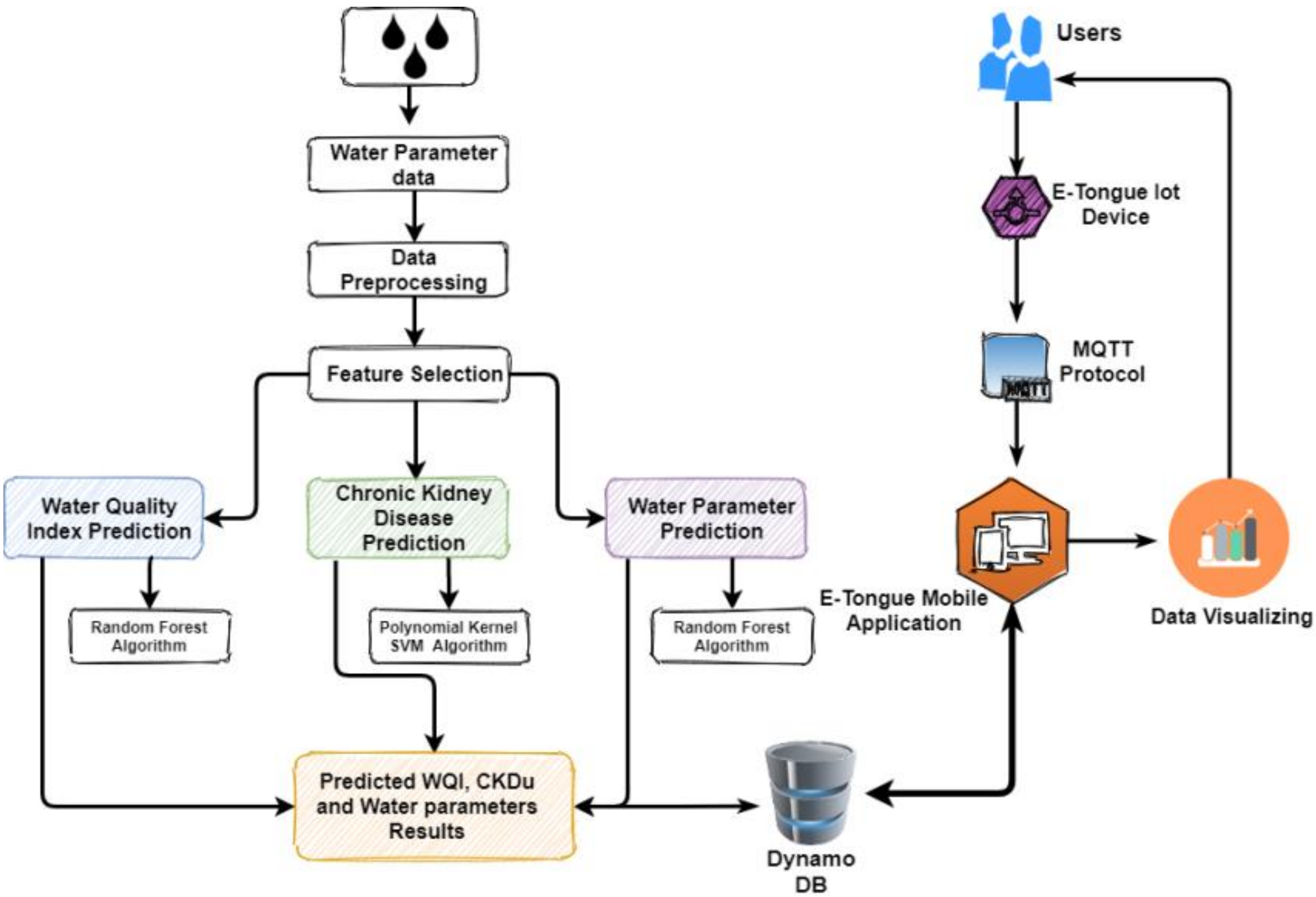


Figure 4: System architecture

E-Tongue is a smart tool to predict the quality of ground water for a safe consumption and usage. This intelligent system consists of 4 main functionalities:

- Developing a smart gadget to get water samples for predictions purposes.
- Predicting the WQI value for ground water.
- Forecasting water quality parameters.
- Envision to CKDu outbreak and water borne diseases.

Identification of the water quality functionality focuses on predicting the Water Quality Index value using several input parameters using supervised machine learning techniques. Several mathematical models were trained using the dataset obtained from the authorities to compare and select the best model that have high accuracy and low error values. Upon selecting the most accurate model to be used in the system, the system uses the sensor readings to predict the WQI of the used water sample in real time.

In the meantime, forecasting water quality parameters includes the location of a certain source to forecast the parameters and the parameter readings are used to predict the possibility of any CKDu and water borne diseases using machine learning techniques.

As the identification of water quality plays a crucial part in this project, this particular component is divided further into sub categories to achieve a reliable and efficient solution: Selection of the best machine learning regression model to predict the Water Quality Index (WQI).

Prediction of the Water Quality Index of the ground water sample using the selected machine learning model.

Making suggestions on the methods of purification to follow based on the predicted WQI values.

The system tends to store the readings obtained from the device in a centralized database where all the components can access the data. The real time prediction of the water quality and the disease outbreak possibilities makes this system outstanding from the ones that are being practices and used in the industries in the present time.

2.2 Component Overview

2.2.1 Developing a smart gadget to get water samples for predictions purposes.

Water safety estimating appliance was to implement as shown in figure 9, In this implemented methodology, a nodemcu microcontroller is utilized as the fundamental controller of the unit and Arduino Nano microcontroller for control purpose of sensors and for data storing dynamo database was utilized. As we are targeting this for normal end-users the device should be affordable. The framework capacities naturally and freely as per the code transferred to the microcontroller. Here, five sensors are utilized to measure the fundamental parameters. Water pH level, turbidity (darkness), temperature, turbidity, electric conductivity, and total dissolved solids to measure groundwater.

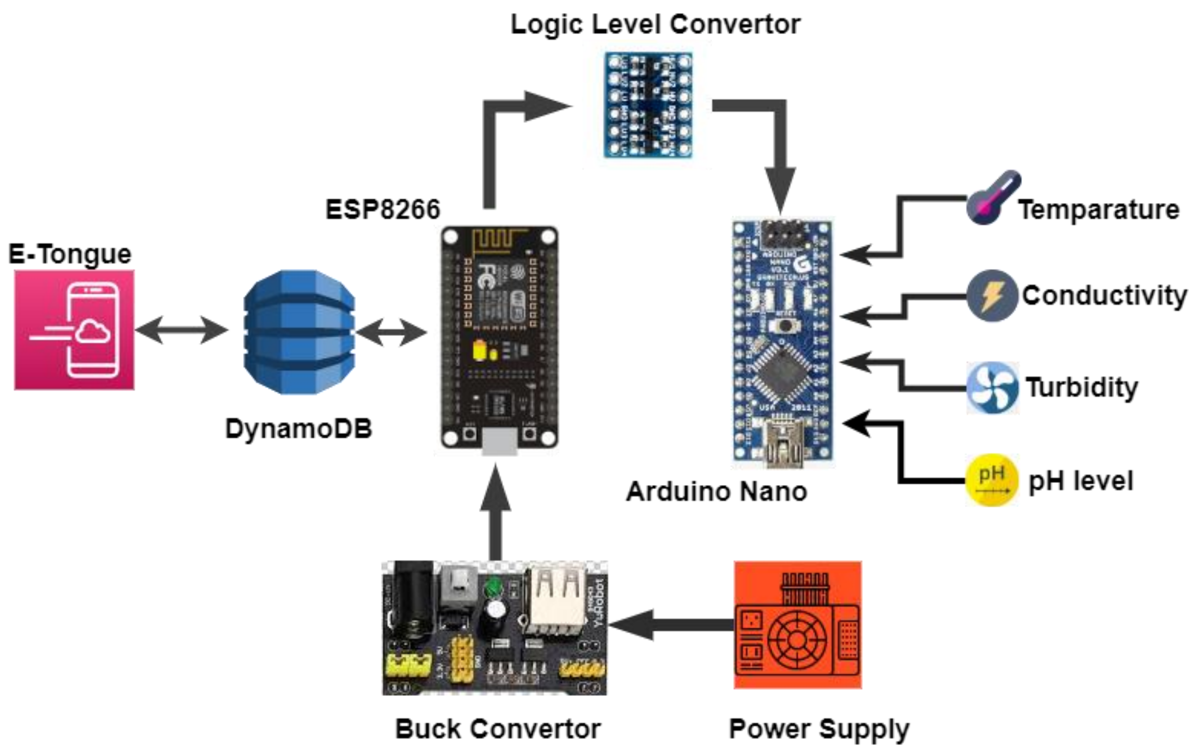


Figure 5: Component system architecture

2.2.2 Predicting WQI

Identification of the quality of ground water by predicting WQI value involves analyzing and obtaining the reading of the water quality parameters from the parameter sensors that are used in the fashioned device. The readings from the sensors acts as the input for the prediction of the single numeric index called Water Quality Index (WQI) which will be used to identify the quality of the water sample that have been used.

The problem that has to be solved through this particular component is to implement a functioning mathematical model that could be used to predict the quality of the ground water sample and to display the results and precautionary steps that could be followed to make the sample water more suitable for drinking.

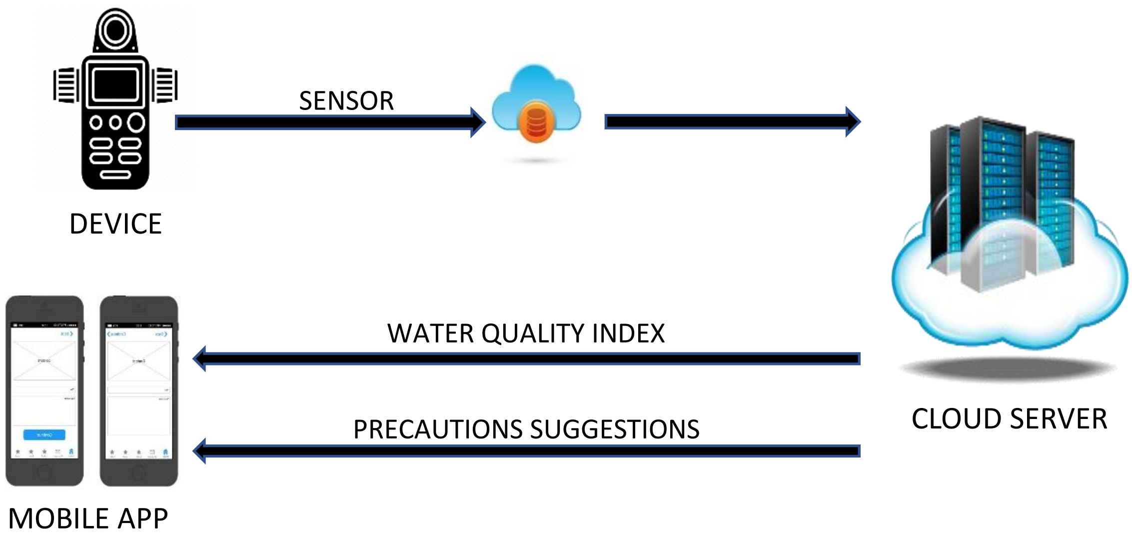


Figure 6: WQI system diagram

2.2.3 Forecasting Water quality parameters

According to the conclusion from the literature review, technology selection and software solutions are the most important part of this research. Forecasting water quality parameters in the future is one of the components in this proposed system. Consumers are able to know the water quality parameter before. Further, they able to identify the nearby quality of water sources based the result of this component will be forecasted water quality parameter value for the future. When the user locates their location or searches the known location, the application will be shown the water quality parameter for the future. Here, location is the main input for getting the output of this feature. This input sends to a model which tries to forecast water quality parameter by using historical water quality data through a Machine Learning algorithm. The trained ML model will be stored in the AWS.

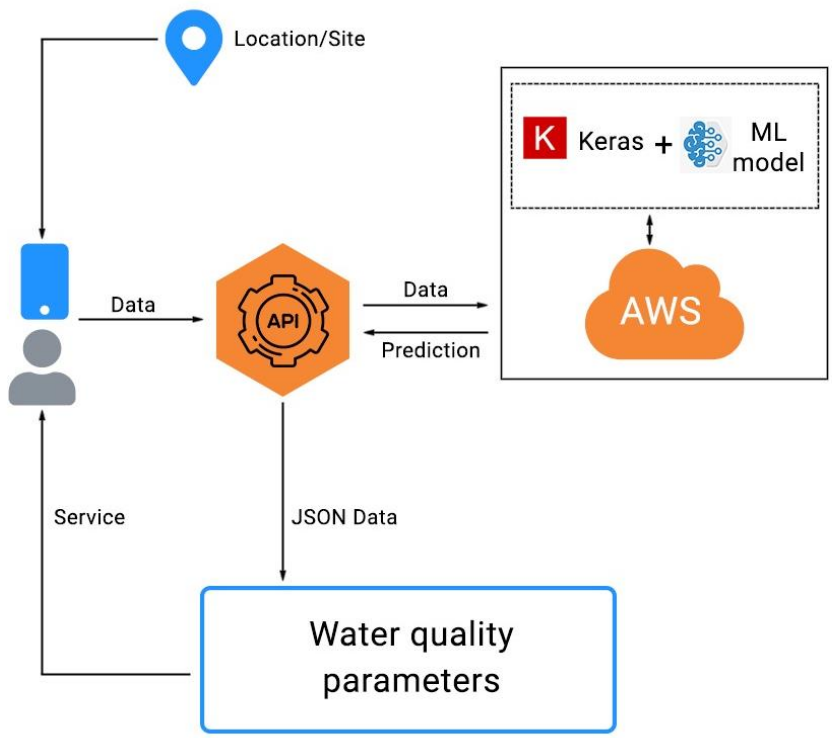


Figure 7 : Forecasting water parameter system diagram

2.2.4 CKDu outbreak Prediction

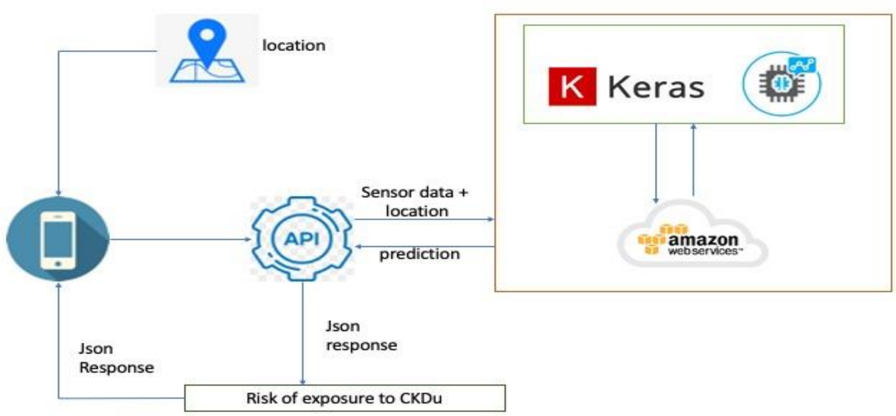


Figure 8: CKDu prediction system diagram

The main objective of this individual component is to alert user if there is a possible risk of exposure to CKDu in each ground water sample. So as to acquire the functionality the solution is provided with the usage of supervised learning techniques. The location of the user is fetching via the mobile application using Google Map API and the sensor data of the water sample that gathered using the IoT based device is obtained from the AWS database instance. The location and the sensor data are sent to the trained model via RESTful web API and the predicted output is send back to the mobile application using the web API. User can access the output by mobile application and take the precautions according to the results.

Development of sensor circuities.

<u>pH Sensor</u>

pH sensor module diagram

After all this probe are not feasible to be straight forwardly associated with the microcontroller as a result of its low voltage, the pH module circuit utilized in this task. This module circuit is an advance variant of the earlier versions of pH module circuit where the operational components and the segments are changed concurring to the prerequisites [31].

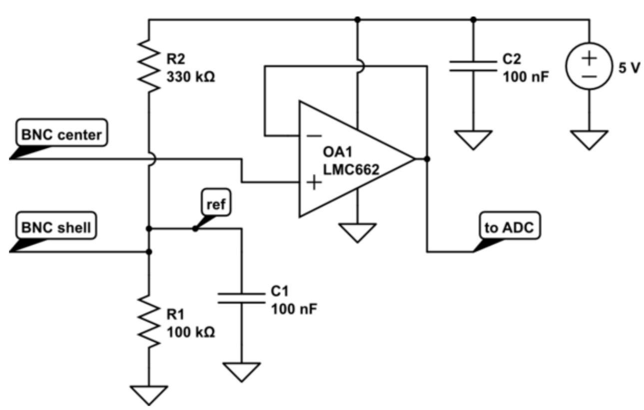


Figure 9 : ph module circuit diagram

Calibration of pH sensor

For the calibration, voltage acquired is linear to the pH, focuses for example pH 4 and 10 their separate voltages were gotten utilizing the known solutions. Table 4 and Figure 12 shows the adjustments outputs.

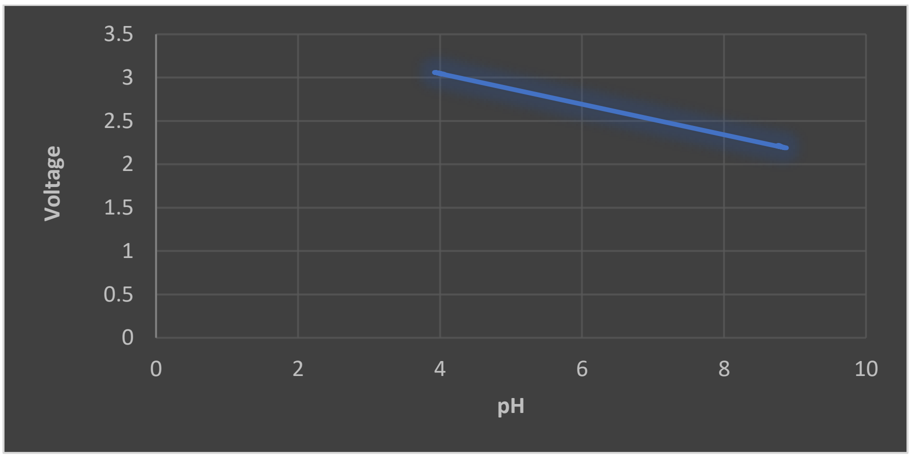


Figure 10 : Graph of ph vs voltage

Table 2: ph values with respect to voltages

ph	voltage
4.06	3.04
3.94	3.06
4.03	3.04
4	3.05
3.92	3.06
8.77	2.22
8.83	2.2
8.88	2.19

Coding part of pH sensor

```
void readPh() {
  for (int i = 0; i < 10; i++)
  {
    buf[i] = analogRead(PHPIN);
    delay(30);
  }
  for (int i = 0; i < 9; i++)
  {
    for (int j = i + 1; j < 10; j++)
    {
      if (buf[i] > buf[j])
      {
        pHtemp = buf[i];
        buf[i] = buf[j];
        buf[j] = pHtemp;
      }
    }
  }
  avgValue = 0;
  for (int i = 2; i < 8; i++)
    avgValue += buf[i];
  float pHVol = (float)avgValue * 5.0 / 1024 / 6;
  phValue = -5.70 * pHVol + calibration;
}
```

Figure 11 : Implementation of ph sensor

The pH variable is characterized as global parameter and so as to compute the pH value through sensor reading, 'readPh()' functionality is characterized which does the accompanying:

1.When 12v given to the unit pH circuit turns on.

2. It gather 10 consecutive values from analog pin A3 from nano microcontroller.

3. Next ordering values taken and discarding the highest and the lowest of the values and ***c***alculating the average with the six remaining samples.

4. The 'pHVol' is then calculated by converting the above value to voltage.

5. Using ph references ‘pHVol’ is converted to ‘phValue’.

Temperature Sensor

Temperature probe circuit

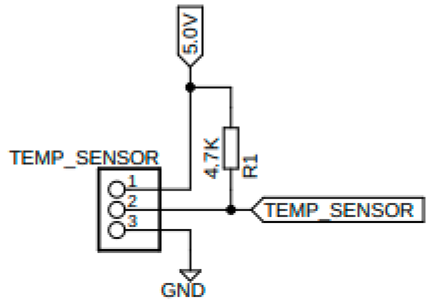


Figure 12 : Circuit of temperature probe

The sensor functions with the method of 1-Wire communication. It need only the data pin connected to the microcontroller. It has a pull up resistor (4.7K) and the other two pins are used for power purposes ground and 5V as shown above. The 4.7K resistor is utilized to keep the line in high state when the controller bus is not being used. The temperature sensor reading estimated by the sensor will be put away in a 2-byte register inside the sensor. This information can be perused by the utilizing the 1-wire strategy by sending in an arrangement of information.

Calibration of temperature sensor.

For sensor calibration, you need to gauge something of which you know the temperature. The easiest method to do it at home is utilizing boiling water and a shower of ice, additionally called a "triple-point" shower. Remember that to make an exact estimation you should know atmospheric pressure and compute the best possible boiling temperature there.

Discerning the value that the sensor read and the value that ought to be, we can change the basic estimation of the DS18B20 into something more legit.

Fill up a container with water and heat up until it boils. Next, submerge the sensor probe and keep it for couple of seconds and take readings to the serial monitor. Mark this reading as ‘Basic_High’. Afterwards fill up another container with small ice cubes and ice water. Clean up the sensor probe and submerge it in ice container and set that reading as ‘Basic_Low’. References measured at this point. For boiling point, it was 99.9 Celsius and melting ice 0 Celsius. Set the basic range and reference range by taking the difference of the values. The following formula was built up with the calibration of the sensor.

$$\boldsymbol{Precise\,Value} = \left(\frac{\left(\left(\boldsymbol{Sensor_{Reading}} - \boldsymbol{Basic_{Low}}\right) * \boldsymbol{ReferenceRange}\right)}{}\boldsymbol{BasicRange}\right) \qquad (1)$$

Coding of the sensor

```
#include <OneWire.h>
#include <DallasTemperature.h>
// Data wire is plugged into pin D9 on the Arduino
#define ONE_WIRE_BUS 9
// Setup a oneWire instance to communicate with any OneWire
devices
OneWire oneWire(ONE_WIRE_BUS);
// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);
/
*****************************************************************
****/
void setup(void)
{
 // start serial port
 Serial.begin(9600);
 Serial.println("Dallas Temperature IC Control Library Demo");
 // Start up the library
 sensors.begin();
}
void loop(void)
{
 Serial.print(" Requesting temperatures...");
 sensors.requestTemperatures(); // Send the command to get
temperature readings
 Serial.println("DONE");
|
 float precise_value = (((sensors.getTempCByIndex(0) -
Basic_low) *ReferenceRange) / BasicRange)
 Serial.print("Temperature is: ");
 Serial.print(sensors.getTempCByIndex(0));
   delay(1000);
}
```

Figure 13: Implementing of temperature sensor

‘DallasTemperature.h’ and ‘OneWire.h’ libraries are included here. Through this sensor we can directly get the reading in digital form. The only data pin is linked to Arduino nano digital pin 9.

Set up the instances where it used to communicate with the sensor.

Firstly sensor is loaded using the sesnors.begin() .

It opens a function where it fetches for temperature using ‘sensors.requestTemperatures()’ . it works has a listener function.

The probe data read byte and byte and convert them to and analog data pattern and stored in sensors.getTempCByIndex(0).

At last precise value is taken by sending above value to the formula.

The value stored in ‘temperature’ is then sent to AWS DynamoDB to show the temperature of water in mobile interface.

<u>Turbidity Sensor</u>

Turbidity sensor module circuit diagram

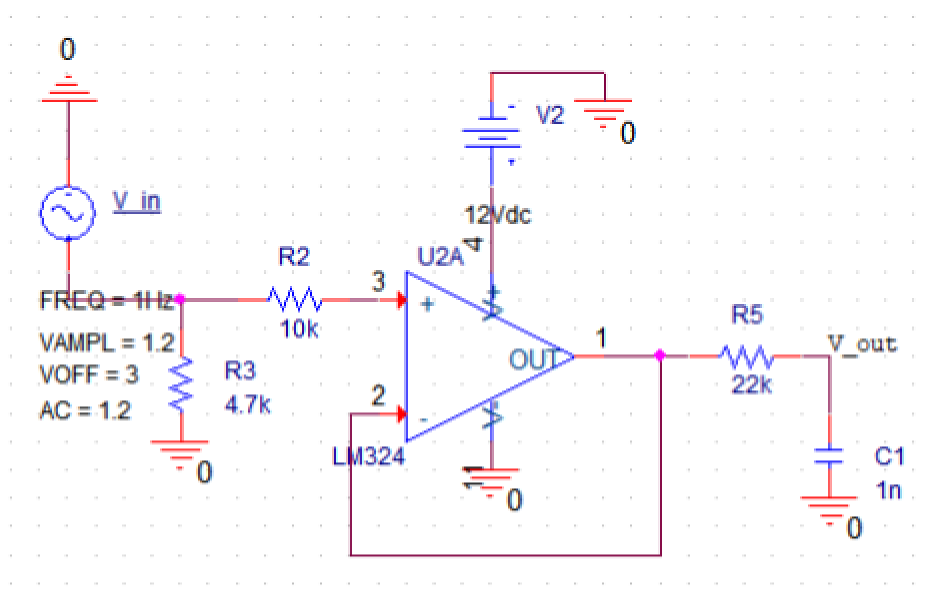


Figure 14: Circuit of turbidity module

The Arduino turbidity sensor recognizes water quality by estimating the degrees of turbidity, or the obscurity. It utilizes light to identify suspended particles in water by estimating the light conveyance and dispersing rate. A few alterations are done to improve execution. Likewise, operational amplifier LMV358 is utilized, furthermore this is simpler to be patched physically.

Turbidity module which fills in as a segregation between probe and Arduino broad by connecting analog wire A6 pin. With this connector circuit, turbidity value will be more accurate. V_out in the circuit, will be associated with analog pin A6. Reading will be available in voltage. So, we have converted it from voltage to nephelometric turbidity units(ntu).

Calibration of turbidity sensor

The turbidity is adjusted with the assistance of tube. It is a proficient and minimal effort strategy to acquire the turbidity in water. This built using a glass chamber with a distance across of 3cm, a print height unit and a review plate. It can gauge turbidity from 0 to 280 ntu. The following equation is used to test changes over the length in cm to turbidity in ntu.

$$Depth\ in\ CM\ =\ 244.14 * (Turbidity)^{-0.662} \qquad (2)$$

We prepared a solution mixing water with milk. Which will be utilized as adjustment answer for the turbidity sensor. At that point, the sensor's test was embedded in various examples, giving the yield voltage at various turbidity. To improve exactness, the first analog information from Arduino (0-1024) is utilized for calibration rather than voltage. A table of simple incentive at various turbidity was developed. The figure 15 shows the procedure to follow to capture these readings.

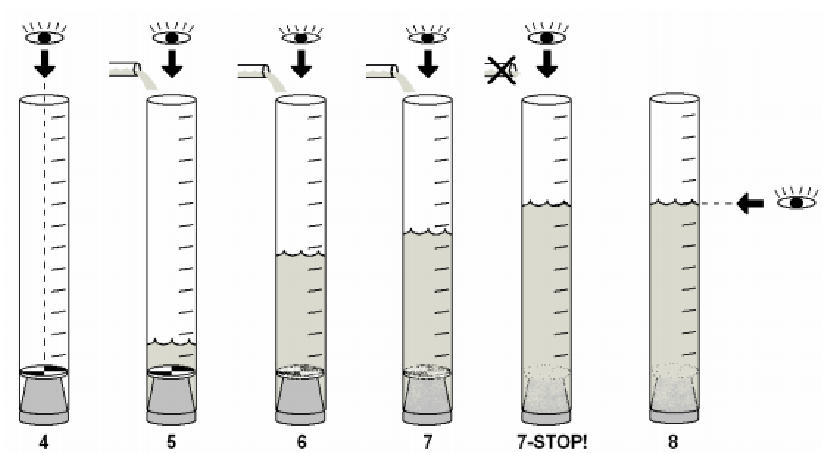


Figure 15: Turbidity tube

Table 3: Dataset with length and turbidity levels

Length/cm	Turbidity/NTU	Analog Value
6	280	495
8	175	565
11	108	637.5
13	84	663.5
15	68	689.5
17	56	706
18	51	714
19	47	721
20	44	730
21	41	735.5
23	35	748
24	33	750
25	32	755.5
26	31	762
27	30	765

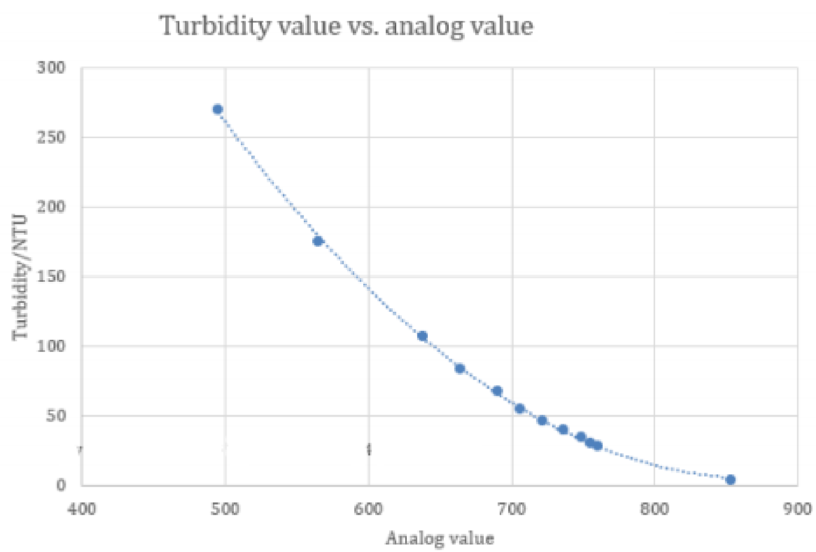


Figure 16: Graph of analog value vs turbidity

From the figure 18, it very well may be seen that simple incentive from 766 to 495 relates to 30~280 Ntu. The fluctuation is excluded by taking the mean of six continuous measurements. The connection between turbidity and analog value is in equation 3.

$$y = -1120.4\,x^2 - 5742.3x - 4353.8 \qquad (3)$$

The adjustment of turbidity sensor was done at 26.3ºC steady temperature, anyway at various temperatures, the analog output can shift generously. As per figure 19, condition temperature will change the yield voltage of the probe, the slant for analog output are - 2.05 ºC.

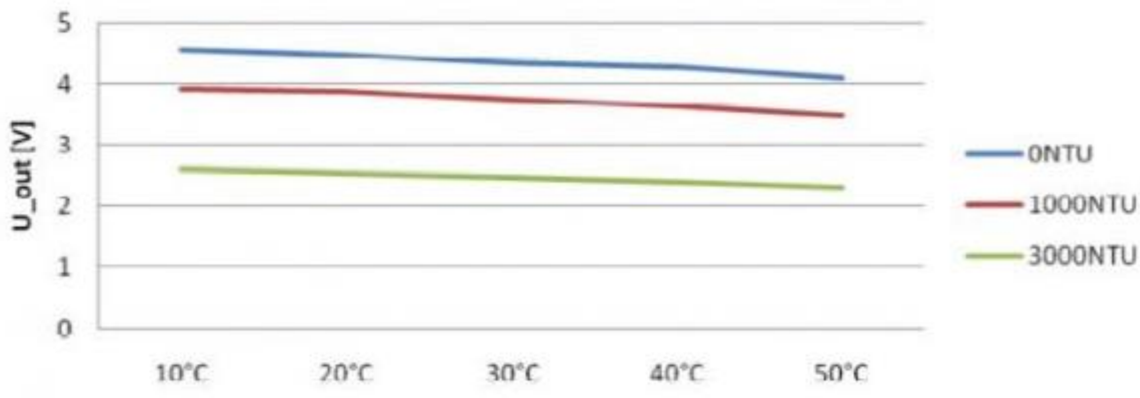


Figure 17: Voltage differences

If the steady temperature is 26.3 then the equation 4 will build up in this manner.

$$Precise\ analog\ value\ \approx\ actual\ analog\ value\ +\ (T\ -\ 26.3)\ \times\ 2.05 \qquad (4)$$

The turbidity probe is adjusted in a room at 26.3ºC and the water is 26.3ºC too, so the alignment condition is substantial since given that the water is at 26.3ºC, the temperature must be mulled over to ensure the test's precision is independent.

The yield from turbidity sensor in arrangement at fixed turbidity 100 NTU, when the temperature is between 26 ºC to 35 ºC is appeared in table underneath.

Table 4: Temperature dataset

Temperature	Analog Value
26.6	631
28.5	627.3
29.4	624.1
30.4	621.5
32.3	617.2
33.4	614.2
34.1	613.3
34.7	612.8

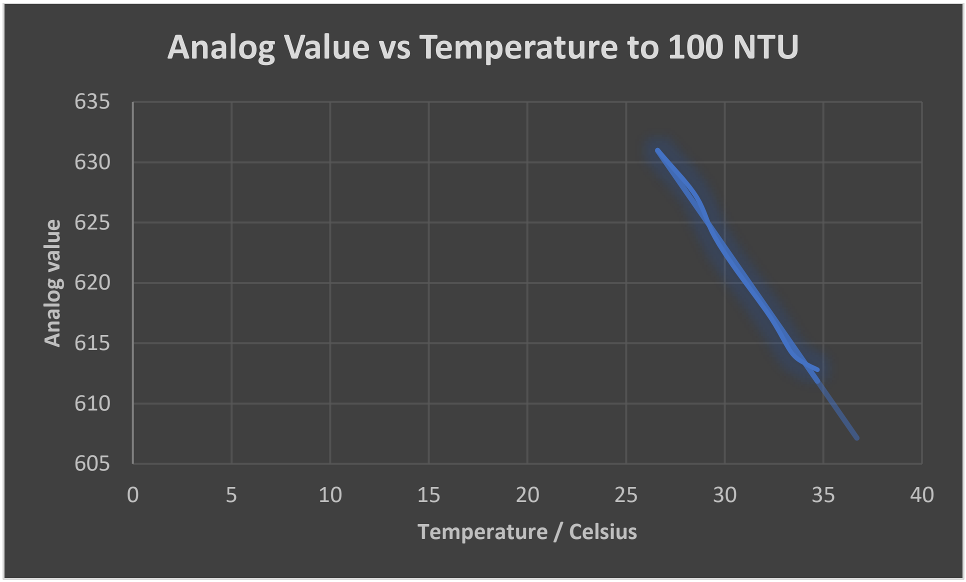


Figure 18: Graph of temperature vs analog value for 100NTU

As expressed in table 6, a chart is outlined to discover the connection among analog values and temperature. It is seen that the information focuses structure a straight line with a slant of - 2.3562. It is close to the reference esteem (2.05). The underneath equation is utilized to change over the yield at any temperature.

$$Precise\ analog\ value\ \approx\ actual\ analog\ value\ +\ (T\ -\ 26.3)\ \times\ 2.3562 \qquad (5)$$

By taking temperature into thought, the polynomial condition equation 5 is joined with temperature adjustment condition equation 6, which becomes:

$$Turbidity\ (NTU) = -1120.4\ \times\ (x\ +\ (T\ -\ 26.3)\ \times\ 2.3116)\ 2\ +\ 5742.3\ \times\ (x\ +\ (T\ -\ 26.3)\ \times\ 2.3116)\ -4353.8 \qquad (6)$$

Coding of the sensor

```
void readTurbidity() {
  turbSensorVal = analogRead(TURBIDPIN);
  turbSesnorVal = turbSensorVal + (temp - 26.3)* 2.3562
  turbVolt = turbSensorVal * (5.0 / 1024.0);
  if (turbVolt > 4.20024)
    turbVolt = 4.20024;
  turbidity = -1120.4 * sq(turbVolt);
  turbidity = turbidity + (5742.3 * turbVolt);
  turbidity = turbidity - 4352.9;
}
```

Figure 19: Implementing turbidity sensor

The following are the process of handling turbidity sensor.

1. power on the circuit to capture sensor readings.

2. Through the analog pin A6 analog value is taken to 'TurbSensorVal' variable.

3. To eliminate the impact of temperature its use the equation X.

4. Finally with calibration equation analog value is converted to turbidity value in the units of ntu.

<u>Conductivity Sensor</u>

Conductivity Circuit Diagram

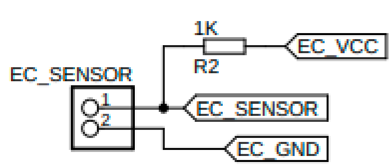


Figure 20: Conductivity circuit

Conductivity is the capacity of a mixture, a metal to pass an electric flow. Current is conveyed by cations and anions while in metals it is conveyed by electrons. In liquid plans the level of ionic quality contrasts.

Conductivity might be estimated by applying a rotating electrical flow (I) to two cathodes drenched in a solution and estimating the subsequent voltage (V). The cations move to the negative terminal, the anions to the positive cathode and the arrangement act as an electrical transmitter.

Hence the conductivity sensors in the market and much expensive and due to covid pandemic we were unable to purchase this conductivity sensor. We had the basic idea behind the sensor, at last we decide to design and build a conductivity sensor.

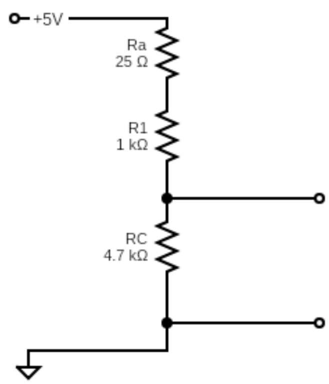


Figure 21: Resistance flow

The functionality of the circuit diagram will be discussed here.

Ra = 25Ω - Internal resistance in Arduino pin.

R1 = 1000Ω - This resistance used for voltage divider circuit. The value of this resistance defines the resolution of the measuring range of the sensor.

Rc – Calculated resistance between two terminals.

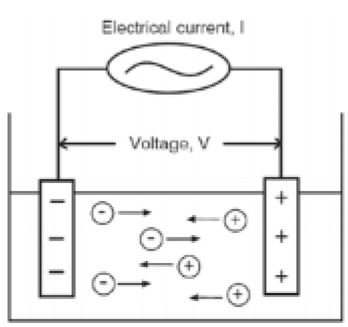


Figure 22: Voltage difference between probes

Using Ohm's law, we can determine the water resistance (R).

$$R = \frac{v}{I} \qquad (7)$$

The reciprocal of the resistance (R) of a water between two electrodes can be used define conductance (G). In the above equation 'V' and 'I' are voltage and current, respectively. Conductance measured in Siemens (S).

$$G = \frac{1}{R} \qquad (8)$$

The proportion to the displacement (D) between the electrodes to the area (A) of the electrodes known as cell constant (K).

$$K = \frac{D}{A} \qquad (9)$$

This demonstrates particles in arrangement will direct electricity. The ability of conductivity is to pass electricity. The conductivity perusing of a testing samples will change with temperature. The conductivity (C) is characterized as:

$$C = K * G \qquad (10)$$

Calibration of the sensor.

The adjustment cycle of the sensor incorporates two arrangements with 1.515 (mS) and 10.86 (mS) conductivity. These arrangements are reasonable for estimating conductivity with K=1.76 test.

The alignment cycle follows the beneath steps:

1.Dip the temperature probe in the solution for at any rate 1-2 minutes all together for the arrangement to balance out its temperature.

Spot the conductivity sensor into 1.515 (mS/cm) arrangement and mix the solution.

Ensure the arrangement is settled at 26.3°C degree and watch the acquired conductivity voltage.

Subsequently the test the setup inside the 10.88 (mS/cm) arrangement and like the last advance the conductivity arrangement ought to be watched also, recorded.

The graph can be plotted using above values for the temperature of 26.3 ^{o}C.

While playing out the alignment cycle, a steady voltage was gotten from every arrangement table 7. The alignment chart was portrayed dependent on the acquired outcomes from the beneath table.

Table 5 : Conductivity dataset

Solution	Voltage
1.515	392.34
10.88	3350.3

Coding

```
void readEC() {
  pinMode(ECVCC, OUTPUT);
  pinMode(ECGND, OUTPUT);
  delay(30);

  digitalWrite(ECVCC, HIGH);
  digitalWrite(ECGND, LOW);
  ecSensorVal = analogRead(ECPROBE);
  digitalWrite(ECVCC, LOW);
  pinMode(ECVCC, INPUT);
  pinMode(ECGND, INPUT);

  ecVolt = ecSensorVal * (5.0 / 1024.0);
  Rc = (ecVolt * R1) / (5.0 - ecVolt);
  Rc = Rc - Ra;
  EC = 1000 / (Rc * K);
  EC25   =  EC / (1 + tempCoef * (temp - 25.0));
  ppm = (EC25) * (PPMconv * 1000);
}
```

Figure 23: Implementation of conductivity sensor

Communication Structure

Overview

The communication medium to transmit sensors data to the user is achieved through WI-FI. In here, we have implemented two broads Nodemcu ESP8266 and Arduino Nano. For data communication within the broads done via serial communication using a logic level convertor. At last, transmission between the smart tool and the AWS DynamoDB built up using Message Queuing Telemetry Transport (MQTT) protocol.

Coding

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include "FS.h"
#define WIFI_SSID "###"
#define WIFI_PASSWORD "#######"

const char* AWS_endpoint = "a2w43nj3we329s-ats.iot.us-
west-2.amazonaws.com";

char receivedData [256];
const size_t jsonCapacity = JSON_OBJECT_SIZE(10) + 30;
char msgOut[256];

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();
}

WiFiClientSecure espClient;
PubSubClient client(AWS_endpoint, 8883, callback, espClient);

void setup() {
  Serial.begin(9600);
  Serial.setDebugOutput(true);
  pinMode(LEDPIN, OUTPUT);
  digitalWrite(LEDPIN, LOW);
  setup_wifi();
  delay(1000);
}
```

Figure 24: Implementation of MQTT

2.3.2 WQI prediction

Model training using Regression Algorithms

In machine learning, the problem could be divided into 2: Regression and Classification. In general, regression normally helps to predict continuous quantities whereas classification helps to predict discrete class labels. This problem identified through this component could a regression problem solving as the objective is to predict a continuous quantity in other words, the output that is predicted is a single numeric index.

As stated above, among the several algorithm options that are present, it is vital to choose the best algorithm that could ensure a reliable output. Selecting such models needs a continuous evaluation of their performances and the percentage of the error that it offers.

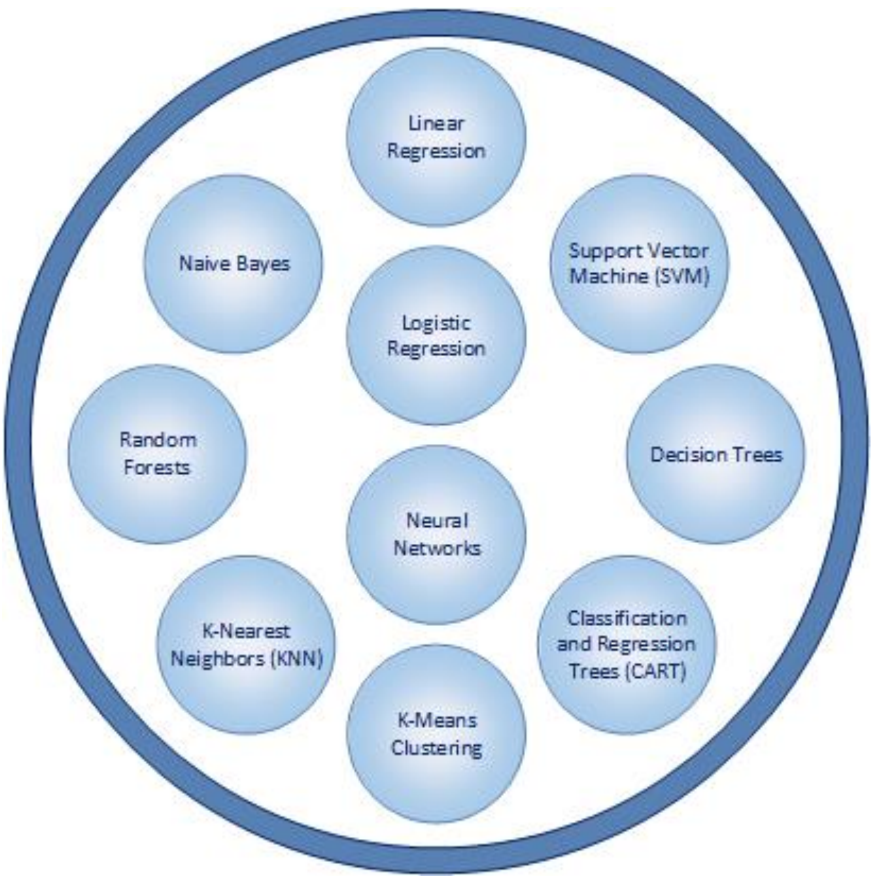


Figure 25: Commonly used Machine Learning algorithms and techniques

It was decided that testing out as many as the algorithms to identify the best one could be a great task because different regression algorithms has different characteristics and, studying and understanding those behaviors are mandatory.

A thorough understanding of the selected regression algorithms were gained and understood using a sample set of data which is closer to the real time data that we looked for.

```
reg = linear_model.LinearRegression()
#fit() method is used to train the model using the training set
reg.fit(df[['Temperature', 'pH', 'Turbidity']], df.WQI)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
reg.coef_
```

```
array([1.81129717e+00, 2.33231153e+00, 4.64678867e-04])
```

```
reg.intercept_
```

```
-16.14603442567497
```

```
reg.predict([[30.6, 7.5, 203]])
```

```
array([56.8663253])
```

Figure 26: Model predictions

Once the model is trained with 5 water quality parameters as the input variables and the results were observed using linear regression algorithm, the list of algorithms that are selected for the model training process was finalized:

- Linear Regression
- Ridge Regression
- Lasso Regression
- Elastic Net Regression
- Random Forest Regression
- K-Nearest Neighbors
- Artificial Neural Network regression

Manual Water Quality Index (WQI) calculation

Water Quality Index (WQI) is a single numeric index that is used to measure the quality of water in general. The advantage of using WQI over traditional reference methods is due to its reliable and precise output. Traditional methods used in Sri Lanka is based on a cross reference method. The parameter readings are obtained through sensors or other measurement methods and they are cross referenced with the standard SLS (614:2013) approved chart to see whether the readings are falling within an approved range.

He contradiction comes in when one parameter falls within the approved range and the other doesn't. These type of scenarios pushes the identification process towards a subjective way where people have to decide whether the water is suitable or not for usage, especially for drinking.

On the other hand, WQI can be used to precisely state the quality of the water that is being tested. All we have to do is compare the final numeric output (which is the WQI) and look for the class it belongs to. The classes generally range from very poor to very good.

WQI was developed by Horton in 1965 by selecting 10 most commonly used water quality parameters [19]. With the popularity of Horton's work, the world-wide scientists came forward with new modes of calculating WQI which were subjected to further studies and final set of approved approaches were selected.

Among the widely used approaches, 3 WQI calculation methods are very popular these days:

1. National Sanitation Foundation Water Quality Index (NSFWQI)
2. Canadian Council of Ministers of the Environment Water Quality Index (CCME WQI)
3. Oregon Water Quality Index (OWQI)

Based on the opinions of research in Sri Lanka who has prior knowledge of WQI, we were advised to use the National sanitation Foundation Water Quality Index (NSFWQI) since it has close proximity to the Sri Lankan standards.

As the concept of WQI is still up to the minute, majority of the authorities are not yet familiar with the term as well as its function. Since this research study is an initiative step to introduce the concept of WQI in Sri Lanka, we had to manually calculate WQI for the data obtained from NWSDB. The dataset was split into two. 1/3 of the dataset was subjected to manual calculation using equations where the remaining 2/3 of the data was subjected to manual calculations using online tools [20].

National sanitation Foundation Water Quality Index (NSFWQI) calculation

The usual Water Quality Index method involves a series of steps to be followed before the real WQI calculations. As the beginning step, we selected the water quality parameters that are being used as the input parameters for the WQI calculation. Upon selecting the parameters, we need to develop a common scale and weights should be assigned. The water quality data are recorded and transferred to a weighting curve chart, where a numerical value of Q_i is obtained. The mathematical expression for NSF WQI is given by:

$$WQI = \sum_{i=1}^{n} Q_i W_i \tag{11}$$

Where,

Q_i = sub-index for i^{th} water quality parameter.

W_i = weight assigned to i^{th} water quality parameter.

n = number of water quality parameters.

Once the WQI values are calculated, it is added to the dataset as a separate column corresponding to the input parameter values in order to start the model training process.

Training Machine Learning models to predict WQI

A machine learning model also known as mathematical model is simply a file that has been trained to acknowledge certain variety of patterns. As complex it sounds, the training of models involves usage of algorithms.

As stated according to the solution design, several algorithms are to be selected to train models to evaluate them and select a best mathematical model among them. As the research problem is clearly comprehended and understood, it states a regression problem where the model's intention is to predict a single numeric output (WQI) based on the multiple input parameters (water quality parameters). The models are trained with selected regression algorithms with the use of same set of data for all the models and the best model is selected based on Root Mean square Error (RMSE) value and R-Squared value which is a statistical measure of how close the data has fitted to the regression line.

For the model training and performance evaluation we have used Jupyter Notebook (anaconda 3) as the IDE which runs on Python programming language. Sklearn libraries and numpy and pandas are the libraries used for basic function and GitLab is used for version controlling.

Water quality parameters used as inputs:

- pH
- Temperature
- Turbidity
- Conductivity
- Total Dissolved Solids (TDS)

Linear Regression

Linear regression is one popular regression algorithm where the output predicted using a known set of parameters which are correlated with the output. In general, the predicted output is said to be in a linear relationship with the inputs and are continuous as well.

This specific algorithm was specially selected because of its popularity and its simplicity on training models as well as predicting the output.

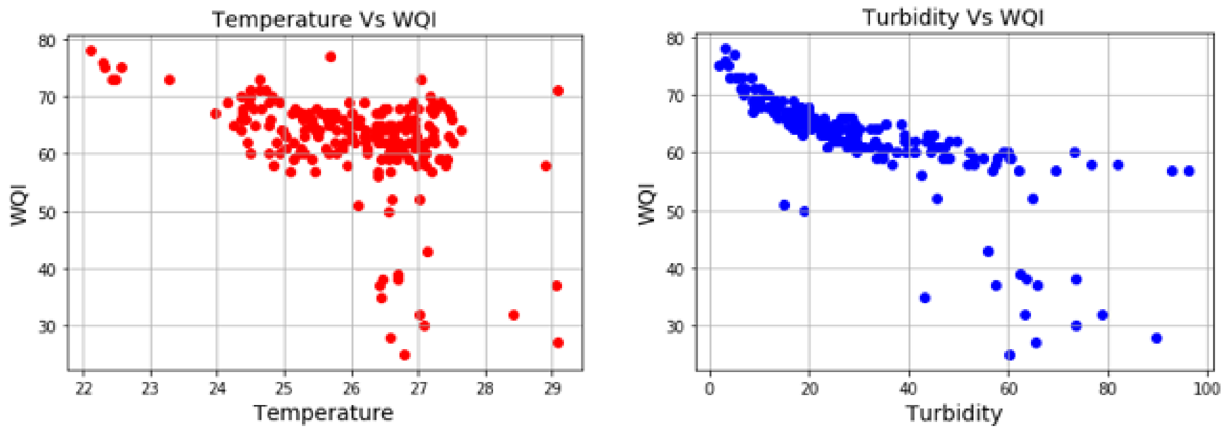


Figure 27: Graphs of temperature vs WQI and turbidity vs WQI

```
fig, ax = plt.subplots()
ax.scatter(y_test, pred_test_lr, color = "red")
ax.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=4)
ax.set_xlabel('Measured')
ax.set_ylabel('Predicted')
plt.show()
```

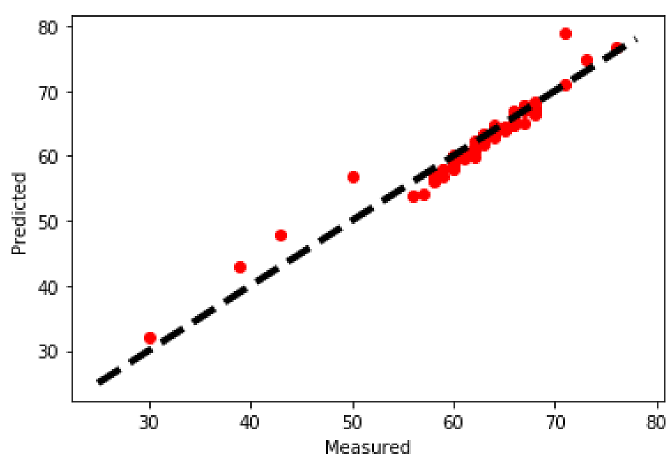


Figure 28: Predicted measurements

Ridge Regression

Ridge regression is one of the mostly used regression algorithm in a multiple regression problem. Multiple regression or most commonly known as multivariate regression is the regression that has multiple input parameters and ridge regression usually adds a penalty term to prevent overfitting.

One of the important reasons for choosing ridge regression is that it could be applied to a model that has only a small number of data as the dataset to train the model. These types of problems are very popular in a machine learning environment where a best solution must be selected using a restricted set of data.

```
rr = Ridge(alpha=0.01)
rr.fit(X_train, y_train)
pred_train_rr= rr.predict(X_train)

rmse = np.sqrt(mean_squared_error(y_train,pred_train_rr))
r2 = r2_score(y_train, pred_train_rr)
mae = mean_absolute_error(y_train, pred_train_rr)
mse = mean_squared_error(y_train, pred_train_rr)
```

Figure 29: Ridge regression

Lasso Regression

Least Absolute Shrinkage and Selection Operator which is commonly called as LASSO is a regression analysis step which is widely used due to its dual performance as variable selector and as regularization tool which is highly used to enhance the accuracy of the prediction and to enhance the interoperability of the produced model.

It is used to select the subsets of the variables.

```
pred_test_lasso= model_lasso.predict(X_test)

rmse = np.sqrt(mean_squared_error(y_test,pred_test_lasso))
r2 = r2_score(y_test, pred_test_lasso)
mae = mean_absolute_error(y_test, pred_test_lasso)
mse = mean_squared_error(y_test, pred_test_lasso)
```

Figure 30: Lasso regression

Elastic Net Regression

Elastic Net regression algorithm logically combines the functioning of ridge and lasso algorithms in order to emphasize and elevate the performance of its own self. It basically combines the linear penalties of the ridge and lasso regression. In more technical aspect Elastic Net combines the feature elimination function from lasso regression model and feature coefficient reduction function from the ridge for the improvement of its own prediction accuracies.

```
pred_test_enet= model_enet.predict(X_test)

rmse = np.sqrt(mean_squared_error(y_test,pred_test_enet))
r2 = r2_score(y_test, pred_test_enet)
mae = mean_absolute_error(y_test, pred_test_enet)
mse = mean_squared_error(y_test, pred_test_enet)
```

Figure 31: Elastic net regression

Random Forest Regression (RFR)

Random forest is one widely used and more reliable algorithm when it comes to certain machine learning problems. As popular is the decision trees are in the AI universe, the random forest algorithm basically relies itself to a group of decision trees.

There are two main types of random forest algorithm:

- Random Forest Classification algorithm/ Random Forest Classifier (RFC)
- Random Forest Regression algorithm/ Random Forest Regressor (RFR)

Even though these two varieties are applied in different scenarios; the base function can be explained simply by looking at the function of a decision tree. As the name suggests, the Random Forest is comprised of multiple decision trees which makes it a forest.

RFR could be considered as an ensemble learning technique where multiple algorithms contribute in the final prediction of the model. One of the two types of ensemble learning is Boosting, which uses the weighted average of multiple or a group of algorithms to make a stronger prediction.

As the rest of the regression models, RFR uses multiple inputs to predict the output (WQI) but the outstanding fact is that the prediction happens individually in many trees as defined by the user and the final output would be the average which is obtained from the entire trees' individual outputs.

As the preprocessing is completed, we have splatted the dataset into training and testing set in 3:1 ratio. The training set is used to train the model where the test set is basically used in the evaluation of the performance and the accuracy of the trained model.

```
#Dividing the dataset in 20:80 ratio where 20% for testing and 80% for training
#random_state - Controls the randomness of the bootstrapping of the samples used when building trees
X_train, X_test, Y_train, Y_test = train_test_split(dataset, Y, test_size=0.2, random_state = 42)

print('The Train Features: ', X_train.shape)
print('The Train Lables: ', Y_train.shape)
print('The Test Features: ', X_test.shape)
print('The Test Lables: ', Y_test.shape)
```

```
The Train Features:  (168, 4)
The Train Lables:  (168,)
The Test Features:  (43, 4)
The Test Lables:  (43,)
```

Figure 32: Random forest testing

Once the data is split, we have to create the model and define the number of trees that should be present in the forest to make final predictions.

```
#oob_score - whether to use out-of-bag samples to estimate the R^2 on unseen data.
#n_estimators - The number of trees in the forest
#random_state - Controls the randomness of the bootstrapping of the samples used when building trees

model = RandomForestRegressor(n_estimators=1000, oob_score=True, random_state=42)
model.fit(X_train, Y_train)
```

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=None, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=1000, n_jobs=None, oob_score=True,
                      random_state=42, verbose=0, warm_start=False)
```

Figure 33: Random forest predictions

Where,

- n_estimators defines the number of trees.
- random_state controls the randomness of the bootstrapping of the samples.

Once the predictions are over, we have displayed the pattern of the predicted value to observe the flow of the prediction values.

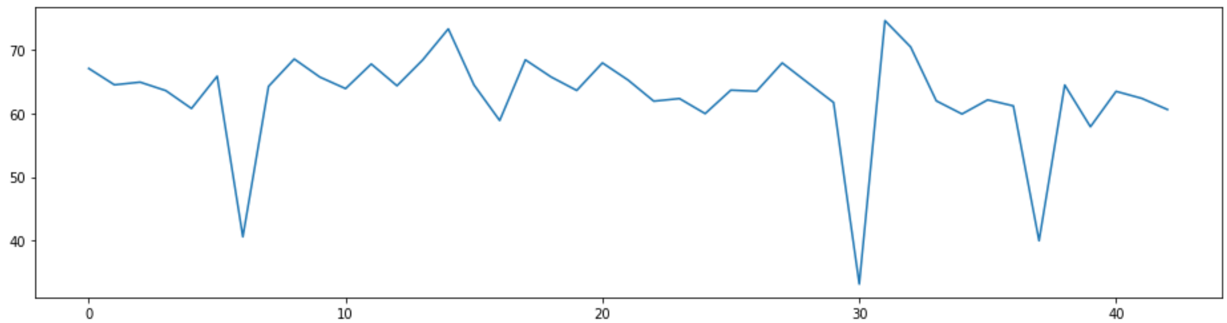


Figure 34: Pattern of predicted values

K-nearest Neighbors

This specific algorithm is used in both classification and regression problem solving where the algorithm stores all the available cases and it predicts the target based on the similarity. Although KNN is mostly suited for classification problems where pattern recognition is mostly applied, but in regression problems it can be applied where there is a possibility of a linear regression pattern where the input and output parameters are in a linear relationship.

```
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)
```

Figure 35: K-nearest implementing

Like any other model training, it is vital to split the dataset into training and test sets where one set is used for training and the other is for testing the results of the trained model to evaluate its performance and accuracy.

```
import numpy as np
from sklearn.neighbors import KNeighborsRegressor
from sklearn import metrics
knnr = KNeighborsRegressor(n_neighbors=40)
knnr.fit(X_train, Y_train)
```

```
KNeighborsRegressor(algorithm='auto', leaf_size=30, metric='minkowski',
                    metric_params=None, n_jobs=None, n_neighbors=40, p=2,
                    weights='uniform')
```

Figure 36 : Importing parameters

Artificial Neural Network

When considering regression in context with Neural Networks which is used in training this particular model, it takes the water quality parameters which are the dependent variables as the input parameters, subject them to a series of multiplication with their coefficients and runs them through a sigmoid activation function. Once the model is trained, gradient descent is performed to identify the better coefficient to fit the data till it becomes a suitable linear expression coefficient.

Once the data is split into test and training data, the model training process starts with creating suitable mathematical model.

```
from sklearn.neural_network import MLPRegressor
```

```
nn = MLPRegressor(hidden_layer_sizes=(10,10), activation='relu', max_iter=500)
nn.fit(X_train, Y_train)
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:1342: DataConversionWarning: A c
olumn-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ra
vel().
  y = column_or_1d(y, warn=True)
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:571: ConvergenceWarning: Stochas
tic Optimizer: Maximum iterations (500) reached and the optimization hasn't converged yet.
  % self.max_iter, ConvergenceWarning)
```

```
MLPRegressor(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
             beta_2=0.999, early_stopping=False, epsilon=1e-08,
             hidden_layer_sizes=(10, 10), learning_rate='constant',
             learning_rate_init=0.001, max_fun=15000, max_iter=500,
             momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
             power_t=0.5, random_state=None, shuffle=True, solver='adam',
             tol=0.0001, validation_fraction=0.1, verbose=False,
             warm_start=False)
```

Figure 37: Mathematical model

2.3.3 Forecasting Water Quality Parameters

Collecting a data is a one of the prominent work that has to be done in this component, The sense of selecting dry zone areas for gathering qualitative and quantitative data of water which need to be historical data of water quality parameters on specific water sources which are used to calculate the water quality index. This has been maintained in the Department of National Water Supply and Drainage Board Sri Lanka.

The process is started with getting thousands of data which will be the last five years data of the Northcentral region specifically in Anuradhapura and Polonnaruwa districts. Model accuracy depends on the number of data in the dataset. Every month data should be included in the training data set which is used to avoid inconsistent data. Figure 8 shows the first five row of the dataset. Gathering a data of the monsoon seasonal changes and map them into a quality parameter data.

	Year	Month	site_no	temperature	dissolved_	pH	turbidity	tds	ec
0	2014	Oct	Galenbindunuwewa	29.52299608	7.650057	5.926044	2.59059	417.4345	0.269126
1	2014	Nov	Galenbindunuwewa	28.89396872	8.933402	5.685175	1.170542	495.2804	0.209475
2	2014	Dec	Galenbindunuwewa	27.31716484	9.382118	7.080541	2.46005	549.2531	0.133334
3	2015	Jan	Galenbindunuwewa	29.15159183	9.889479	5.988921	5.548226	215.9646	0.187158
4	2015	Feb	Galenbindunuwewa	28.01887205	10.32992	6.752327	2.36568	288.2284	0.202208
5	2015	Mar	Galenbindunuwewa	27.40938592	8.607366	7.914075	3.282751	398.8879	0.174096

Figure 38: water parameter data set

Feasibility Study

It is an essential phase of every software development. As we discuss the factor and problems about drinking water with people who are in dry zone areas as well as the National water supply & Drainage Board Sri Lanka. According to their statement, we are planning to do an alternative solution as this proposed system "E-Tongue – A Smart Intelligence tool to predict safe consumption of groundwater" which has AI, ML, and IoT technologies, that are trending technologies is used to solve their problems effectively through this research. Since it is an android application following technologies and tools were used to use to develop the proposed system.

Software Boundaries

- AndroidStudio

  One of the popular official IDE for Android Operating System (OS) of Google' and that is built by JetBrain IntelliJ IDEA. It provides enhanced features of the software that is designed especially for android application to develop development.

- Pycharm

  It is a popular IDE for computer programming, which is specially used for Python programming language developing environment. It is developed and delivered by the Czech company Jetbrains. It has enhanced features for graphical debuggers, code analysis, testing, and version controller.

- Google API

  It is an application programming interface that is developed by Google that allows us to communicate with Google Services and third-party apps could be integrated with that easily. It provides the features to customize the services.

- DynamoDB

  Amazon DynamoDB is a NoSQL database that is supported by key-value and documents data structure. That delivers an output in single digits of milliseconds. The performance of the database is scaled anywhere. It is a multi-master and fully managed database.

- Amazon Web Services (AWS)

  AWS is a broadly adopted on-demand cloud computing platform and APIs for individual or business organizations. It provides all services under the pay-as-you-go policy.

- GitLab

  It is a DevOps platform for open source and end-to-end software development that builds in version control, Continuous Integration and Deployment, code review, and issue tracking. It provides self-host on our own servers or on cloud providers.

Hardware Boundaries

For backend high-performance server machine needed. To run this application an android mobile is needed with the below requirements.

- The processor speed should be 1.2 GHz or later.
- Ram should be a minimum of 2GB.
- Internal storage 1GB or more than that should be available.
- Screen resolution 1280 x 720 or higher.

Communication Boundaries

"E-Tongue" is a location-based mobile application. To fetch information from the database and get location details the application should connect to the internet. The application should be connected to Wi-Fi or mobile data to fetch data.

Memory constraints

This application is functioning by using a machine learning model to train the machine learning model more memory needed. After that optimized model will be used in the android application therefore android app does not need much memory.

For Back-end:

RAM - It should be a minimum of 8GB or more.

GPU - NVIDIA GeForce GTX 1050 or more.

Storage – 3GB or more.

For Front-end:

RAM - It should be a minimum of 2GB or more.

Storage – 1GB or more.

Operations

User able to perform below operations

- ❖ Locate their location: Users able to enter their desired water source location manually or it will be fetched current location automatically.
- ❖ View water quality parameters' level: User able to view each water quality parameter value for a given location and month year. It will forecast water quality parameters for the future. Data will be visualized on graph view.
- ❖ View precaution: User able to view the standard level of the parameter value and can get precaution when the parameter value was increased.

2.3.4 CKDu Outbreak Prediction

Model training using classification algorithms.

As the obtained dataset has a known label, supervise learning techniques were used to model training process. Predicting the risk of CKDu involved the classification of the data. Hence the dataset has trained with the number of classification algorithms. The following algorithms were used for training the model.

- Random forest Algorithm
- Polynomial Kernel SVM algorithm
- Sigmoid Kernel SVM algorithm
- Gaussian Kernel SVM algorithm

Based on the accuracy Polynomial SVM algorithm which gives the highest accuracy is used for the final model training.

Generally, Support Vector Machines is considered to be a classification approach, it but can be employed in both types of classification and regression problems. It can easily handle multiple continuous and categorical variables. SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplane in an iterative manner, which is used to minimize an error. The core idea of SVM is to find a maximum marginal hyperplane(MMH) that best divides the dataset into classes. The polynomial kernel can distinguish curved or nonlinear input space.

$$K(x, xi) = 1 + sum(x * x)^d \quad (12)$$

Where d is the degree of the polynomial.

Create RESTful web API to access database values and predicted results.

To predict the risk of CKDu the sensor data needs to be feed to the machine learning model. In order to retrieve the sensor data from the AWS dynamo DB instance the RESTful web API is developed using Node JS as a programming language.

In order to get the predicted results of the machine learning model and to feed the data to the model API is created using Flask web framework for python. Both of the API services are accessible through mobile application using the Retrofit client.

2.4 Wireframes and designs

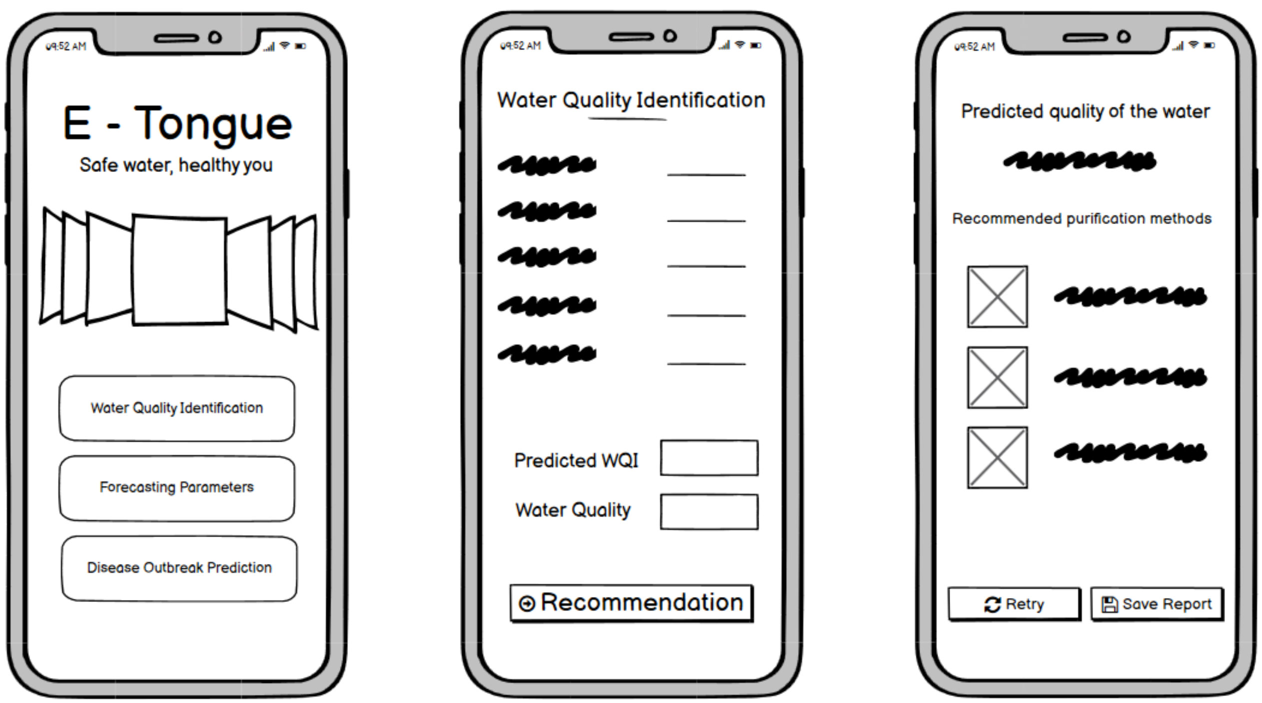


Figure 39: wireframes of the smart application

Figure 40: Forecasting water parameters wireframes

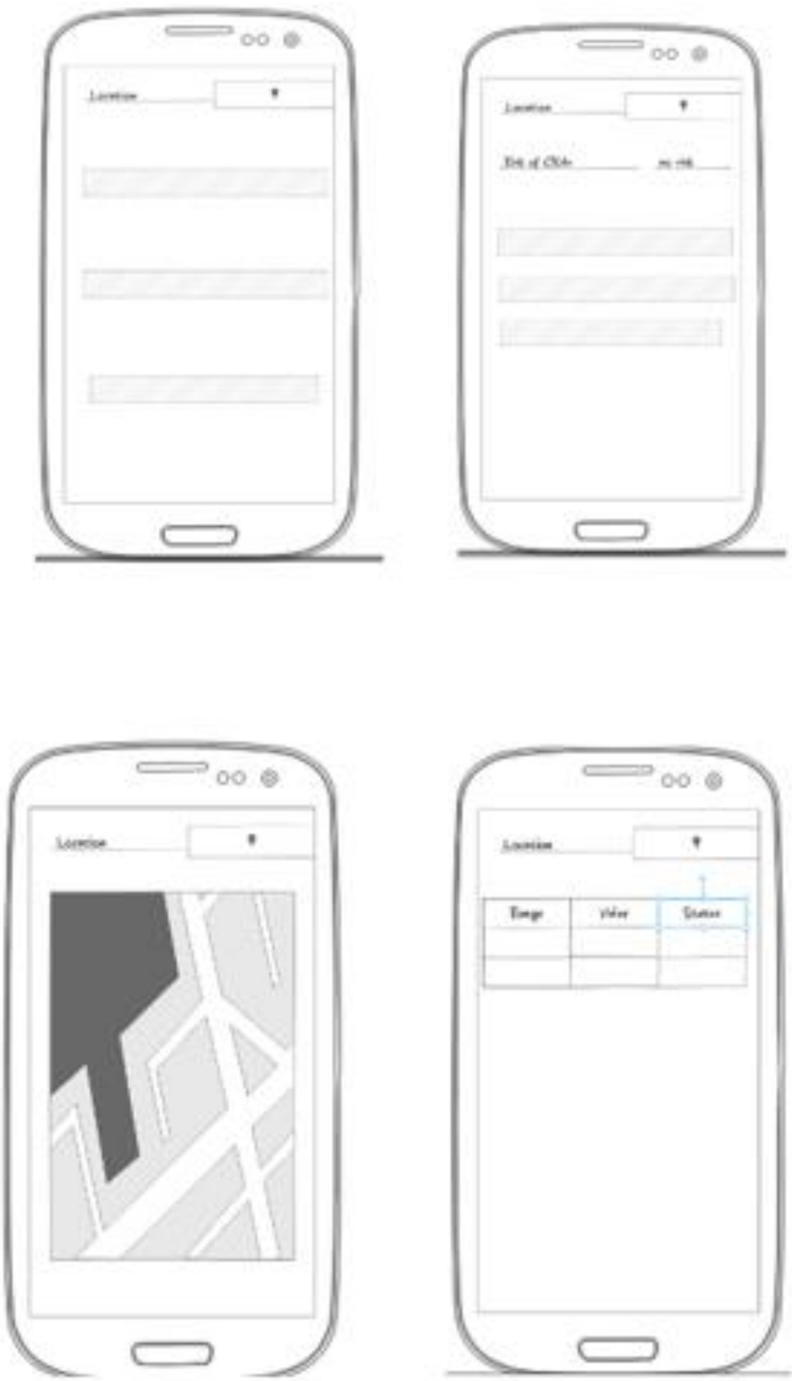


Figure 41: Location wise wireframes

2.6 Testing and Implementation

2.6.1 Implementation

E-Tongue is a mixture of both hardware and software components. The Hardware component consists of an Arduino NANO and Nodemcu ESP8266 which connects to the pH, conductivity, temperature and turbidity sensors whereas the software component consists of a mobile application which acts as a mode of communication and a backend node.js server that operates in accessing the database. The server that runs the model is based on Python and the API has been written using PyCharm.

At the development stage of the system, many strategies and awareness steps were followed to guarantee the secureness of the application user data. Coding standards and best practices were followed to ensure the quality of the code and a proper security has been focused in the AWS database where the sensor data were stored.

Front End

Technologies used: Android and Java

Platform used: Android Studio

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development [21]. When using this platform for front end development, we need to create separate activities to define specific functionality.

As an initial step, an activity was created to display the readings obtained from the device. Once an activity has been created, the platform will generate a layout file which is linked to the created activity. The layout is defined using an Extensible Markup Language (XML) file where we can work on the design of the user interface.

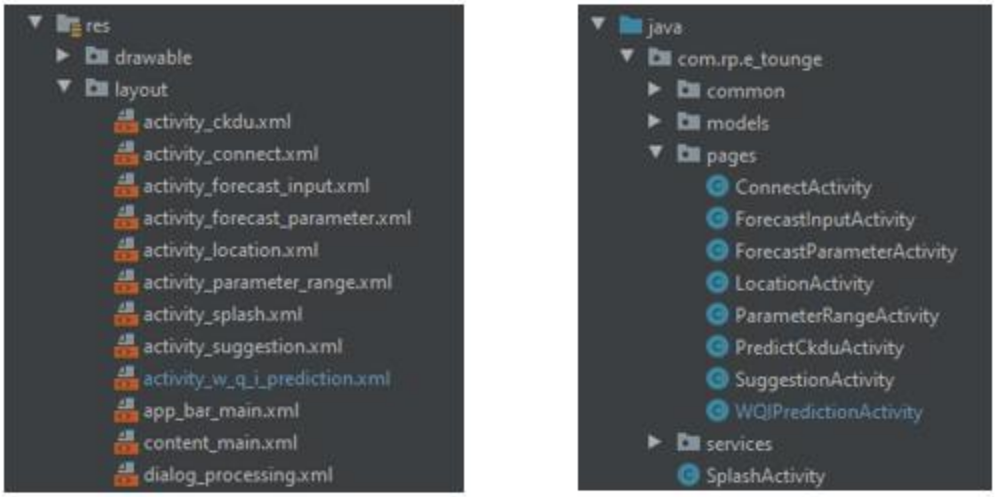


Figure 42: Android studio classes and activities

When implementing the backend server to connect with the front end, a method is called where the server API is called to obtain data and predicted result to be displayed in the user interface.

```java
public interface ApiServices {

    @GET("/resultParameterForMonth")
    Call<List<PredictWaterParameters>> getJson(@Query("Year") String year, @Query("Site") String site,
                                               @Query("month") String month);

    @GET("/resultWQI")
    Call<List<WQI>> getJsonWQI(@Query("Temperature") float Temperature, @Query("pH") float pH,
                               @Query("Turbidity") float Turbidity, @Query("Conductivity") float Conductivity,
                               @Query("TDS") float TDS);


    @GET("/resultCKD")
    Call<CkdResult> getCkduResult(@Query("site") String site,
                                  @Query("temperature") float temperature,
                                  @Query("ph") float ph,
                                  @Query("turbidity") float turbidity,
                                  @Query("tds") float tds,
                                  @Query("ec") float conductivity);


    @GET("/parameters/getParameters")
    Call<SensorDataResponse> getSensorData();
}
```

Figure 43: Api services

Back End

Technologies used: Python 3.8.5, NodeJS

Platform used: PyCharm, Visual Studio Code

The backend of the system comprises of 2 different components. The results from these 2 backend servers must be directed to the front end for the users to view the data.

As the device reads parameter readings from the sample, it is designed to store the reading values directly to the cloud where the AWS DynamoDB is hosted. The device is designed in a way where it sends 5 rows of data from five timestamps with equal intervals. As this particular technique ensures the quality of the data that we are obtaining, we have created a backend server using NodeJS to obtain the average of the stored 5 rows and to direct them to the python model which was hosted in another server.

```
AWS.config.update(awsConfig);
let docClient = new AWS.DynamoDB.DocumentClient();

var ParameterController = function(){
    this.getData =()=>{
        var params = {
            TableName : 'ESP8266TEST',
            IndexName : 'device-index',
            KeyConditionExpression : 'device = :deviceVal',
            ExpressionAttributeValues : {
            ':deviceVal' : 4468785
            }
        };
        docClient.query(params, function(err, data) {
            if (err) {
                console.error("Unable to read item. Error JSON:", JSON.stringify(err,
                        null, 2));
            } else {
                console.log("GetItem succeeded:", JSON.stringify(data, null, 2));
            }
        });
    }
}
```

Figure 44: Database data fetching api call

The second server where the machine learning random forest regression model is hosted will have to obtain the data from the database in order to predict the WQI and send the results to the mobile application for the results to be displayed.

```
private void getWQIPrediction() {
    retrofitClient = ApiClient.getClient();
    apiServices = retrofitClient.create(ApiServices.class);
    Utils.showProgressDialog(context);
    Call<List<WQI>> call = apiServices.getJsonWQI(temperature, ph, turbidity, conductivity, tds);
    call.enqueue(new Callback<List<WQI>>() {
        @Override
        public void onResponse(Call<List<WQI>> call, Response<List<WQI>> response) {
            Utils.hideProgressDialog();
            Log.d( tag: "Direction",  msg: "Response Request " + response.raw().request());
            Log.d( tag: "Body",  msg: "Response body " + response.body());
            if (!response.isSuccessful()) {
                Utils.showMessageDialog( message: "Something Went Wrong,Please try again later", context);
                Log.d( tag: "Code",  msg: "Response code : " + response.code());
                //textView.setText("Code : "+response.code());
                return;
            }

            List<WQI> results = response.body();
            wqi = results.get(0).wqi;
            Wqitextview.setText(Double.toString(wqi));
            setData();
        }
```

Figure 45: WQI prediction

E-tongue tool was implemented as the solution for identification of good quality water. Basically, the following items were used to build the structure.

- Three feet water pipe
- Sockets
- Stop value
- Wooden blocks

Figure 46: Prototype design

we have made a portable device which can carried out at any place to capture water samples for real-time predictions. To capture sensor readings, we have built three cylinders. From that two of them are utilized to gather sensor data, and the remaining one as water inlet. All the sensor probes are placed on the lids of the cylinders.

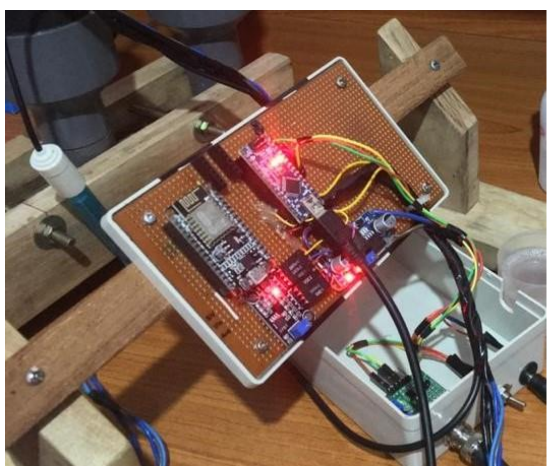


Figure 47: Embedded Circuits

In Figure 28, it shows how the circuit was implement foe ‘E-tongue’ tool. The followings are the components used.

- AC Adapter
- Buck Convertor

- Nodemcu ESP8266
- Arduino Nano
- Logic level Convertor
- Ph Module
- Turbidity Module

With the use of above components, we were able to construct this intelligent tool. 12v current supply is given to operate this unit. Since 12v is high to operate these broad we have safely reduced the voltage up to 7v with the help of buck convertor. This is used to protect the broads. Nodemcu operates with 3.3v while nano works with 5v. For bridging esp8266 and Arduino nano we utilized logic level convertor to act as receiver and transmitter. In this manner we were able to accomplish hardware sector.

Backend created using node language to fetch real-time for predictions. It takes values from the sensors and send it to the database as well as for the device. We keep log of readings for seasonal forecasting.

2.6.2 Testing

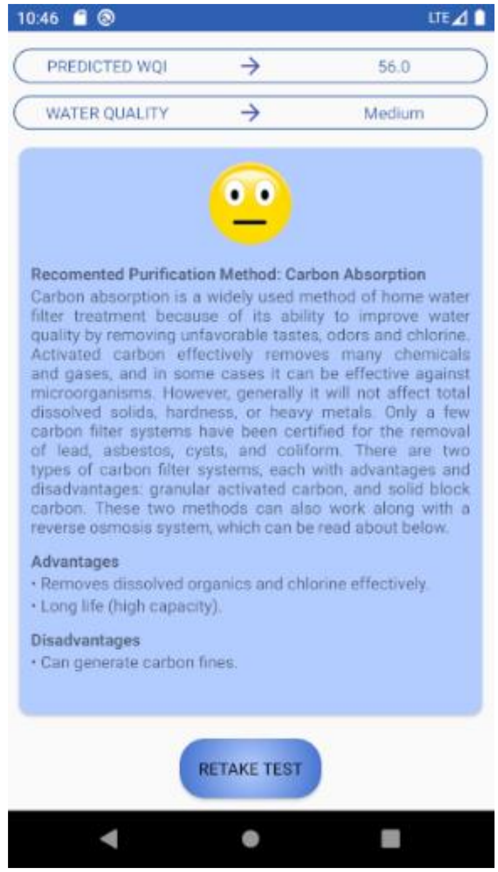
A. WQI prediction - Functionality Test

Test Case 01: Predicting the WQI of a ground water sample collected from a well in Kegalle.
Pre-condition: Model trained with data obtained from NWSDB
Input parameters:

- Temperature: 25.69
- pH: 6.76
- Turbidity: 0.02
- Conductivity: 0.09
- TDS: 65

Expected output: WQI in the range of Bad to Medium, parameters that exceeds the range must be highlighted.

Actual output:



Comments: Correct values are retrieved from the database, correct class of the predicted WQI is displayed and suggestions are relevant to the prediction.

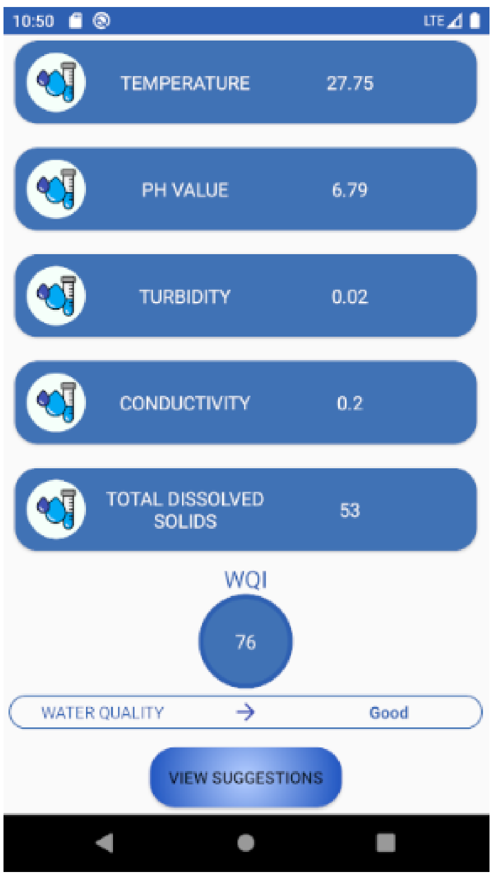
Status: **Pass**

Test Case 02: Predicting the WQI of a tap water sample collected from the university premises.
Pre-condition: Model trained with data obtained from NWSDB
Input parameters:

- Temperature: 27.75
- pH: 6.79
- Turbidity:0.02
- Conductivity: 0.08
- TDS: 53

Expected output: WQI in the range of Medium to Good, parameters that exceeds the range must be highlighted.

Actual output:



Comments: Correct values are retrieved from the database, correct class of the predicted WQI is displayed and suggestions are relevant to the prediction.

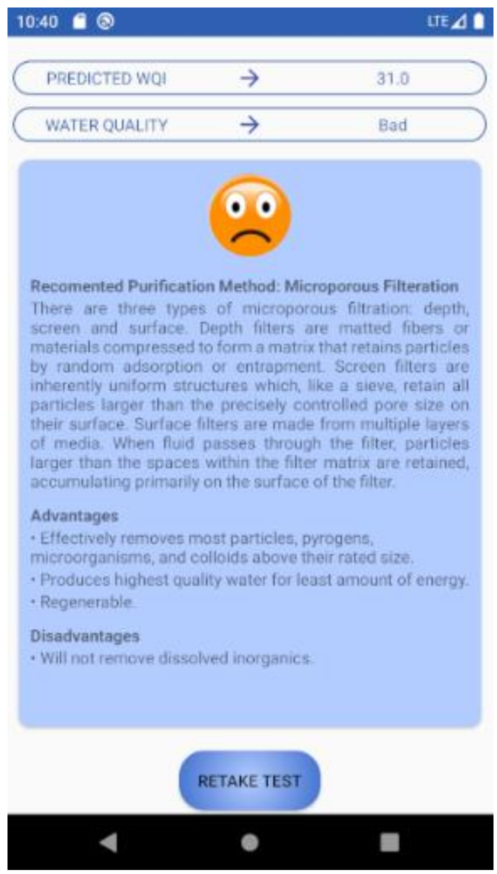
Status: Pass

Test Case 03: Predicting the WQI of a surface water sample collected from a lake after rain.
Pre-condition: Model trained with data obtained from NWSDB
Input parameters:

- Temperature: 27.00
- pH: 5.99
- Turbidity: 3.77
- Conductivity: 0.31
- TDS: 217

Expected output: WQI in the range of Bad to Very Bad, parameters that exceeds the range must be highlighted.

Actual output:



Comments: Correct values are retrieved from the database, correct class of the predicted WQI is displayed and suggestions are relevant to the prediction.

Status: Pass

B. CKDu Outbreak prediction

The testing procedure of the above elaborated implementation is explained below.

Front-end testing

Test case ID	001
Test case scenario	Check to fetch user current location
Test steps	a. User navigate to the home page b. Tab locate me button c. Display the current user location
Test data	Location
Expected result	User location should be shown in the home screen
Actual result	As expected
Pass/Fail	Pass

Test case ID	002
Test case scenario	Search user entered location
Test steps	a. The user navigates to the forecast page b. Enter the user's desired location c. Select the water resource area d. Submit
Test data	Location = Galle, Sri Lanka
Expected result	The location should be shown on the home screen
Actual result	As expected
Pass/Fail	Pass

Test case ID	003
Test case scenario	Show warning alert on water quality parameter values based on range
Test steps	a. The user navigates to the home page b. User select the user location c. The user navigates to the predict CKDu page
Test data	Temperature =27.36 Ph =6.94 Turbidity =14.15 TDS = 162.80 EC =0.2
Expected result	The color of the card should be change accroding to the range of the water quality parameter
Actual result	As expected
Pass/Fail	Pass

C. Forecasting parameters – front end testing

Test case ID	001
Test case scenario	Check to fetch user current location
Test steps	d. User navigate to the home page e. Tab locate me button f. Display the current user location
Test data	Location
Expected result	User location should be shown in the home screen
Actual result	As expected
Pass/Fail	Pass

Test case ID	002
Test case scenario	Search user desired location
Test steps	e. The user navigates to the forecast page f. Enter the user's desired location g. Select the water resource area h. Submit
Test data	Location = Nochchiyagama, Sri Lanka
Expected result	The location should be shown on the home screen
Actual result	As expected
Pass/Fail	Pass

Test case ID	003
Test case scenario	Check calendar
Test steps	a. The user navigates to the forecast page b. Hit the calendar button c. Select month and year d. Submit
Test data	Month = December , Year = 2022
Expected result	Display year and month
Actual result	As expected
Pass/Fail	Pass

Test case ID	004
Test case scenario	Check the submit button disabled when the calendar field empty.
Test steps	a. The user navigates to the forecast screen b. User enter location or site c. Keep the calendar field empty
Test data	Location = Nochchiyagama, Sri Lanka
Expected result	The alert dialog should be shown as “Input expected year and month”
Actual result	As expected
Pass/Fail	Pass

3. RESULT & DISCUSSIONS

3.1 Results

3.1.1 Device

The research is focused on implementing a water quality measuring device with the smart application ensuring real time data. The goals were to predict water quality index, outbreak of water borne diseases and forecasting water quality parameters seasonal wise. In this section it provides the results of the experiments performed to solve the main research question.

E-Tongue device shows the ph level, temperature, turbidity, and total dissolved solids of the tested water samples from groundwater resources. The following table 8 shows some samples gathered from different water sources.

Table 6 Water sample dataset

Electric Conductivity (mS/cm	ph	TDS (ppm)	Temperature (C)	Turbidity (ntu)
0.23	7.16	161	25.44	0.02
0.14	5.07	97	25.56	197.51
0.23	6.95	159	25.25	0.02
7.98	5.58	5586	26.69	0.02
0.31	5.83	218	27.06	3.77
0.24	5.84	171	27.19	3.77
0.24	6.82	169	25.69	0.02
0.08	6.99	58	28.69	0.02
0.08	6.79	53	28.19	0.02

3.1.2 WQI prediction

A. Algorithm Training

As explained in the methodology above, the initial objective was to select a suitable model to proceed with the system implementation. As a part of the study, we have come up with a solution to select the best way to choose one single algorithm is to conduct a performance analysis. As this analysis will result with one algorithm which is suitable to this scenario.

This particular research problem requires regression algorithms as we are predicting a single continuous numerical value, and this could be achieved by the above selected regression algorithm training.

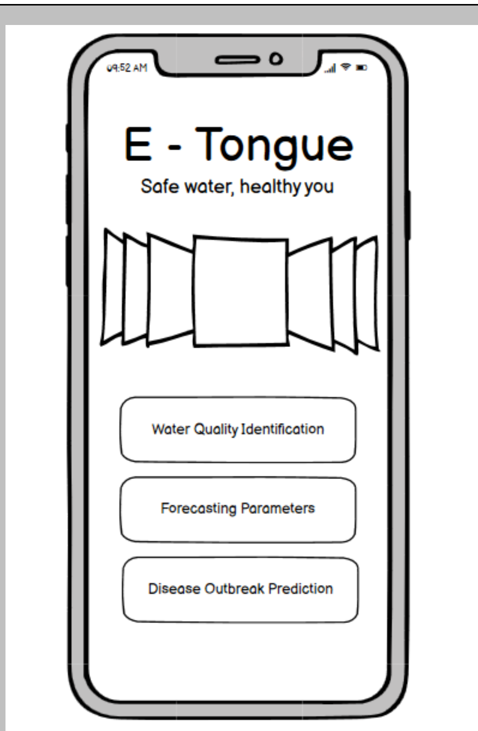
The selection of a best algorithm must be done in an impartial process. We have used the same dataset which we have obtained from the National Water Supply and Drainage Board, Sri Lanka. The platform that we have trained, and the hardware components of the trained computer are maintained consistent for all the models to ensure the quality of the selected final model by keeping a consistent environment.

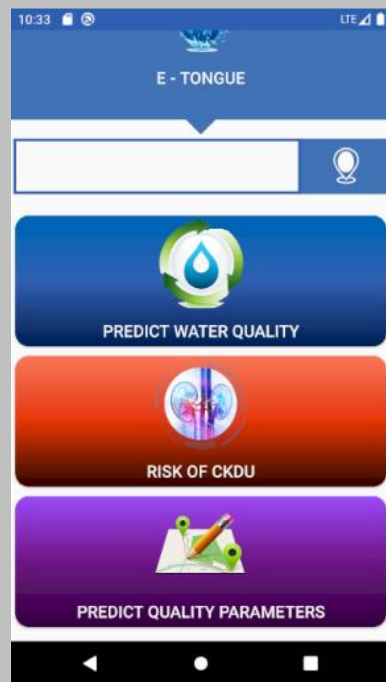
Table 7: Algorithm accuracy chart

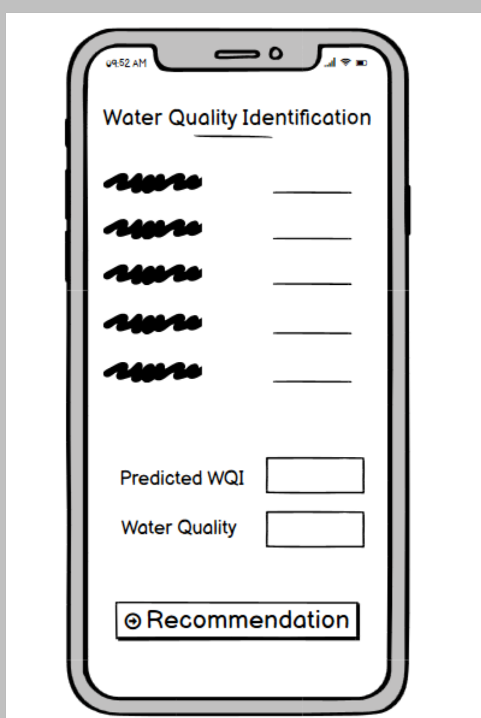
Algorithm Used	***MAE***	***MSE***	***RMSE***	***R-SQUARED***
Linear Regression	1.3082	3.6730	1.9165	0.9249
k Nearest Neighbors	1.9203	17.2976	4.1590	0.6424
Random Forest Regression	0.6673	1.6477	1.2836	0.9782
Artificial Neural Network	7.5060	90.2375	9.4993	0.2381
Ridge Regression	1.3084	3.6738	1.9167	0.9249
Lasso Regression	1.3133	3.6931	1.9217	0.9245
Elastic Net Regression	1.3215	3.7473	1.9358	0.9234

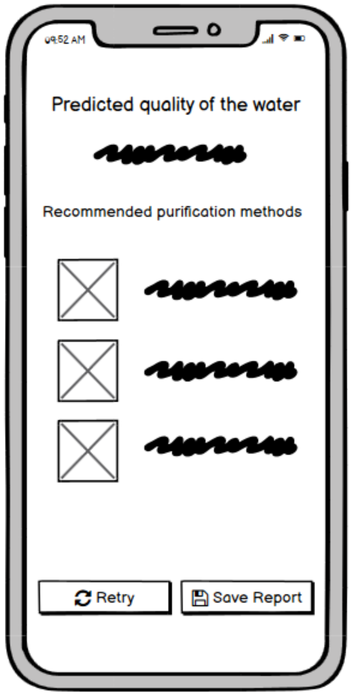
In a regression problem, the R-Squared value defines the coefficient of determination. Based on the observed results, the Random Forest Regression model has highest R^2 and lowest Root Mean Square Error. As a result of the model selection process, RFR was selected as the model to be implemented into the system to finally predict the WQI.

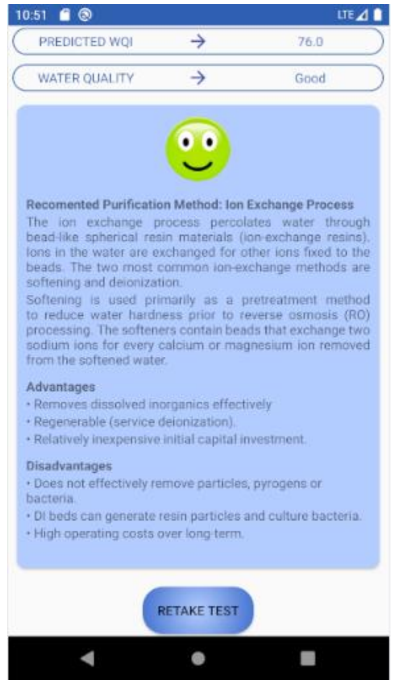
B. Implementing the UIs











The UIs were implemented as they were designed during the design phase of the project. The above table shows the comparison of the designed UIs with the planned wireframes

3.1.3 Forecasting parameters

Results of each model

- **VAR**

The VAR algorithm, which has the capability to predict the time series data. VAR model is trained and tested with the dataset it gives less accuracy because of the less amount of data set. Each place contains 53 months of data. Therefore, it gives the 69.47% accuracy of the model.

- **RFR**

RFR is a provides the best accuracy even the dataset consists less amount of raw data. The accuracy of the model is not going to depend on the number of samples. The data set consists of 53 months of water quality parameter for each site. Therefore, it gives the 83.24% accuracy of the model.

- **LSTM**

It is a time series forecasting algorithm using in RNN. That obtains a huge amount of data for training to provide the most accurate output. The accuracy of the model is depending on the number of samples. Eventually, combined all different locations data together for training and testing.

- **SVR**

  It is a counterpart of the Support Vector Machines (SVM). That admits the presence of no relation of input variables that are provided in the data set. Multiple outputs that also does not have any relation. Therefore, the accuracy of the model doesn't depend on the input and output feature. Therefore, this model gives 75.27% accuracy.

- **KNN**

  KNN model is a bit easy to implement compared to the model. On the other hand, it will be a nonlinear model. The tends of parameter fitting to be quick. Therefore, it takes less computational power than the other model. Multi variant input and output don't affect the accuracy of the model. This model provides 72.92% accuracy.

  As a result of the model was figured in below figure 28 shows the model evaluation which was tested against the test data set

Eventually, RFR is selected for the finalized model because of these given reasons. It produces high accuracy even the dataset consists of a few samples, a very effective way to handle the multiple inputs and output variables, it has a reliable method to estimate the missing data and it maintains the accuracy when a big proportion of the data missing.

Model Deployment

Deployment is a method that integrates the ML model into a production environment or centralized server to make more practical decisions based on the given data. This is the final stage of the machine learning life cycle. Trained the machine learning model on a local PC environment. RFR model that has enough capability to predict effectively and efficiently. The model file is generated once the training was completed. That will be stored as a pickle file which is a serialized format to store the objects. ('model.pkl').

Flask is an efficient python server that is using microservice that allows us to build RESTful APIs that need to communicate between the back end and front end via HTTP protocols with minimum configuration. Figure 48 shows the sample output of the GET method of this component. Google map API is used to fetch the user location and the user able to search the location via that service.

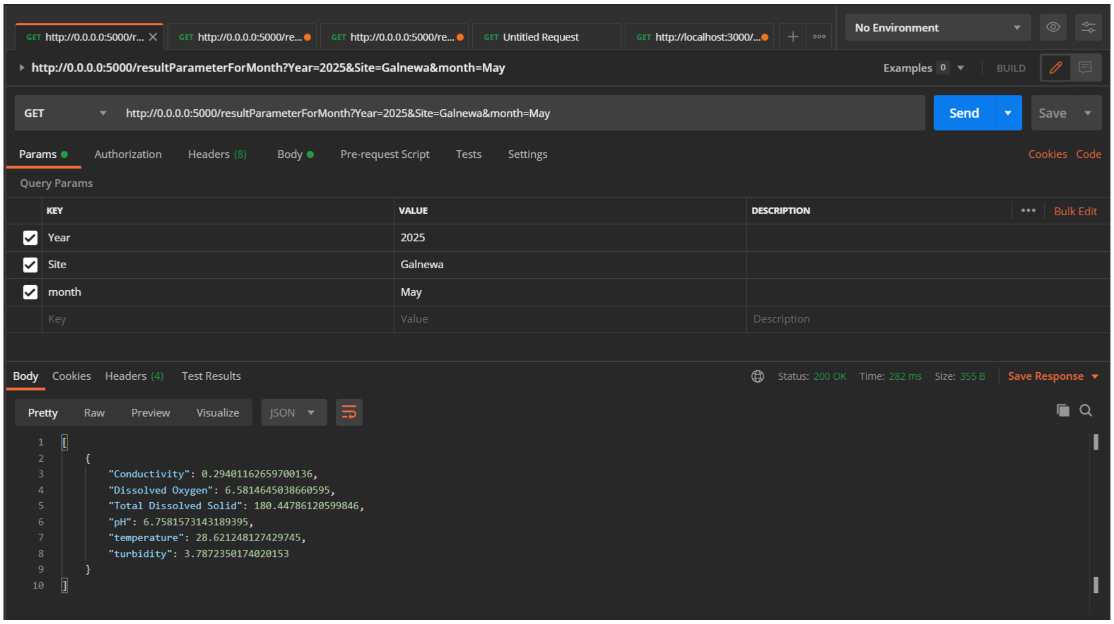


Figure 48: postmen get method

User Interface of Mobile Application

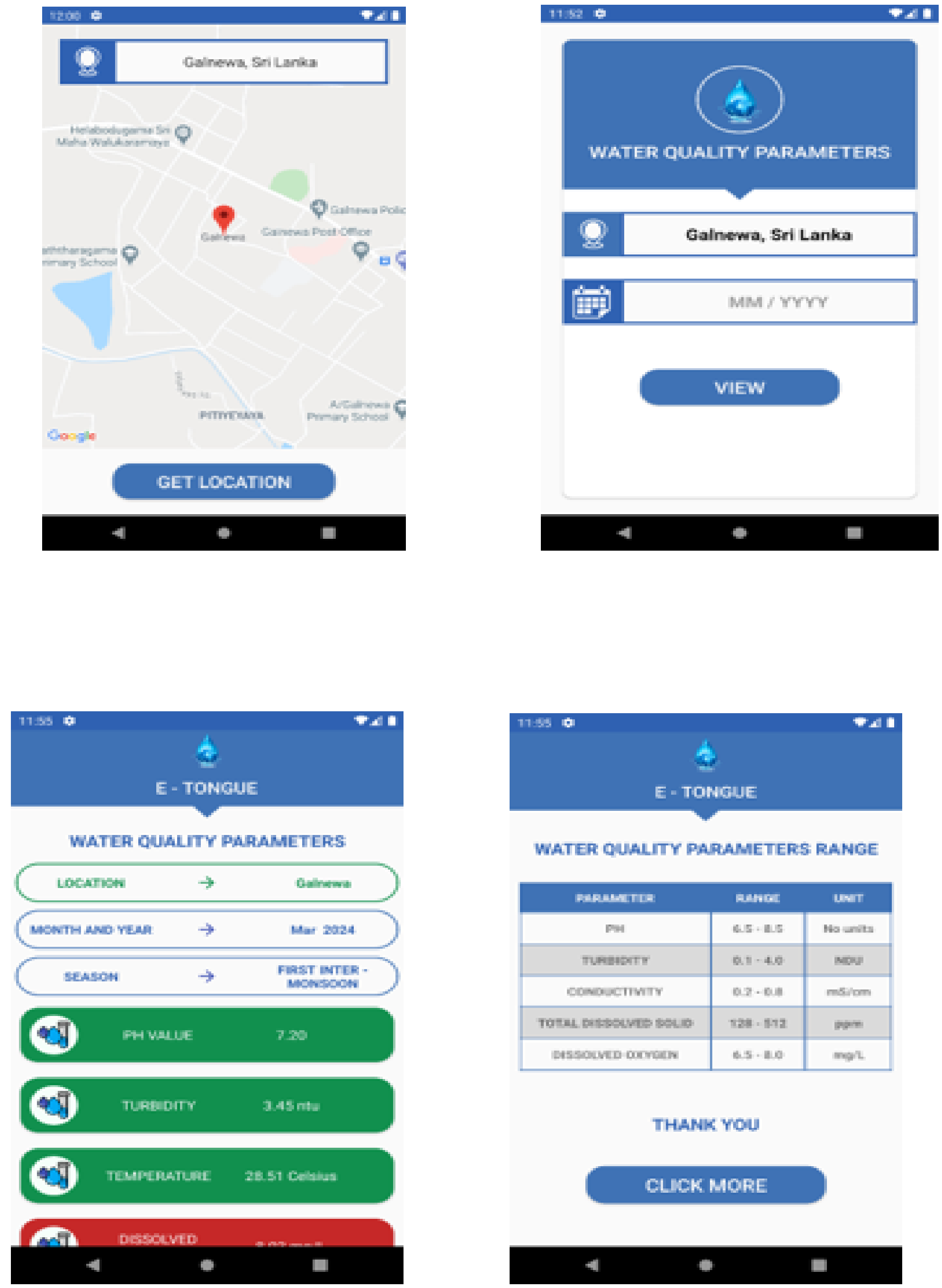


Figure 49: mobile user interfaces

3.1.4 CKDu Outbreak Prediction

Results of each model

Table 8: ckdu algorithm accuracy rates

Model	*Avg accuracy*	*Avg Error*
Polynomial Kernel SVM	0.7699	0.4201
Random Forest Regression	0.4969	0.5031
Gaussian Kernel SVM	0.5683	0.4317
Sigmoid Kernel SVM	0.5432	0.4568

User interface of mobile application

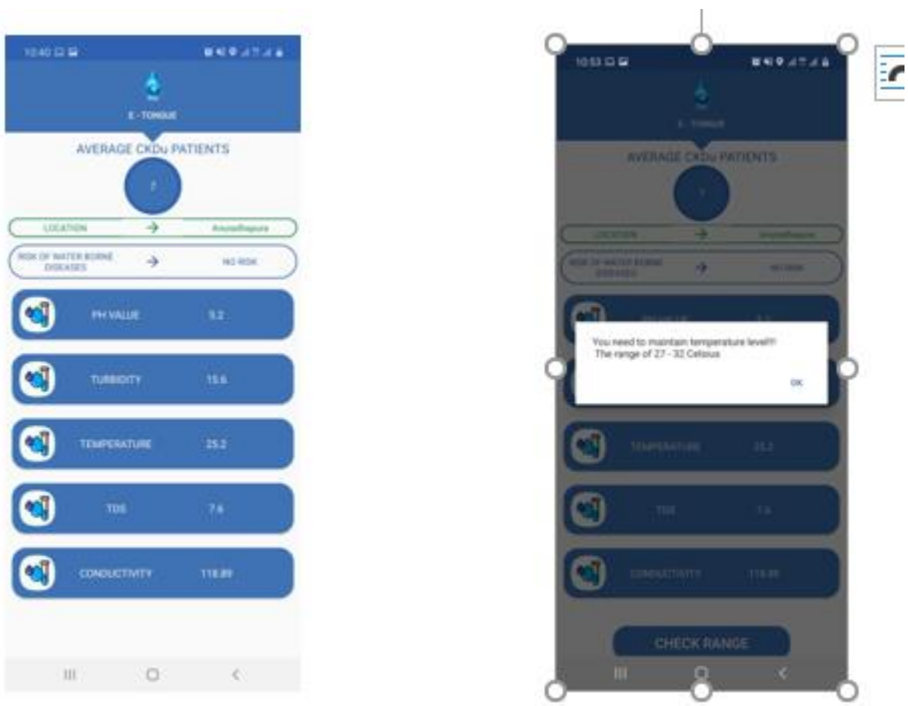


Figure 50: ckdu mobile interfaces

3.2 Research Findings

When revisiting the literature survey at the beginning of the document, it was stated obvious that machine learning approaches are way better and efficient when applying to the solutions that most of the current problems leads to, and solving issues regarding water quality management is no exception.

3.2.1 Device Implementation

Selecting the best combinations of sensors.

The key component of the device was the sensors. Analyzing literature surveys and earlier findings we were able to out the most suitable sensors which is need for predictions. They were ph probe, turbidity, temperature, and electric conductivity and by using conductivity there was a possibility of calculating total dissolved solids. Since conductivity sensors are much expensive. We had to build the conductivity sensor by utilizing basic concepts of science.

Among these parameters conductivity and turbidity depend on environment temperature. We took this also into consideration to give an accurate result by calibrating the sensors accordingly.

3.2.2 WQI Prediction

By considering the results from the conducted literature survey, the focus of this research study was to propose, design and implement a solution that is faced by the currently available systems. By looking at all the involving factors and limitations, the solution proposed should be an asset to the public where the ultimate focus is flown sent towards them.

As the traditional models which are being practiced specially in our country is more manual and time consuming, the aim is to fashion a device with the inclusion of mathematical models so that the precise results could be driven directly on time. That being said, the scope of this study to introduce the concept of WQI to the currently working bater bodies where it could be used to calculate the quality of the water samples more accurately without any contradiction in the results. The most challenging concept starts at the very beginning when selecting some water quality parameters to be used in the machine learning model. With the presence of 30 different water quality parameters and 15 different parameters among them which are being used by the industrial professionals, 5 main water quality parameters such as pH, turbidity, temperature, electrical conductivity and total dissolved solids are selected for our study based on the correlation of the parameters and the importance they carry.

Preparing a suitable dataset for the machine learning approach was something which needed extra care. As discussed earlier, the concept of this single numerical unit, WQI was a new branding and it needed a lot of background work and understanding to pick a suitable one among the available ones which reflects the nature of water that are present in the country and for the purpose of introducing the Water Quality Index concept to the industry, we had to manually calculate WQI of each parameter set to deliberately show how difficult it is as it involves a lot of complex mathematical procedures and the usage of graphs. These manually calculated WQIs were added correspondingly to their respective parameter set to prepare the final data.

In the end, after a thorough study and much deliberation, seven regression machine learning models such as linear, ridge, lesso, elastic net, random forest, kNearest neighbors and artificial neural network regression models were handpicked for completing the initial objective. All of the above-mentioned models were trained individually using the same dataset which was obtained after a hectic search in the industry, to select the best model out of it based on their results. Individual model's test data splits were evaluated and compared, and Random Forest Regression model was selected to carry on this study because of its Highest R-Squared value and its lowest root mean square error value. As this process was a success, the selected RFR model was put into test and the number of trees were adjusted till an acceptable accuracy percentage was obtained.

3.2.3 Forecasting Parameters

Analyzing the technology area

According to the literature survey, our final system should be the mobile application and figure out the specific features, In order to develop the prediction model, need to use the real dataset that consists water quality parameter for a certain location. Therefore, historical data of the water quality parameter level for each site has to be combined to develop the model that is going to predict the parameter values for the future.

Model training is a prominent basic procedure to develop an ML model in which performance depends on the learning outcome of the dataset. When trained the model, have to complete each phase of the ML model development cycle. The incomplete phrase is not going to give the best model to the completed phase model. Data need to be preprocessed before starting the model training stage. Mean normalization and feature scaling are done at the preprocessing phase.

Choosing the best algorithm

The developed system able to forecast the water quality parameters for a given location. It is fully managed by the ML techniques. Before achieving the main goal of the system, need to satisfy other mandatory requirements, In order to find the best approach, training the ML model with most appropriate algorithm and optimize the accuracy level of it, all the pre-trained model should be examined with test data set then obtaining the output.

CKDu Outbreak Prediction

Main objective of this research study is to create smart tool to predict safe consumption of ground water. Along with the research project objective, main objective of this individual component is to alert user if there is a possible risk of exposing to CKDu.

Main approach for the component's solution is supervised learning techniques. Training model is the major functionality of it and the whole result of the prediction is depends on the algorithm that is used for the model training. Therefore, selecting best suited model plays the significant importance. To achieve that dataset needs to be trained with multiple algorithms that are suitable for the given scenario and select the best algorithm based on the accuracy of each model. The likelihood of increasing the accuracy of the prediction can be affected by the size of the dataset.

It is plausible that CKDu is multifactorial. Agricultural practices, geographical area distribution and number of other factors are suspected to be the cause of it. Based on literature survey and finding the people who uses shallow wells in close proximity to irrigation systems for agriculture are more affected by the CKDu comparing to the other areas. Hence, we can assume that values of water quality parameters or quality of the water can be one of the causes for the CKDu. Therefore, this research study gives the prediction of possible risk of exposing to the CKDu by analyzing the water quality parameters, location and the past data of the CKDu patients.

3.3 Discussion

The purpose of this research study is to implement a smart tool that could be used in real time to predict the safe consumption of ground water. As this topic was further modified and 4 individual components were identified based on the need of the present society and the extent of how the thought process works. Setting aside the device design and implementation, 3 main components were proposed which uses machine learning techniques. Out of the three further classified sub objective, one of them, this component focuses solely on introducing the concept of Water Quality Index to the authorities and predicting the WQI using machine learning methods to identify the quality of ground water.

As discussed above in the literature survey that was done, in Sri Lanka, the concept of WQI is non familiar and yet to be introduced and people had a little knowledge of how well this index could be used to precisely identify the quality of the water from any source when it is subjected to an analysis. Although this concept of WQI is familiar in other countries, many researchers have implemented systems to incorporate machine learning into the analysis of water using the index and have successfully been able to propose results. However, the results were not implemented to a system where the commercial aspect was neglected.

3.4 Summary of the student contribution

Member	***Component***	***Tasks***
AMPB Alahakoon	Device implementation and Sensor calibrations	• Feasibility study for the requirement needed. • Identify the most suitable sensors which will be used to measure water quality by training models. • Developing a smart IoT embedded device for measure quality of water. • Constructing new sensor to measure conductivity levels in water. • Implementing data transmission media to transfer data between two broads and between database and smart application. • Calibrating sensors to give away readings with minimum error fluctuation. • Improving accuracy of sensors by considering environment factors. • Implement an inception environment for the research team to test water samples.
Nibraz MM	Water quality identification by predicting the WQI	• Thorough feasibility study to gain knowledge on the study area. • Identifying dataset to be incorporated with the model training process. • Data preprocessing, standardization and normalization making it flexible to be fed into the model.

		• Identifying suitable regression algorithm to proceed with the implementation. • Testing and evaluating the performance of the trained models to choose a best amongst them. • Elevating the performance and accuracy of the selected model. • Deploying the trained model into RESTful web service. • Developing Mobile application to connect the user with the service. • Displaying the prediction and readings in the mobile application to be accessed by the user.
Thenuja S	Forecasting water quality parameter	• Feasibility study to perceive the requirement of the aspect. • Identifying the dataset which will be used in machine learning model. • Data preprocessing to reduce the redundant data and increase the accuracy rate. • Identifying the most suitable algorithm in order to produce accurate results. • Testing the developed machine learning model with a dataset that has known results in order to obtain the accuracy. • Deploying a model into REST full web service. • Connecting with Google Map API to fetch user location.

		• Alerting the user via mobile app if the parameter range is different in future.
Gunarathna PMSSB	Predicting possible risk of exposing to CKDu	• Feasibility study to perceive the requirement of the aspect. • Identifying the dataset which will be used in machine learning model. • Performing a thorough study of CKDu and identify the parameters that can be used be used for the solution • Identifying the most suitable algorithm in order to produce accurate results. • Testing the developed machine learning model with a dataset that has known results in order to obtain the accuracy.

4.CONCLUSION

The E-Tongue intelligent system targets water sources that needs an identification of their quality for drinking and domestic usage. It is designed for the people who reach water bodies especially ground water sources to fulfill their day to day needs. To reduce the consequences such as water borne diseases that people can expose themselves to by consuming unsafe water, and to monitor the water quality levels by predicting them way ahead of time, this system can be a positive asset to people as well as the authorities.

We have designed this system to address some major problems that are faced when treating water and identifying the quality. This procedure when done manually takes a minimum of 4 to 5 days in average to generate a final report to be given and in Sri Lanka the process doesn't necessarily gives a conclusion as the quality of water belongs to a certain category.

Considering the component which predicts the WQI, it uses a mathematical regression model to do the prediction. Once a sample from any required source is fed into the device, the parameter readings, the water quality, and industrial suggestion method for purifying the water from the selected source is displayed to the user.

When developing the mathematical model, we faced few hardships through the journey. As the accuracies of the prediction plays a vital role in overall performance of the model, it is important to evaluate the performance of the model that is used in the final system. Although the final model for this specific component was chosen on an experimental procedure, there are few complex models that are present in the industry which could be incorporated into the study. Some complex models like Convolutional Neural Network (CNN) for regression and other deep learning models which can give more advanced and more accurate results can be added into the system as a future work.

Although this scenario is a regression type, some classification models too can be used to classify the water sample into five classes to which the water sample belongs to. Same way as this study, a set of classification algorithms can be selected and trained in the same environment with the same dataset to select the best algorithm and the implementation can be proceeded. As our sub objective included introducing the concept of WQI to the authorities, we had to drop the idea of using classification algorithms as it will not give an index output.

For further development, adding more sensors to the system which can include more parameter readings can further improve the accuracy of the water sample that is being tested. The main hardship that we faced during the testing process was testing ground water samples which has a contamination of salt water. In some parts of the country, specially the coastal region where the ground water sources are near by the sea, the sources tend to contaminate with sea water and gives a salty taste which could not be used for consumption or domestic purposes. Adding sodium sensors to address this problem and adding calcium sensors to precisely identify the CKDu outbreaks could be considered to be the future works to improve the performance of the system.

In the future it is suggested to release an IOS version of the application targeting to cater the users who cannot access the android version of it. WQI calculations can be improved and better solutions can be given to the users by considering deep factors like environmental pollution and air pollution and how they can directly or indirectly affect the quality of water.

5. REFERENCES

[1] C. B Disanayake, “Water Quality in the dry zone area in Sri Lanka – Some Interesting Health Aspects,” *J. Natn.Sci.Foundation Sri Lanka*, vol. 33, no. 3, p. 161-168, 2005.

[2] Imbulana, K.A.U.S, Wijesekera, Sohan, Neupane, Bhanu, “Sri Lanka. Ministry of Agriculture, Irrigation and Mahaweli Development,” *program and meeting document*, 955-8395-01-3, 2006.

[3] Yuanyuan Wang, Jian Zhou, Kejia Chen, Yunyun Wang, Linfeng Liu, “Water Quality Prediction Method Based on LSTM Neural Network,” 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), 2007.

[4] Sri Lanka - Drinking water crisis (DG ECHO, DMC, media) (ECHO Daily Flash of 23 March 2020) FormatNews and Press Release Source ECHO Posted 23 Mar 2020 Originally published 23 Mar 2020.

[5] Central Environment Authority in Sri Lanka. http://www.cea.lk/web/en/water

[6] P.A.C.T. Perera, T.V. Sundarabarathy, T. Sivananthawer and U. Edirisinghe, “Seasonal Variation of Water Quality Parameters in Different Geomorphic Channels of the Upper Malwathu Oya in Anuradhapura, Sri Lanka,” *Tropical Agricultural Research* vol. 25, no. 2, p. 158 – 170, year 2014.

[7] Hemant Pathak, S.N.Limaye, “Study of Seasonal Variation in Groundwater Quality of Sagar City by Principal Component Analysis,” *E-Journal of Chemistry*, ISSN: 0973-4945; CODEN ECJHAO, 9 January 2011 [Online]. Available: http://www.e-journals.net. [Accessed 1 March 2011].

[8] Sri Lankan Society for Microbiology (SSM), “Contaminated Public well water as a source of sporadic outbreak of enteric infection in Northern Sri Lanka,” *Sri Lankan Journal of Infectious Diseases*, 27 October 2015.

[9] Shie-Yui Liong., Pavel Tkalich, Sundarambal Palani, “An ANN application for water quality forecasting,”. *Marine Pollution Bulletin*, Autonomic and Secure Computing, vol. 56, no. 9, p. 1586- 1597, [Accessed: 16 July 2008], Available: https://doi.org/10.1016/j.marpolbul.2008.05.021

[10] AbdollahTaheri Tizro1, Maryam Ghashghaie1, Pantazis Georgiou, Konstantinos Voudouris, “Time series analysis of surface water quality,” *Journal of Applied Research in Water and Wastewater*, vol. 1, p. 43 – 52, 13 February 2014 [Online]. Available: www.arww.razi.ac.ir. [Accessed 23 March 2014].

[11] MeraBhujal, Available: https://play.google.com/store/apps/details?id=in.gcrs.merabhujal

[12] Water Quality 4Thai, *Pollution Control Department*, Thailand, Weather, Available: https://play.google.com/store/apps/details?id=com.twofellows.thaiwaterqualitymobile

[13] Durudu Omer Faruk, “A hybrid neural network and ARIMA model for water quality time series prediction,” *Engineering Applications of Artificial Intelligence*, vol 23, no. 4, p. 586-594, [Accessed 28 October 2009], Available: https://doi.org/10.1016/j.engappai.2009.09.015.

[14] Mahesh Kumar.Akkaraboyina, prof B.S.N.Raju, “Time Series Forecasting Of Water Quality Of River Godavari,” *Journal of Mechanical and Civil Engineering*, vol. 1, no. 3, p. 39-44, [Accessed: July 2012], Available: www.iosrjournals.org

[15] Joy Parmar, Mosin I Hasan, “Forecasting of River Water Quality Parameters,” *International Journal of Scientific Research in Engineering*, vol. 1, no. 5, [Accessed: May], Available: www.ijsre.in.

[16] S. Emamgholizadeh, H. Kashi, I. Marofpoor, E. Zalaghi, “Prediction of water quality parameters of Karoon River (Iran) by artificial intelligence-based models,” *Islamic Azad University*, vol.11, p.645- 656, [Accessed: 1 October 2013].

[17] T. Papalaskaris, G. Kampas, “Time series analysis of water characteristics of streams in Eastern Macedonia – Thrace, Greece,” *European Water*, vol.57, p. 93- 100, year 2017.

[18] Brownlee, J. (2019). “How to Create an ARIMA Model for Time Series Forecasting in Python,” *Machine Learning Mastery*. Available at: https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python [Accessed 16 Sep. 2019].

6. APPENDICES

Forecastparameter.py

```
from pandas import read_csv
import numpy as np
from sklearn.ensemble import RandomForestRegressor
# Using Skicit-learn to split data into training and testing sets
from sklearn.model_selection import train_test_split
# Import the model we are using
import pickle


dataset = read_csv('../../datasets/dataset_forecastparameter.csv',
index_col=0)

dataset.fillna(0.0, inplace=True)
dataset.mean()

sites = np.unique(dataset[['site_no']].values)
# to be encoded month and site
encode = {'Month': {}, 'site_no': {}}

# Temporary count variable
count = 0
# encode months to numeric values by selecting distinct values from the
dataset
for m in np.unique(dataset[['Month']].values):
    encode['Month'][m] = count
    count = count + 1

# Temporary count variable
count = 0
# encode months to numeric values by selecting distinct values from the
dataset
for s in np.unique(dataset[['site_no']].values):
    encode['site_no'][s] = count
    count = count + 1


# replace the collection values with created encoded values
dataset.replace(encode, inplace=True)

# print(dataset.dtypes)
# dictionary for months
dict_month = encode['Month']

# dictionary for site
dict_site = encode['site_no']


# Labels are the values we want to predict
labels = dataset[
    ['temperature', 'dissolved_oxygen', 'pH', 'turbidity', 'tds', 'ec']]

# Saving label names for later use
```

```
labels_list = list(labels.columns)

# Convert to numpy array
labels = np.array(labels)

# Remove the labels from the features
# axis 1 refers to the columns
features = dataset.drop(
    ['temperature', 'dissolved_oxygen', 'pH', 'turbidity', 'tds', 'ec'],
    axis=1)

# Saving feature names for later use
feature_list = list(features.columns)

# Convert to numpy array
features = np.array(features)

# Split the data into training and testing sets
train_features, test_features, train_labels, test_labels =
train_test_split(features, labels, test_size=0.25,

random_state=42)

print('Training Features Shape:', train_features.shape)
print('Training Labels Shape:', train_labels.shape)
print('Testing Features Shape:', test_features.shape)
print('Testing Labels Shape:', test_labels.shape)

# Instantiate model with 1000 decision trees
rf = RandomForestRegressor(n_estimators=1000, random_state=42)

# Train the model on training data
rf.fit(train_features, train_labels)

# Use the forest's predict method on the test data
predictions = rf.predict(test_features)

# Calculate the absolute errors
errors = abs(predictions - test_labels)

print('Mean Absolute Error:', round(np.mean(errors), 2), 'degrees.')

# Calculate mean absolute percentage error (MAPE)
mape = 100 * (errors / test_labels)

# Calculate and display accuracy
accuracy = 100 - np.mean(mape)
print("\n")
print('Accuracy:', round(accuracy, 2), '%.')

# print(predictions)
print("Input Features :", feature_list)
print("Output Features :", labels_list)

parameters = ['temperature', 'dissolved_oxygen', 'pH', 'turbidity', 'tds',
'ec']
```

```
pickle.dump(rf, open('../../models/model-forecastParameter.pkl', 'wb'))
model = pickle.load(open('../../models/model-forecastParameter.pkl', 'rb'))


def predictParameters(year, month, site):
    if site in sites:
        month = dict_month[month]
        site = dict_site[site]
        inputs = [[year, month, site]]
        parameter_list = model.predict(inputs)
        for a in parameter_list:
            output = [{"temperature": a[0],
                       " Dissolved Oxygen": a[1],
                       "pH": a[2],
                       "turbidity": a[3],
                       "Total Dissolved Solid": a[4],
                       "Conductivity": a[5]}]
            return output


    else:
        return 'We will be reach soon'


# input data
print(predictParameters(2021, 'Oct', 'Galnewa'))
```

wqiPrediction.py

```
# emsemble methods are used here to combine predictions of estimators built
with the random forest algorithm
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import train_test_split
import numpy as np
from sklearn import metrics
from pandas import read_csv
import pickle

# Pandas are an open source data analysis and manipulation library
import pandas as pd

dataset = read_csv('../../datasets/dataset_wqi.csv')

# Import and read data from the dataset
# dataset = pd.read_csv("WaterData1.csv")
X = dataset.drop('WQI', axis=1)
y = dataset['WQI']



# Displaying the read data from the csv file
print(dataset.head())
```

```
print("\n")

dataset.describe()

# pH has some missing values
# Fixing these missing values using mean
# dataset["pH"].fillna(dataset.pH.mean(), inplace=True)
# dataset["Temperature"].fillna(dataset.Temperature.mean(), inplace=True)

dataset.describe()
print(X)
# Dividing the dataset in 20:80 ratio where 20% for testing and 80% for
training
X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

print('The Train Features: ', X_train.shape)
print('The Train Lables: ', Y_train.shape)
print('The Test Features: ', X_test.shape)
print('The Test Lables: ', Y_test.shape)
print("\n")

# Assigning the random forest classifier from sklearn.ensemble
# randClass = RandomForestClassifier(n_estimators=400)
# randClass.fit(X_train, Y_train)

modelRF = RandomForestRegressor(n_estimators=1000, oob_score=True,
random_state=42)
modelRF.fit(X_train, Y_train)

# Predicting using the test dats
Prediction = modelRF.predict(X_test)
Prediction


# Evaluating Training Set

mae = metrics.mean_absolute_error(Y_train, modelRF.predict(X_train))
mse = metrics.mean_squared_error(Y_train, modelRF.predict(X_train))
raq = metrics.r2_score(Y_train, modelRF.predict(X_train))
rmse = np.sqrt(mse)

print('Evaluating Training Set: ')
print('MAE: ', mae)
print('MSE: ', mse)
print('RMSE: ', rmse)
print('R^2: ', raq)
print("\n")

# Evaluating Test Set

mae = metrics.mean_absolute_error(Y_test, modelRF.predict(X_test))
mse = metrics.mean_squared_error(Y_test, modelRF.predict(X_test))
raq = metrics.r2_score(Y_test, modelRF.predict(X_test))
rmse = np.sqrt(mse)

print('Evaluating Test Set: ')
```

```
print('MAE: ', mae)
print('MSE: ', mse)
print('RMSE: ', rmse)
print('R^2: ', raq)

# pickle.dump(model, open('model.pkl', 'wb'))
# model1 = pickle.load(open('model.pkl', 'rb'))

pickle.dump(modelRF, open('../../models/model_wqi.pkl', 'wb'))
model = pickle.load(open('../../models/model_wqi.pkl', 'rb'))


def predictWQI(Temperature, pH, Turbidity, Conductivity, TDS):
    inputs = [[Temperature, pH, Turbidity, Conductivity, TDS]]
    output = model.predict(inputs)
    return output


print(predictWQI(24.14855511,7.058125,139.8151967,0.41185,137.0188928))
```

ckduPrediction.py

```
import pandas as pd
import numpy as np
import pickle
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split

# %matplotlib inline

data = pd.read_csv("../../datasets/dataset_ckdu.csv")
data.head(10)

sites = np.unique(data[['site_no']].values)

encode = {'site_no': {}}

count = 0

for m in np.unique(data[['site_no']].values):
    encode['site_no'][m] = count
    count = count + 1

data.replace(encode, inplace=True)

dict_site = encode['site_no']

X = data.drop(['status', 'patients', 'hardness', 'Year', 'Month',
'dissolved_oxygen'], axis=1)
y = data['status']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)

"""Polynomial SVM"""

print(data.head())
```

```
svclassifier = SVC(kernel='sigmoid', degree=8)
svclassifier.fit(X_train, y_train)

y_pred = svclassifier.predict(X_test)

from sklearn.metrics import accuracy_score

accuracy_rate = 100.0 * accuracy_score(y_test, y_pred)
print("The test data accuracy : " + str(accuracy_rate))

from sklearn import metrics

mae = metrics.mean_absolute_error(y_train, svclassifier.predict(X_train))
mse = metrics.mean_squared_error(y_train, svclassifier.predict(X_train))
raq = metrics.r2_score(y_train, svclassifier.predict(X_train))
rmse = np.sqrt(mse)

print('MAE: ', mae)
print('MSE: ', mse)
print('RMSE: ', rmse)
print('R^2: ', raq)

pickle.dump(svclassifier, open('../../models/model_ckdu.pkl', 'wb'))
model = pickle.load(open('../../models/model_ckdu.pkl', 'rb'))


# site_no,temperature,pH,turbidity,tds,ec
def predictCkduRisk(site, temperature, ph, turbidity, tds, ec):
    if site in sites:
        site = dict_site[site]
        inputs = [[site, temperature, ph, turbidity, tds, ec]]
        output = model.predict(inputs)
        return output

    else:
        return 'ERROR'


print(predictCkduRisk("Malabe", 27.23644587, 6.943591428, 14.1575216,
162.8088854, 0.269126173))
```

app.py

```
import numpy as np
from pandas import read_csv
from flask import Flask, request, jsonify, render_template
import pickle



app = Flask(__name__)
model_ForecastParameter = pickle.load(open('models/model-
forecastParameter.pkl', 'rb'))
model_ckdu= pickle.load(open('models/model_ckdu.pkl', 'rb'))
```

```
model_wqi = pickle.load(open('models/model_wqi.pkl', 'rb'))

feature_list = ['Year', 'Month', 'site_no', 'wqi']

labels_list = ['temperature', 'dissolved_oxygen', 'pH', 'turbidity', 'tds',
'ec']
parameters = ['temperature', 'dissolved_oxygen', 'pH', 'turbidity', 'tds',
'ec']


############################################IT16170780######################
##################################
# predicting water quality parameter for a given month of a year
def predictWaterQualityParameterForAMonth(year, month, site):
    dataset = read_csv('datasets/dataset_forecastparameter.csv', index_col=0)
    output = {"temperature": 0, "Dissolved Oxygen": 1, "pH": 2, "turbidity":
3, "Total Dissolved Solid": 4, "Conductivity": 5}
    sites = np.unique(dataset[['site_no']].values)
    # to be encoded
    encode = {'Month': {}, 'site_no': {}}
    # Temporary count variable for month
    count = 0
    # encode months to numeric values by selecting distinct values from the
dataset
    for m in np.unique(dataset[['Month']].values):
        encode['Month'][m] = count
        count = count + 1

    # Temporary count variable for site
    count = 0
    # encode months to numeric values by selecting distinct values from the
dataset
    for s in np.unique(dataset[['site_no']].values):
        encode['site_no'][s] = count
        count = count + 1

    # replace the collection values with created encoded values
    dataset.replace(encode, inplace=True)
    # dictionary for months
    dict_month = encode['Month']
    # dictionary for site
    dict_site = encode['site_no']
    if site in sites:
        month = dict_month[month]
        site = dict_site[site]
        inputs = [[year, month, site]]
        parameter_list = model_ForecastParameter.predict(inputs)
        for a in parameter_list:
            output = [{"temperature": a[0], "Dissolved Oxygen": a[1], "pH":
a[2], "turbidity": a[3], "Total Dissolved Solid": a[4],
                       "Conductivity": a[5]}]
            return output

    else:
        return 'V404'

############################################################################
```

```
########################################


########################################IT16168800###########################
########################################
def predictCKD(site, temperature, ph, turbidity, tds, conductivity):
    dataset = read_csv('datasets/dataset_ckdu.csv', index_col=0)
    output = {"status": 0}
    sites = np.unique(dataset[['site_no']].values)
    # to be encoded
    encode = {'site_no': {}}

    # Temporary count variable for site
    count = 0
    # encode months to numeric values by selecting distinct values from the
dataset
    for s in np.unique(dataset[['site_no']].values):
        encode['site_no'][s] = count
        count = count + 1

    # replace the collection values with created encoded values
    dataset.replace(encode, inplace=True)
    # dictionary for site
    dict_site = encode['site_no']
    if site in sites:
        site = dict_site[site]
        inputs = [[site, temperature, ph, turbidity, tds, conductivity]]
        parameter_list = model_ckdu.predict(inputs)
        formatParam = int(parameter_list[0])
        output = [{"status ": formatParam}]
        return output

    else:
        return 'V404'


def calclAvgPatientsForSite(site):
    dataset = read_csv('datasets/dataset_ckdu.csv', index_col=0)
    sites = np.unique(dataset[['site_no']].values)
    patients = dataset.loc[dataset['site_no'] == site, 'patients']
    averagePatients = np.mean(patients)
    return averagePatients
    print(averagePatients)

#############################################################################
########################################


########################################IT16165694###########################
########################################
def predictWQI(Temperature, pH, Turbidity, Conductivity, TDS):
    inputs = [[Temperature, pH, Turbidity, Conductivity, TDS]]
    parameter_list = model_wqi.predict(inputs)
    formatParam = int(parameter_list[0])
    output = [{"WQI": formatParam}]
    return output
#############################################################################
```

```
########################################

@app.route('/')
def index():
    return "Water Quality Parameter Prediction"


@app.route('/resultParameterForMonth', methods=['GET'])
def predictAllParameters():
    body = request.get_data()
    header = request.headers

    try:
        year = int(request.args['Year'])
        site = str(request.args['Site'])
        month = str(request.args['month'])
        if (year != None) and (site != None) and (month!= None):
            res = predictWaterQualityParameterForAMonth(year, month, site)
            print(res)
        else:
            res = {
                'success': 'True',
                'message': 'Incorrect input'
            }
    except:

        res = {
            'success': 'False',
            'message': 'Unknown Error'
        }

    return jsonify(res)


@app.route('/resultCKD', methods=['GET'])
def predictRiskCKD():
    body = request.get_data()
    header = request.headers
    try:
        temperature = float(request.args['temperature'])
        site = str(request.args['site'])
        turbidity = float(request.args['turbidity'])
        ph = float(request.args['ph'])
        tds = float(request.args['tds'])
        conductivity = float(request.args['ec'])

        avgPatients = calclAvgPatientsForSite(site)

        if (temperature != None) and (site != None) and (turbidity != None)
and (ph != None):
            output = predictCKD(site, temperature, ph, turbidity, tds,
conductivity)
            res = {
                "result": output[0],
                "avgPatients": avgPatients
            }
            print(res)
```

```
        else:
            res = {
                'success': 'True',
                'message': 'Incorrect input'
            }

    except:

        res = {
            'success': 'False',
            'message': 'Unknown Error'
        }

    return jsonify(res)


@app.route('/resultWQI', methods=['GET'])
def predictionWQI():
    body = request.get_data()
    header = request.headers
    try:
        temperature = float(request.args['Temperature'])
        ph = float(request.args['pH'])
        turbidity = float(request.args['Turbidity'])
        conductivity = float(request.args['Conductivity'])
        tds = float(request.args['TDS'])

        if (temperature != None) and (ph != None) and (turbidity != None) and
(conductivity != None) and (tds != None):
            res = predictWQI(temperature, ph, turbidity, conductivity, tds)
            print(res)
        else:
            res = {
                'success': 'True',
                'message': 'Incorrect input'
            }

    except:

        res = {
            'success': 'False',
            'message': 'Unknown Error'
        }

    return jsonify(res)


# for mobile connection
if __name__ == "__main__":
    app.run(host='0.0.0.0', port=5000)

# if __name__ == "__main__":
#     app.run(host='0.0.0.0', port=5000, debug=True)

# if __name__ == "__main__":
#     app.run(debug=True, port=5000)
```