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

Nibraz MM

IT16165694

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 SAFE CONSUMPTION OF GROUND WATER

Nibraz MM

IT16165694

Dissertation submitted in partial fulfilment of the requirements for
the Special Honors degree of Bachelor of Science
in Information Technology
Specialized in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

September 2020

DECLARATION

I declare that this is my own work and dissertation does not incorporate without acknowledgment of any material previously submitted for a Degree or Diploma in SLIIT or any other university or institute of higher learning. To the best of my knowledge and belief the document does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Project ID : 2020-164

Student Name	Registration No	Signature
Nibraz M M	IT16165694	

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

Signature of the supervisor : ....

Date : ....

ABSTRACT

Water dominates the surface of the earth and it is an indispensable element when making up the human body. A healthy lifestyle explicates the intake of healthy food, water, air and making sure that we are around all these building blocks which are safe and clean is essential. Although different people use different types of water from different sources, it is proven that the groundwater sources satisfy the necessities of the majority of people living in Sri Lanka, and extra care should be grasped when utilizing this direct water from sources to avoid the consumers from acquiring fatal or non-fatal diseases. Taking this problem into consideration, it is comprehended that the importance of water quality must be spoken out to the public, and the process of making sure that the quality of water used in day to day consumption falls under a safe range.

One of the paramount solutions for this issue is to propose a fast and easy method using a technological blend to replace the manual and time-consuming chemical methods which are currently practiced in Sri Lanka. “E-Tongue – a smart device to identify the quality of a ground water sample” is a smart solution which hopes to address the ongoing subject of identifying the water quality with the use of technology. This device uses a Machine Learning (ML) model by using Water Parameters as the inputs. The output of this model, the Water Quality Index (WQI) will be used to clearly determine the quality of the water sample obtained from a water source. The final outcome of this attempt is to be showcased in the form of a mobile application.

Keywords: Water quality, Machine Learning (ML), water parameters, Water Quality Index (WQI).

ACKNOWLWDGEMENT

First and foremost, I would like to express my sincere gratitude to my supervisor Ms. Chathurangika Kahandawaarachchi for the constant guidance and support which helped me at all times for the successful completion of my undergraduate research.

Besides my supervisor, my sincere thanks also goes 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. My 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.

Finally, I express my sense of gratitude to my family, friends, to one and all, who directly or indirectly have extended their support throughout this venture.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Meaning
ML	Machine Learning
WQI	Water Quality Index
CKD	Chronic Kidney Disease
CKDu	Chronic Kidney Disease of unknown etiology
AChE	Acetylcholine Esterase
DNN	Deep Neural Networks
SVM	Support Vector Machines
NN	Neural Networks
WHO	World Health Organization
ANN	Artificial Neural Network
PFA	Principal Factor Analysis
TOC	Total Oxygen Content
THM	Total Heavy Metal
KNN	K-Nearest Neighbors
RFR	Random Forest Regressor
TDS	Total Dissolved Solids
NWSDB	National Water Supply & Drainage Board
NSFWQI	National Sanitation Foundation Water Quality
CCMEWQI	Canadian Council of Ministers of the Environment Water Quality Index
OWQI	Oregon Water Quality Index
API	Application Programming Interface
UI	User Interface
AWS	Amazon Web Service
MAE	Mean Absolute Error
MSE	Mean Square Error
RMSE	Root Mean Square Error
CNN	Convolutional Neural Network

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.

Water, one key element that is found ample in nature and is very important to living kind and yet it is hard to find a good source that is safe for consumption, especially for mankind. The quality of water is an overall description of the biological, chemical and physical characteristics of water in connection with intended uses and a set of standards [1]. Hence the evaluation of quality of water can be defined as the assessment of the biological, chemical and physical properties of water in accordance to natural quality and intended uses.

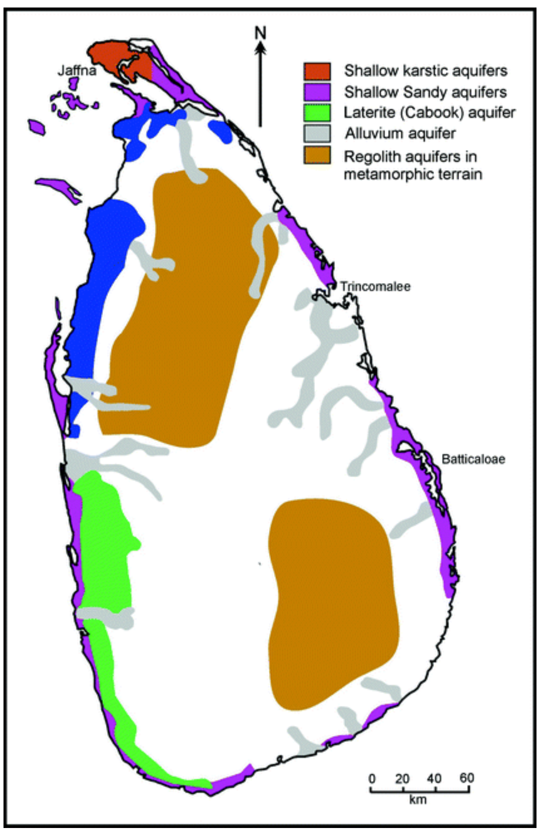


Figure 1.0: Types of aquifers in Sri Lanka

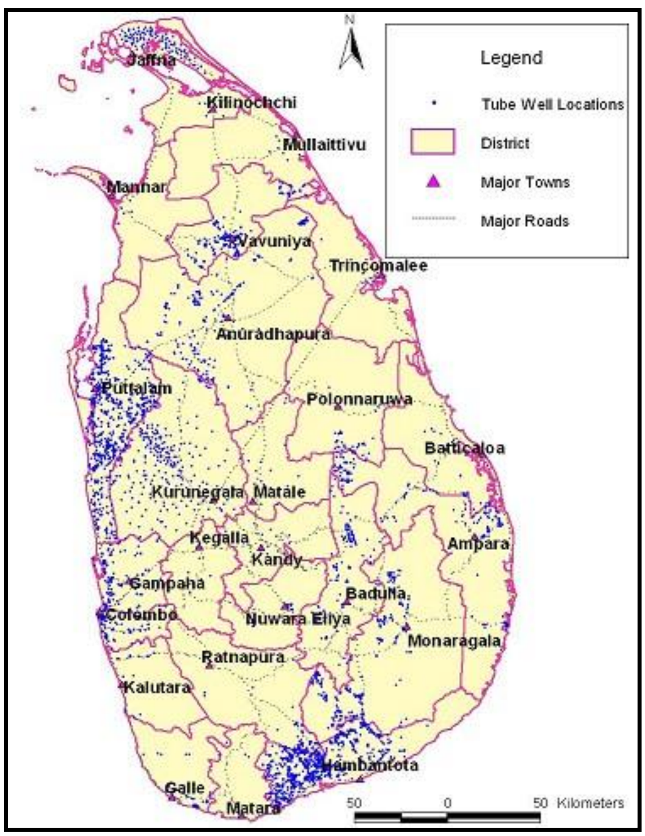


Figure 1.1: Distribution of tube well sites in Sri Lanka

Alongside the lack of safe water source for consumption, Sri Lankans have been facing the consequences of drinking unsafe water and one of the notorious diseases that has been identified as fatal not only to people in Sri Lanka but also to the entire globe is the Chronic Kidney Disease (CKDu) [2].

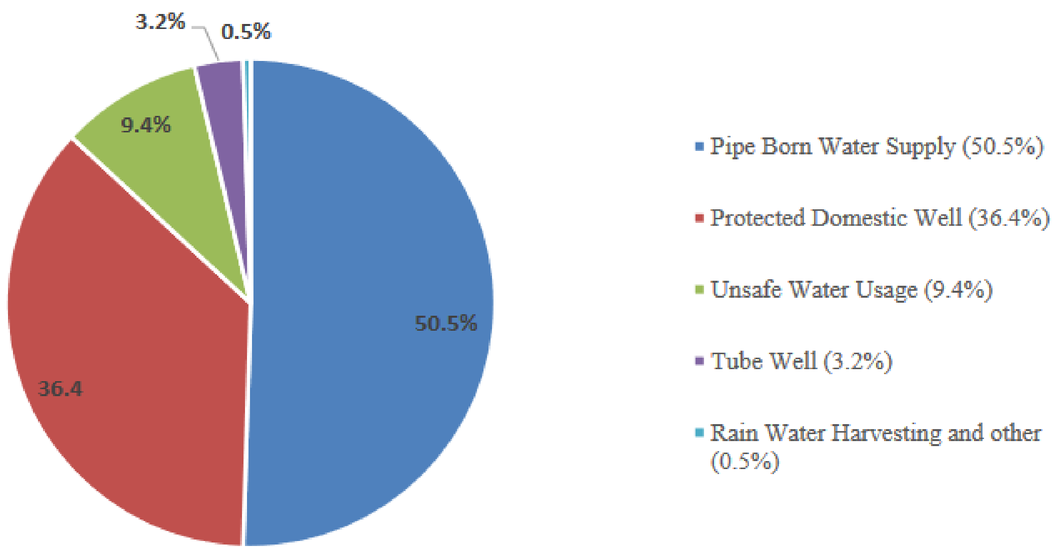


Figure 1.2: Safe Drinking Water Coverage in Sri Lanka by sources-2018

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.

Through a thorough background study, a need of a smart automated system which could be used to predict the water quality has been recognized. As smartphones have invaded many of our lives, it is hardly unlikely to find a user who doesn’t have an access to a smart phone and mobile computing. With the growth of internet and advancement of smart phones that could be used to perform precisely anything, it was understood that to cater more users, a mobile application development would be the best interest to act as the communication mode where the system could be delivered and has a potential to reach the users.

1.2 Background literature

1.2.1 An overview of water borne diseases in Sri Lanka

As a country which is on the path towards industrialization, Sri Lanka has been recognized as a developed country with plenty of natural resources. However, there are some areas which were un touched when making this country and its people safe and secure. The issues that revolves around drinking water safety is one scenario which is present among the locals and have been untouched for centuries now. Although the government has taken necessary action in introducing water supplies to the people, there are some percentage of people who still depends on water sources such as lakes and ground water sources such as wells and tube wells. These communities who uses ground water sources specially for drinking purposes are at the stake of exposing themselves to vital diseases.

Among the spectrum of diseases that could be caused by consumption of unsafe water, one which stands out due to its fatal state is CKDu. Apart from CKDu, several documentaries and researches reveal that some conditions like diarrhea which people consider as a non-critical, simple and curable ailment can be one fatal condition and seize lives if left untreated [3]. This extreme situation puts the people of Sri Lanka to be more cautious when dealing with water for usage specially when it comes to drinking and domestic uses. One group of people at the verge of these risks are the ones who solely depend on natural water sources and ground water sources.

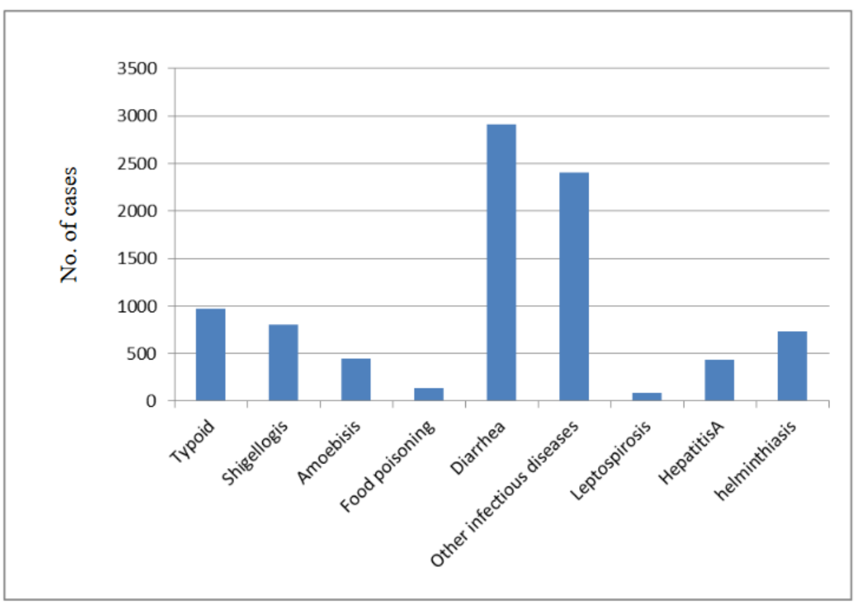


Figure 1.3: The trend of water-borne diseases in Ampara district -2015

According to the study conducted about water borne diseases in Ampara district [4], several interesting, yet shocking results were drawn based on surveys and researches. An optimum number was observed for diarrhea and other infectious disease which was a common problem faced by the people who resides in that area. The reason was observed as these people live near lagoons and the contaminated water sources.

Another common water borne disease which was observed in Ninthavur according to the RDHS report is Typhoid. It is most commonly caused by *Salmonella typhi* bacteria and very rarely caused by *Salmonella paratyphi*. An approximate of 30% of identified cases were observed in this area. The bacteria is commonly deposited in food or water by a human carrier and then continues spreading to others in the targeted area.

On the other hand, Shigellosis which is one dangerous and highly contagious disease caused by *Shigella* bacteria develops fever, diarrhea and ends up infecting the human colon. The areas including Akkaraipattu has seen a 21% infection which has raised common concerns among the locals. Other diseases like Amoebiasis with 13%, hepatitis with 14% and helminthiasis with 17% were observed respectively in the district in 2015.

As the above study suggests that a high percentage of the population has been suffering from these hazardous diseases and the research too suggests some common reasons for the observations such as:

- Deficiency of drinking water facilities
- Lack of safe water for consumption and domestic purpose, poor hygiene and poor sanitation
- Lack of awareness and knowledge among people
- Unsafe proximity between wells and latrines
- Improper drainage system
- Poor access to health services

A need of proper solution could be reflected through these causes that were mentioned above in order to make the society and its people to live in a way where health is mostly assured.

All these hazardous impact on people and society questions the overall ideas of water purification, people's knowledge on the precautions, the accuracy of the water quality predictions that are done by the authorities who supplies water and most importantly a ground water consumer's knowledge on the quality of water they are consuming. In 2015, in another research which was conducted in Jaffna, it was mentioned out of 200 participants who contributed, 87% of people had been using water from sources which were contaminated with *Coliforms* and *Escherichia coli* [5].

Apart from the above-mentioned bacterial diseases, Dengue and Cholera is also one extreme disease that is mostly fatal and is considered as a water borne disease. As observed through the study [4], these were much challenged water-borne disease in the study zone. As observed based on statistics, dengue is mostly seen at the beginning and end of a year where cholera is observed during the drought season. These seasonal correlated diseases were hard to comprehend since there is anyways a chance of getting either one of these disease during the entire period of a year.

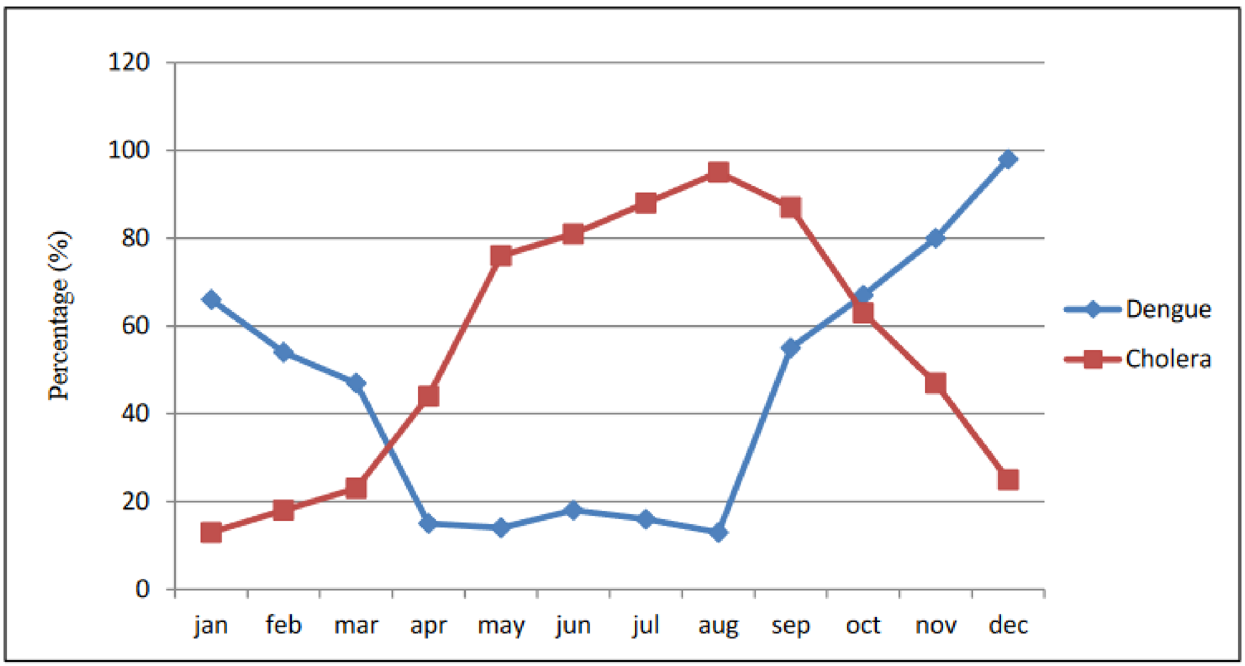


Figure 1.4: Dengue and cholera spread in coastal areas of Ampara District (2015)

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.

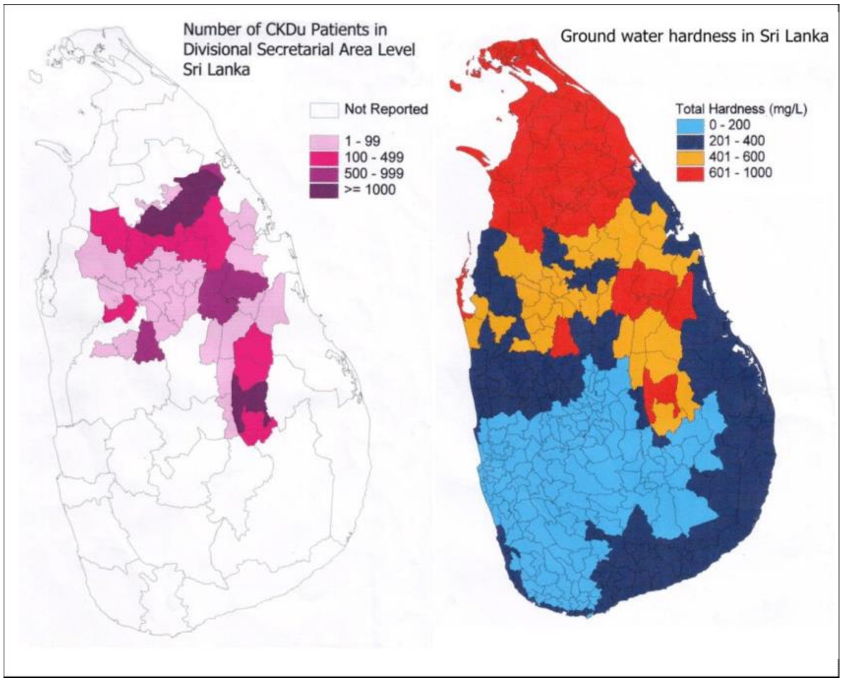


Figure 1.5: Geographical distribution of patients with CKDu and ground water hardness in Sri Lanka

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].

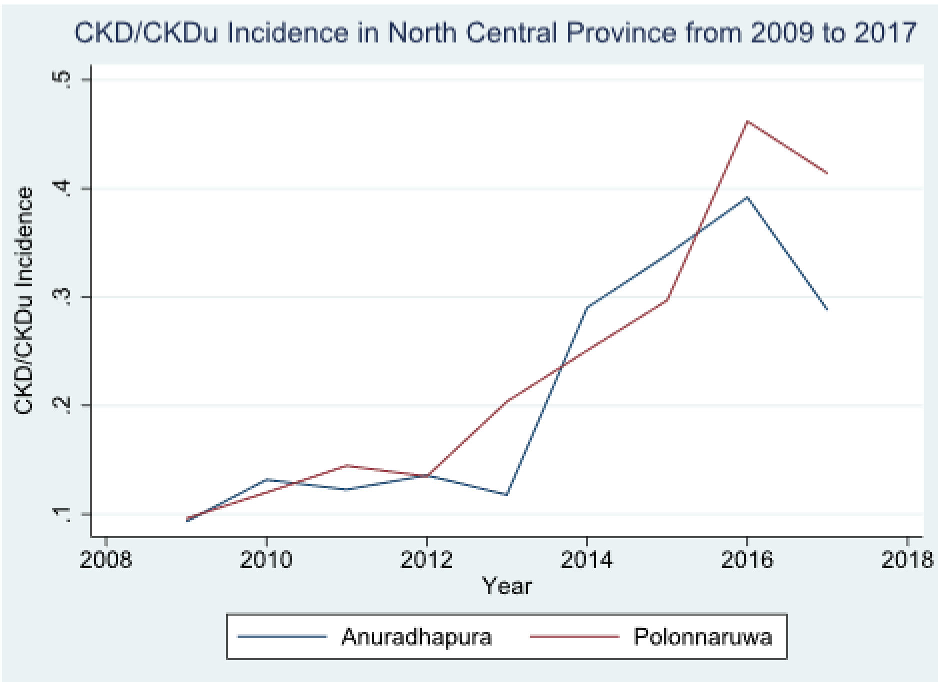


Figure 1.6: Incidence of CKD/CKDu patients in Anuradhapura and Polonnaruwa districts

Although necessary actions have been taken by the government and health care sectors to reduce the occurrence of CKDu in the country, specially 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 Water quality measurement steps followed by the government sectors

Considering the health of the people and development factors, government sectors like National Water Supply and Drainage Board and National water Resources Authority are making necessary steps in checking and maintaining the quality of the supplying water. Although these testing steps are accurate and expensive the time factor deeply implies a negative point in it. In a nutshell, these accurate and expensive testing processes are time consuming.

With the use of many expensive tools, sensors and chemical methods a new water source which is newly found was tested. This process not only applies to a newly found source but also to the newly dug wells and tube wells upon the request of the locals. Once a sample is collected from the water source, the sample is normally taken to the laboratory and the testing processes normally starts. The chemists normally start the tests with simple parameters such as pH and temperature and move on to more complex ones like mineral or ion level measurements which requires chemicals or electrodes.

One hardship that was observed in this manual testing by the authority is that the final report will be given stating the readings of each of the parameters and the reference range they should be in. the authorities normally doesn’t give a final conclusion on the quality of the water they have tested and it is the user’s decision whether to proceed with the source or abort it.

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].

1.3 Research Gap

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.

The final outcome of the above-mentioned projects has been implemented and tested but it had never reached the industry as there were no any medium to publish and allow the people or the industrial experts to actually use the end product for a good cause. Through a thorough analysis of the existing projects, it could be understood that:

- All the existing researches have been solely conducted for the research purpose only where no any implementation of systems for users were found.
- All the existing researches have focused more on building complex models for only surface water such as rivers.

The researchers can obtain certain idea about the quality of river water they have been tested.

Through the comprehensive background study conducted, it could be understood that this procedure can only be considered accurate up to a certain extent due to various reasons such as:

- Sri Lanka as a country is far behind the concept of WQI where it could be used to identify the quality of water in more accurate way.
- People in Sri Lanka are more likely to depend on ground water than surface water and yet face water borne diseases due to the lack of knowledge about the water purity.

Taking these factors into consideration, as an initial step of introducing new terms and methods of measuring the quality of ground water, a system is implemented.

The implemented system will effectively bring a new perspective to the water quality testing by involving Artificial Intelligence in to predicting the quality of ground water from any source.

Name	Physical and Chemical	Bacteriology	Metals and Heavy Metals	Algae	TOC	THM	Pesticides	Oil and Grease	Waste Water	Filter Media
National Water Supply & Drainage Board, Thelawala Road, Ratmalana	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
National Water Supply & Drainage Board, Wathhimi Road, Kurunegala	✓	✓								✓
National Water Supply & Drainage Board, New Vishaka Road, Bandarawela.	✓	✓	✓	✓				✓	✓	
National Water Supply & Drainage Board, Manager Office, Nupe, Matara.	✓	✓						✓	✓	✓
Regional Laboratory, National Water supply &Drainage Board, Vavunia.	✓	✓								
National Water Supply & Drainage Board, Sivan Pannai Road, Jaffna.	✓	✓	✓		✓			✓	✓	

(Table 1.0 – few manual water quality testing labs present in Sri Lanka [17])

As an initiative to reduce the manual testing procedures mentioned in the table above and to speed up the ground water quality determination, a device is implemented with the use of machine learning which will predict the WQI of samples and provides suggestions on how to make the water from a particular sample more suitable for drinking and domestic uses.

1.4 Research Problem

One of the vital sources of water for people living in Sri Lanka is ground water. According to statistics, about 94% of the population directly or indirectly depend on the natural water sources whereas the remaining 6% purchase water from vendors [18]. Sri Lanka, being well known for its rich history based on irrigation and agriculture has put forward the need of water for survival to a more important place. With the increase of water borne diseases, the demand for the safety of the consuming water is highly valued.

Table: Quality management testing details between treatment processes

No	Test ID	Frequency	Test parameters	Test methods	Record details
	T1	2hrs intervals	pH	pH meter	Daily water quality record (2 hours)
			Turbidity	Turbidity meter	
			Colour	spectrometer	
			Conductivity	conductivity meter	
		Once a week	pH	pH meter	Jar test report
			Turbidity	Turbidity meter	
			Colour	spectrometer	
2	T2	2hrs intervals	pH	pH meter	Daily water quality record (2 hours)
3	T3	8hrs intervals	pH	pH meter	Daily water quality record
			Turbidity	Turbidity meter	
			Colour	spectrometer	
4	T4	2hrs intervals	Turbidity	pH meter	Daily water quality record (2 hours)
		8hrs intervals	Al-total	spectrophotometer	
5	T5	8hrs intervals	pH	pH meter	Daily water quality record
			Turbidity	Turbidity meter	
			Colour	spectrometer	
6	T6	2hrs intervals	Turbidity	Turbidity meter	Daily water quality record (2 hours)
7	T7	2hrs intervals	Turbidity	Turbidity meter	Daily water quality record (2 hours)
		8hrs intervals	pH	pH meter	Daily water quality record
			Turbidity	Turbidity meter	
			Colour	spectrometer	
8	T8	2hrs intervals	pH	pH meter	Daily water quality record (2 hours)
			Turbidity	Turbidity meter	
			Colour	pH meter	
			Conductivity	conductivity meter	
			Residual chlorine	Pailing test kit	
		8hrs intervals	Al-total	spectrophotometer	Daily water quality record

Figure 1.7: Quality management testing process details between treatment processes

As discussed above through the background study, the deficiency of a system to determine the quality of the groundwater people are reaching for in day to day basis is one black mark to the legacy we carry.

The techniques that are currently in use to identify water quality in Sri Lanka vary from the methodologies followed by other developed countries and are also less accurate as it still remains a developing country. Based on the evidence collected during the paid visit to Vavunathivu water treatment plant, Batticaloa above in figure 8, it is understood that water quality determination process is a lengthy process which takes 3 days minimum.

The concept of WQI is not utilized here as it involves a complex mathematical based approach for the identification of water quality. Therefore, the introduction of WQI into water quality identification will be an essential to help mitigate issues related to the accuracy of the identified grade of water. A suitable mathematical model must be identified and trained in order to perform an accurate task. Furthermore, the prediction of what will be the quality of that water source in future will help the user to draw up necessary precautions to avoid any unnecessary consumption.

E-Tongue is targeted to solve this time consumption issue by introducing an intelligent system to predict the quality of ground water samples. In addition, the system tends to predict the quality of the water parameters based on season and the possibility of a CKDu outbreak. The specially designed smart device which includes sensors are used to read the values of separate sets of the water quality parameters which are fed to the ML model to predict the WQI.

1.5 Research Objectives

1.5.1 Primary Objectives of the research:

A thorough, in depth study of the past work related to Water Quality and Water Quality Index was a necessary step in carrying out this research. Once clearly understood the basics and the importance of these topics and how they are correlated, the main objectives were determined:

- To study the different types of water borne diseases that are found in Sri Lanka.
- To study the different techniques of machine learning that could be applied to this research project.
- To study the different water quality parameters that could be used as the input to the mathematical model that is trained.
- To train different Machine learning models to see and compare the performances and the accuracy of the available models in the study area.
- To select the best Machine learning model with a highest accuracy and lowest error percentage.
- To suggest suitable methods of purifying water based on the obtained WQI value.

1.5.2 Secondary Objectives of the research:

- To identify a data set which could be used to train, study and evaluate the ML model.
- To encourage the water quality authorities to test and proceed with the proposed system which is believed to be efficient.
- Empower the locals as well as the authorities with ease access to the new technology.
- To make awareness among people and educate them on the importance of knowing the quality of water they access on a day to day basis, the proper way to purify water and the consequences of the environmental pollution.

2 RESEARCH METHODOLOGY

2.1 Methodology

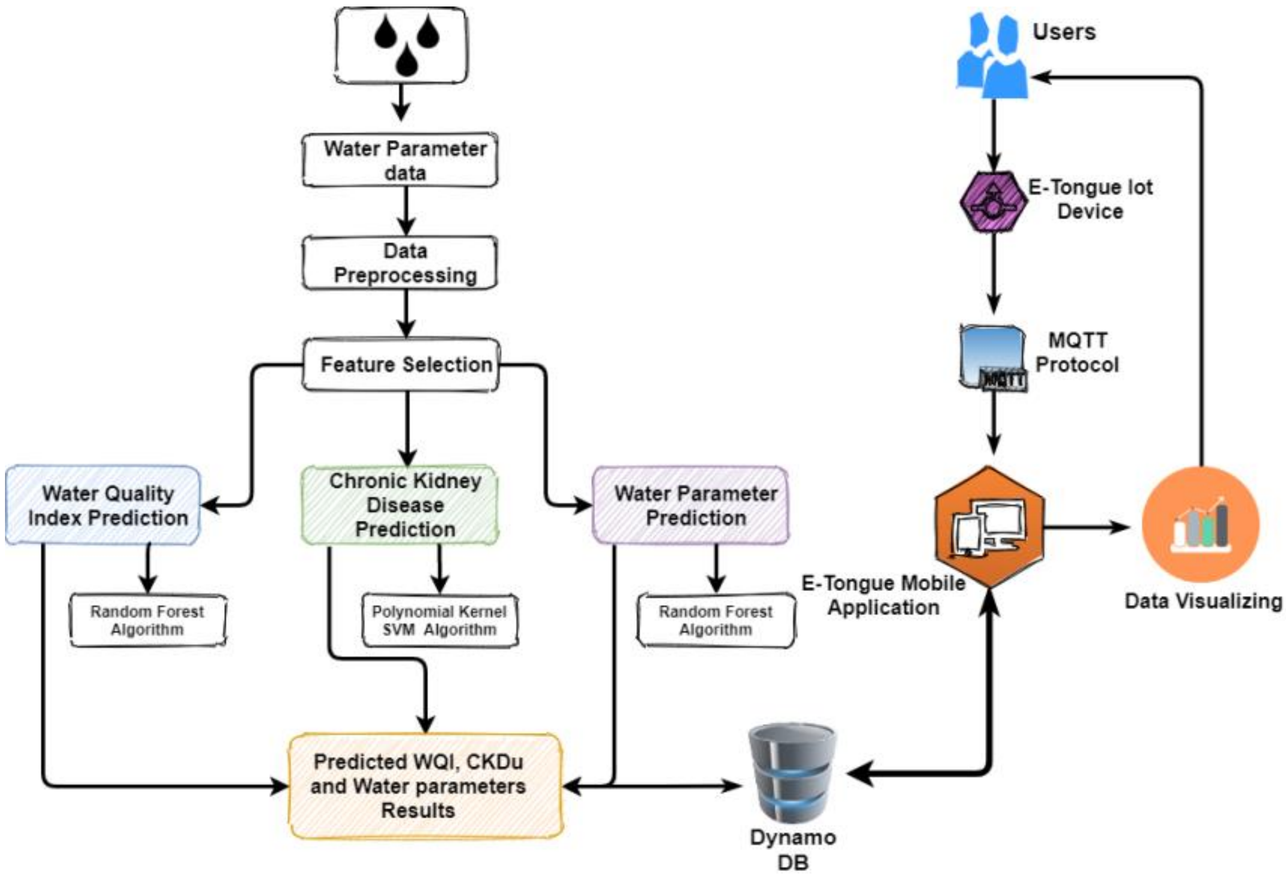


Figure 2.0: System overview diagram

E-Tongue is a smart tool to predict the quality of ground water for a safe consumption and usage. This intelligent system consists of 3 main functionalities:

- Identification of the quality of groundwater by predicting the WQI value.
- Forecasting water quality parameters.
- Prediction of the risk of exposure to CKDu 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.

Identification of water quality by predicting WQI using Machine Learning techniques

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.

Upon breaking down the work done, the following flow chart depicts a clear view of the structure of the work that needs to be followed:

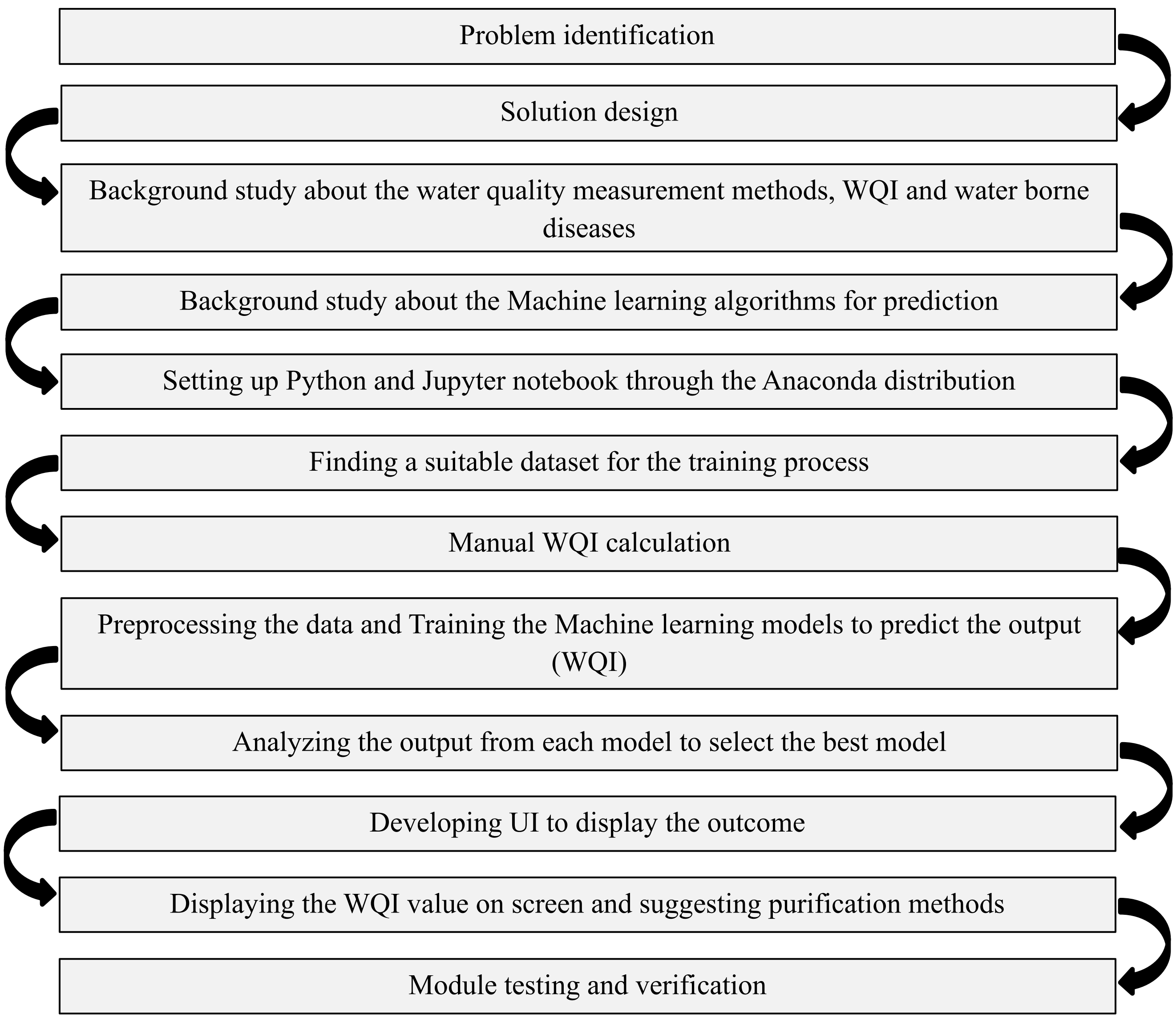


Figure 2.1: Work breakdown structure

2.1.1 Problem Identification

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.

2.1.2 Designed Solution

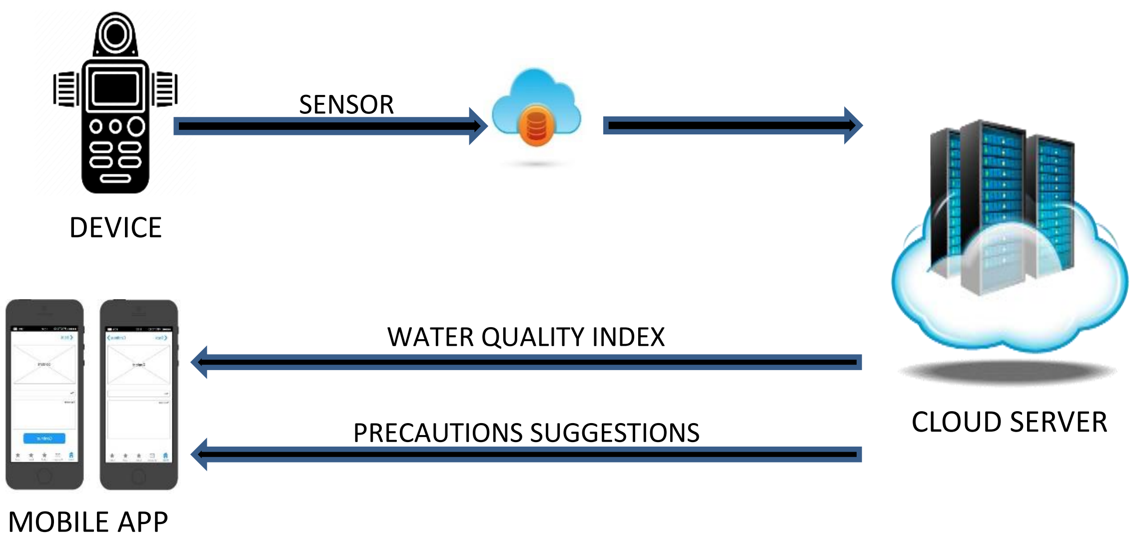


Figure 2.2: Overview diagram of the component

The solution proposed and designed to this component includes a mobile application that is connected to the device which acts as the interface for the used communication. The connected device obtains the sensor readings and the predicted WQI value along with the suggestions from the cloud server to display it in the interface to the user.

With today's technological advancement, abundance of mathematical models that are present in the industry have made ways for a variety of solutions that could be tried out. When designing a solution for the component, these varieties have taken into consideration and a look for the most efficient and accurate model among them was made mandatory. Training several models with the same set of data and selecting the best model that gives the highest accuracy is the best step that could be followed.

The initiation step would be to plan on the mobile application since it is mandatory to design wireframes which could be used to evaluate the user experience and the flow of the design. The user interfaces were drawn as follows

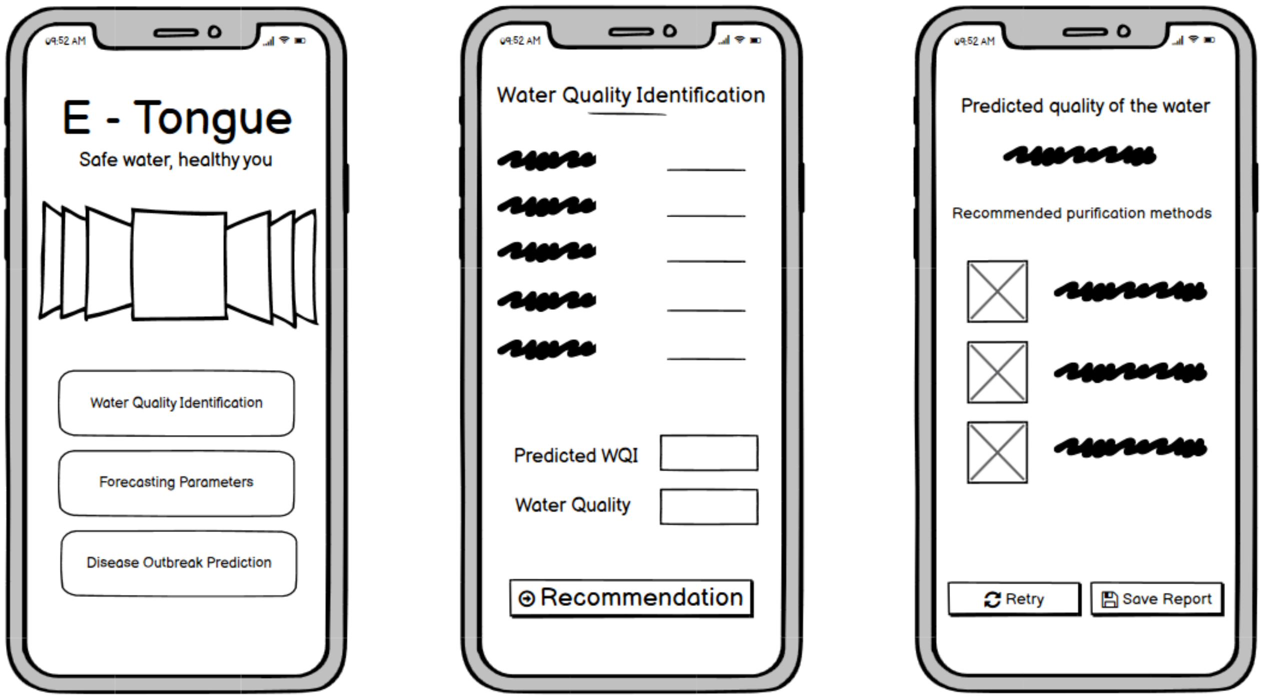


Figure 2.3: Designed wireframes for the user interface of the mobile application

The mobile application represents the entire result of the project. The requirements and the flow of the process must be smooth, ethical and logical. There are several user requirements that needs to be assimilated

1. The user must be able to connect the mobile device to the designed device without any difficulties.
2. The user then must be able select the functionality that is offered through the application by clicking it.
3. Upon selecting the water quality prediction functionality, the user must be redirected to the page where the sensor readings and the predicted WQI values are displayed.
4. The precaution recommendation must be displayed once the user wishes to know more about the actions that could be followed to make the sample water from the source more usable.
5. The user must be able to take back the test.

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.

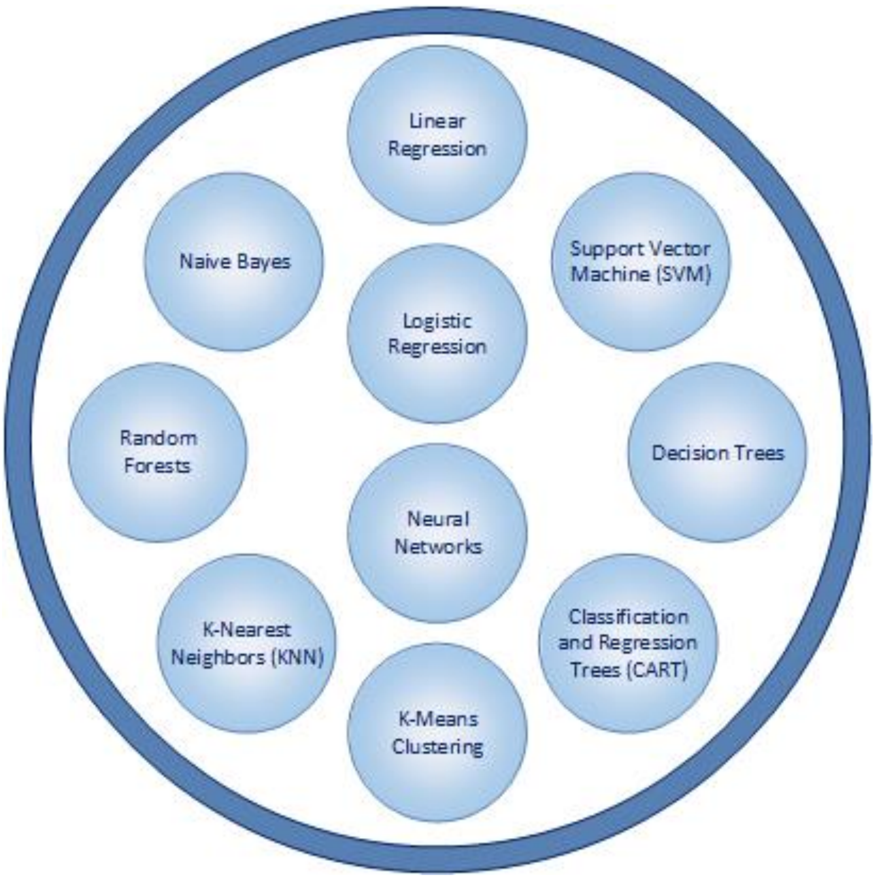


Figure 2.4: Commonly used Machine Learning algorithms and techniques

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 2.5: Linear regression training as a learning activity

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

2.1.3 Data gathering for model training process

Involvement of machine learning in any scenario needs plenty of data specially for the model training process. In a technical quote, more the data, better the accuracy of the training model. Since one of the objectives of this project involves training multiple models to select the best one, it is necessary that the data that is fed into the model are of enough size and meets the standards.

For the motive of this study, the look for a suitable dataset began in the internet as a reference to be used when obtaining the accurate dataset in real life. Several field visits have been made to few centers in search of a dataset and some knowledge transfer sessions.

Field visit 01 to National Building Research Organization, Colombo 05

The National Building and Research Organization acted as an initiative knowledge pool during the research study, where, with the help of its Senior Scientist, Coordinator-Water Quality, Ms. Madara Dissanayake.

Figure 2.6: Laboratory of National Building Research Organization, Colombo

This first visit was a great opportunity to discuss in depth about the water quality issues that the industrial experts as well as the common people are facing and we had a very productive knowledge transfer sessions about the sensors and device designing.

The Summary as follows:

- Suitability of Drinking Water Compliance to the SLS 614:2013
- Parameters to consider when identifying suitability of water for construction according to British standards
- Past testing projects they have carried out
- Introduction to WQI and the closest WQI standards that Sri Lanka can follow i.e. Canadian Standards
- Manual WQI calculation method

A tour was also arranged to their laboratory where some water quality parameters are tested using chemicals and electrode methods. As shown in the above figure, a chemist will be using chemicals to test the conductivity of the water sample where the conductivity is analyzed and measured according to the refractive index of the used water sample upon the addition of color.

As this organization is conducting water quality testing only for research purposes and there is no and database maintained with continuous, historical data, obtaining dataset for the model training process became a great challenge.

Field visit 02 to Vavunathivu Water Treatment Plant, Batticaloa

The first visit to the water treatment plant located in Batticaloa was conducted 22nd of June 2020 through an appointment. The aim that we had during this visit is to obtain a suitable dataset and to know more about the sensors and devices that are used in water quality monitoring.

The authority had a lot to say about the issues we as Sri Lankans are facing due to the consumption and usage of unsafe water and the steps that are followed by the industrial experts to maintain a constant observation on the quality of the water that are being treated at the plant.

Figure 2.7: Turbidity meter

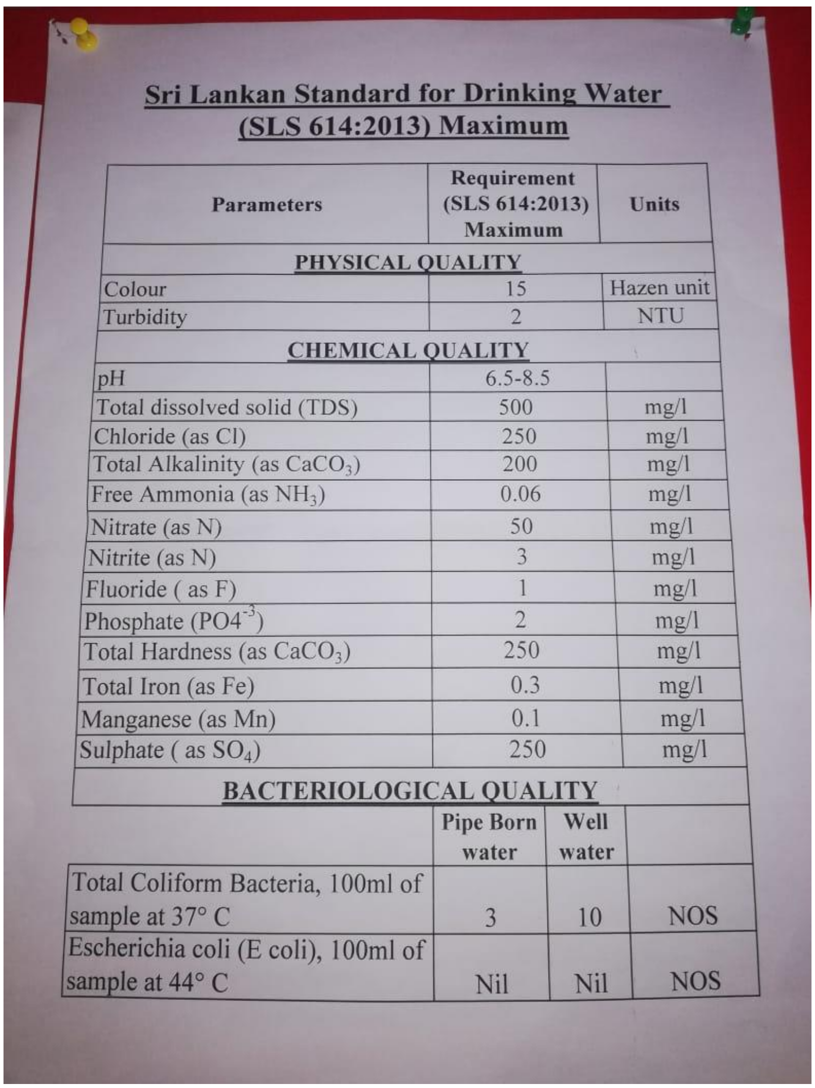
Figure 2.8: TDS meter

Figure 2.9: pH meter

Outcomes of this visit:

- The method the industrial authorities are following when determining the quality of water
- The water quality parameters that are being checked during the water treatment
- The accepted standard reference range of each parameters according to SLS standard
- Techniques followed during the water treatment
- Equipment used in industrial water quality identification and management

Although the treatment plant has some data regarding the water quality and the parameter readings, lack of continuous data for comparison is a drawback when it comes to the requirement of the research study.



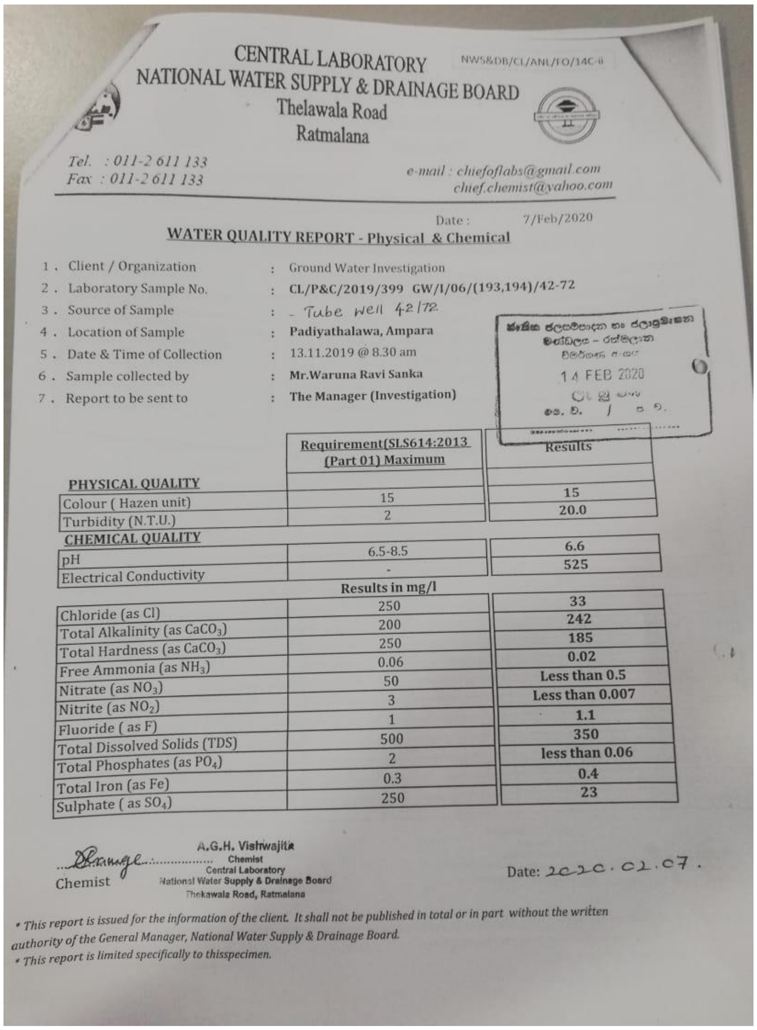
Sri Lankan Standard for Drinking Water (SLS 614:2013) Maximum

Parameters	Requirement (SLS 614:2013) Maximum		Units
PHYSICAL QUALITY			
Colour	15		Hazen unit
Turbidity	2		NTU
CHEMICAL QUALITY			
pH	6.5-8.5		
Total dissolved solid (TDS)	500		mg/l
Chloride (as Cl)	250		mg/l
Total Alkalinity (as $CaCO_3$)	200		mg/l
Free Ammonia (as NH_3)	0.06		mg/l
Nitrate (as N)	50		mg/l
Nitrite (as N)	3		mg/l
Fluoride (as F)	1		mg/l
Phosphate ($PO4^{-3}$)	2		mg/l
Total Hardness (as $CaCO_3$)	250		mg/l
Total Iron (as Fe)	0.3		mg/l
Manganese (as Mn)	0.1		mg/l
Sulphate (as SO_4)	250		mg/l
BACTERIOLOGICAL QUALITY			
	Pipe Born water	Well water	
Total Coliform Bacteria, 100ml of sample at 37° C	3	10	NOS
Escherichia coli (E coli), 100ml of sample at 44° C	Nil	Nil	NOS

Figure 2.10: SLS approved reference chart used by the industrial experts

Field visit 03 to National Water Supply & Drainage Board, Ratmalana

As the search for the dataset needed for model training continued, we paid a visit to the National Water Supply and Drainage Board head office located in Ratmalana. As NWSDB is the main source that supplies safe water to the public for consumption and domestic usage, these body had full information regarding the water sources, water quality analysis and management, and the supply of the safe water.



CENTRAL LABORATORY NWS&DB/CL/ANL/FO/14C-II
NATIONAL WATER SUPPLY & DRAINAGE BOARD
Thelawala Road
Ratmalana

Tel. : 011-2 611 133
Fax : 011-2 611 133

e-mail : chiefoflabs@gmail.com
chief.chemist@yahoo.com

Date : 7/Feb/2020

WATER QUALITY REPORT - Physical & Chemical

1. Client / Organization : Ground Water Investigation
2. Laboratory Sample No. : CL/P&C/2019/399 GW/I/06/(193,194)/42-72
3. Source of Sample : - Tube well 42/72
4. Location of Sample : Padiyathalawa, Ampara
5. Date & Time of Collection : 13.11.2019 @ 8.30 am
6. Sample collected by : Mr.Waruna Ravi Sanka
7. Report to be sent to : The Manager (Investigation)

14 FEB 2020

	Requirement(SLS614:2013 (Part 01) Maximum	Results
PHYSICAL QUALITY		
Colour (Hazen unit)	15	15
Turbidity (N.T.U.)	2	20.0
CHEMICAL QUALITY		
pH	6.5-8.5	6.6
Electrical Conductivity	-	525
	Results in mg/l	
Chloride (as Cl)	250	33
Total Alkalinity (as $CaCO_3$)	200	242
Total Hardness (as $CaCO_3$)	250	185
Free Ammonia (as NH_3)	0.06	0.02
Nitrate (as NO_3)	50	Less than 0.5
Nitrite (as NO_2)	3	Less than 0.007
Fluoride (as F)	1	1.1
Total Dissolved Solids (TDS)	500	350
Total Phosphates (as PO_4)	2	less than 0.06
Total Iron (as Fe)	0.3	0.4
Sulphate (as SO_4)	250	23

Chemist
A.G.H. Vishwajith
Chemist
Central Laboratory
National Water Supply & Drainage Board
Thelawala Road, Ratmalana

Date: 2020.02.07.

* This report is issued for the information of the client. It shall not be published in total or in part without the written authority of the General Manager, National Water Supply & Drainage Board.
* This report is limited specifically to thisspecimen.

Figure 2.11: Report issued by the NWSDB after testing water sample

After having an in-depth conversation about our research project, our objective and motto with Manager, Groundwater – Studies, Senior Hydrogeologist, Mr. W. G. T Indragith, he agreed to help us with our data need for the model training. Also, the manager was kind enough to handover informative articles, leaflets and pamphlets which had information regarding water quality management along with some water quality reports of certain sites of the country.

2.1.4 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 researches 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:

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

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.

2.1.5 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.

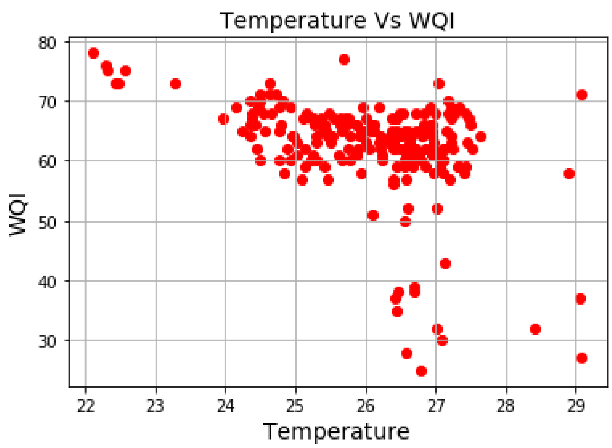


Figure 2.12: Temperature vs WQI graph result

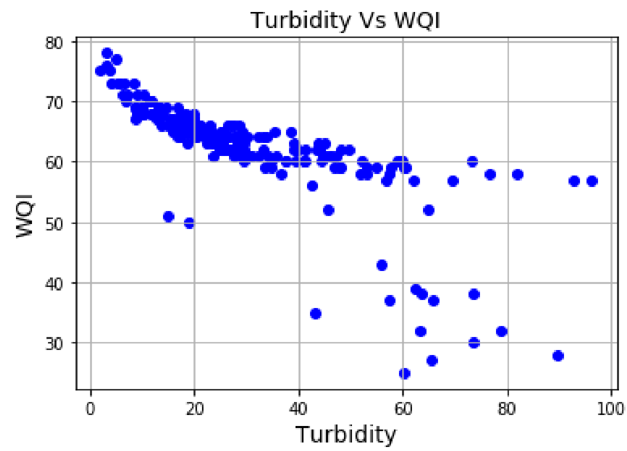


Figure 2.13: Turbidity vs WQI graph result

```
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()
```

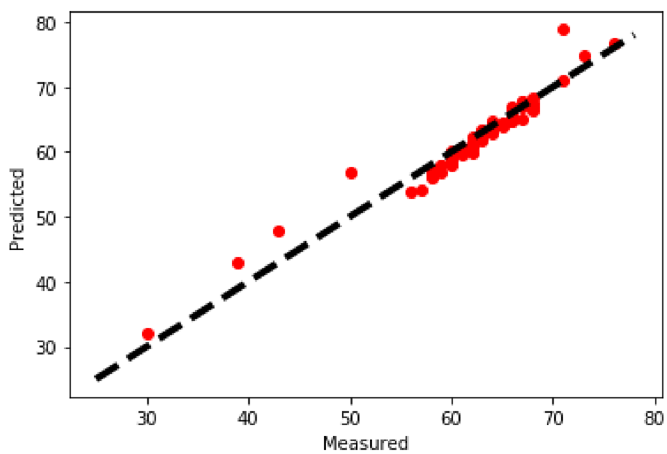


Figure 2.14: Linear regression prediction graph

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 2.15: Ridge regression train dataset evaluation

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 2.16: Lasso regression test data evaluation

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 2.17: Elastic Net regression test data evaluation

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 2.18: Random forest regression test, train data split

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 2.19: Random forest regression model training

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.

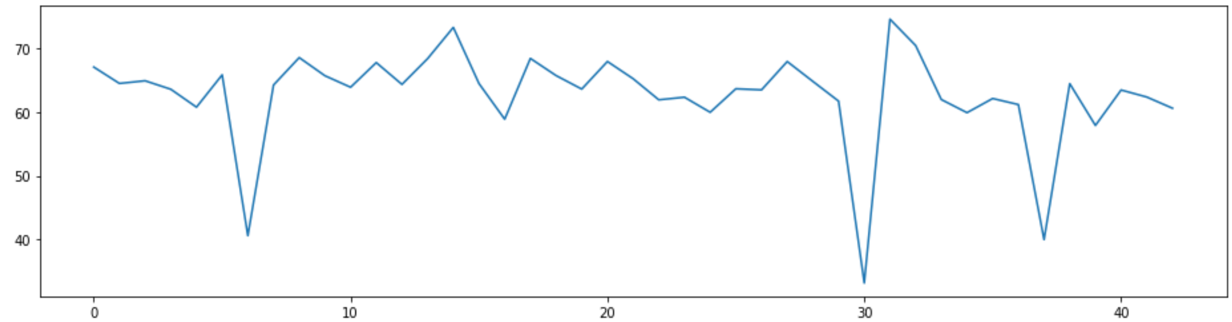


Figure 2.20: Random forest regression prediction result graph

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 2.21: K-nearest neighbors regression test, train data split

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 2.22: K-nearest neighbors regression model training

Artificial Neural Network

A Neural Network is a set of algorithms combined together to understand the underlaying relationships in a dataset through a process that represents how the neurons in human brain operates. The three layers: input, hidden and output layers helps to process a set of (multiple) inputs to predict a single output.

1. **Input Layer** - This layer is comprised of neurons which are designed to receive inputs from the external world. Learning and recognition precisely happens in this layer.
2. **Hidden Layer** – As the name suggests, this layer is hidden in between the input and output layers. It acts as a communication channel in directing the signals between the outer two layers where the signal is destined to be subjected to a maximum use.
3. **Output Layer** – Outer or output layer contain units that respond to the fed information it monitors and confirms whether it learned any tasks or not.

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 2.23: ANN regression model training

2.1.6 Machine Learning model implementation in backend

As per the observations from the above step, the final algorithm that was selected to be implemented in the system for the prediction of WQI is Random Forest Regressor. The model training process proceeds with a different Python IDE, PyCharm which could be used to connect the trained model to the backend.

PyCharm which uses Python 3.8.5 which is the latest version for the model training process. Once the pandas and numpy libraries were set, the training process can be started as a python script. Once the model is created, a path must be set to another folder where the .pkl file could be stored.

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

Figure 2.24: Pickle file storage path setting in PyCharm

Pickle is a commonly used library which is used to serialize and deserialize a python object structure. As the model should be serialized in order to be used in a mobile application, the model.pkl file should be stored in a place where it could be easily connected to the front-end application. Once the python script is completed and its run, the .pkl file will be created in the defined space where the developers can access.

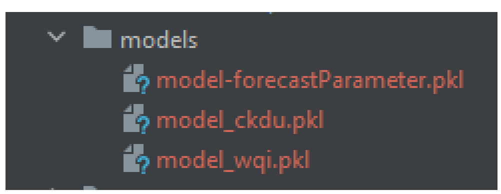


Figure 2.25: Generated pickle files in PyCharm

A separate python file must be created in order to define the input parameters that are used to pass into the model, importing the .pkl model and a route is defined in the same file where the API could be accessed. Once this file runs, the server will run in a pre-defined local host server with a port number of 5000.

```
@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'
        }
```

Figure 2.26: RFR model deployment and API call in PyCharm

This method explicitly creates an API which could be used to access data, in our scenario, get the data to view. Through Postman, a collaboration platform for API development, we can see the predicted value from the model through a GET method by passing all the 5 input parameters.

URL

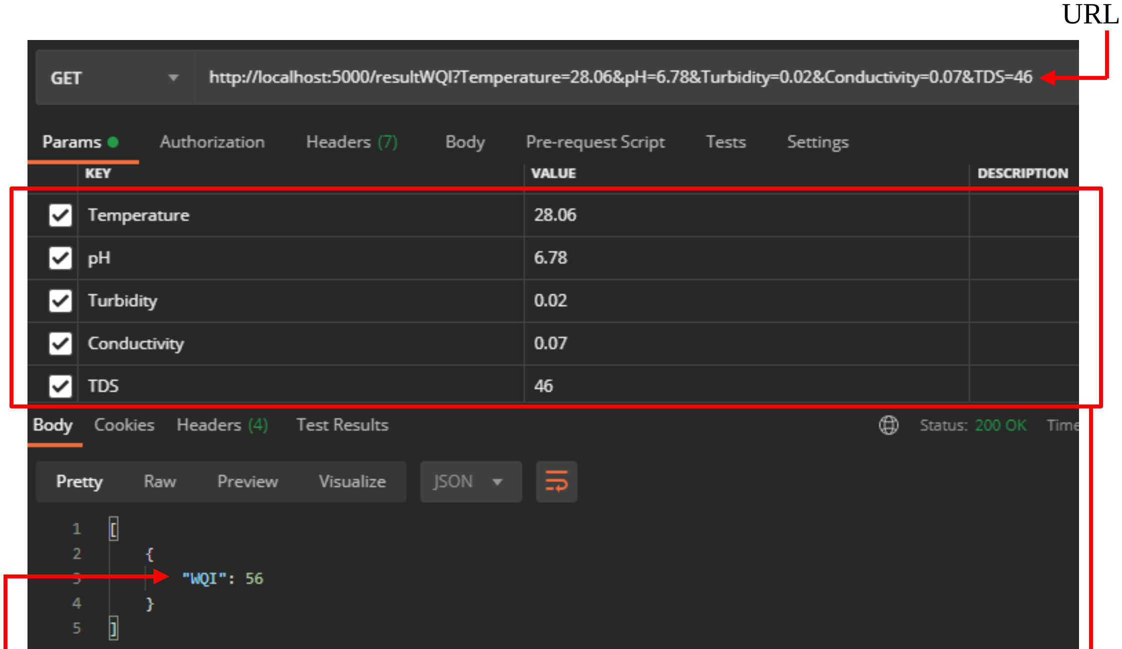


Figure 2.27: View prediction through Postman

Prediction output

Input parameter values

2.1.7 Mobile application development

Developing a front end to the system is vital as this is the interface through which the user communicates with the system. When developing UIs in Android Studio, the wireframes that we have designed earlier was referred and modified according to the user requirements from a user's perspective.

An initial landing page must be present where the functionalities are listed to the user from which the user can select the specific functionality that they need to access.

Figure 2.28: Home page of E-Tongue mobile application

When moving on to this research component, the parameter readings of the water sample which were stored in the database should be displayed to the user where the UI is designed in a way that the readings are displayed and an alert with red blinking area will show the parameter which have exceeded the standard range chart which is accepted by WHO.

Figure 2.29: Predicted medium quality water

Figure 2.30: Predicted bad quality water

Once clicked on the blinking space, the user will be notified about the acceptable range that the values should fall into.

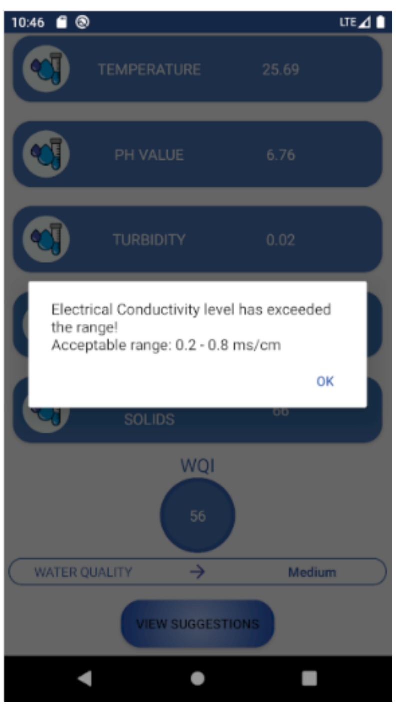


Figure 2.31: Conductivity level error

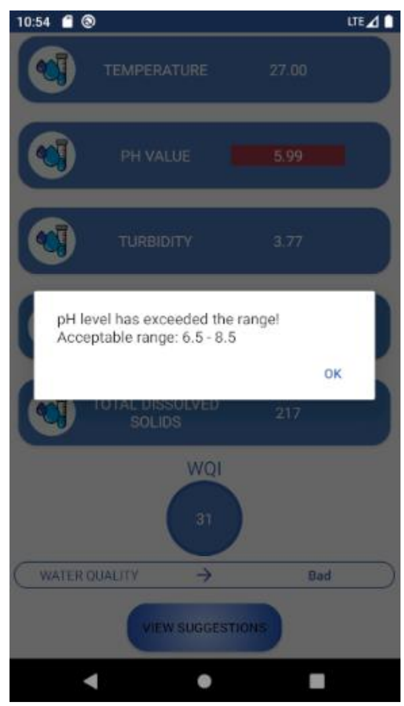


Figure 2.32: pH level error

Once the WQI is predicted, the designed UI must be able to display them. The water sample is then assigned into one of the five classes based on the predicted WQI value and the reference class mentioned below is the standard class range accepted by the WHO.

Range	**Class**
80 - 100	Excellent Quality
65 - 79	Good Quality
55 - 64	Medium Quality
30 - 54	Bad Quality
0 - 29	Very Bad Quality

Table 2.0: WQI reference chart

And based on the quality, the user must be redirected to a separate UI where suggestions are given on the purification method to be followed in order to make the water more usable, on a button click.

2.2 Commercialization

The target of this project as discussed multiple times above is to see people use it when they have doubts about the quality of water they are dealing with. As this E-Tongue device is small and in a manageable size, it is portable to the location where the water sources are located to test the water. the product is more flexible in a way where more sensors could be implemented when forming into a whole product for the industry.

During the field visits, the authorities let us know that there will be circumstances where a particular water source should be monitored in every 3 months or every 6 months in a year. Our product is suitable in such scenarios where the user only needs to test the water, the readings and predictions are displayed and stored where a report can be generated for further analysis.

The E-Tongue mobile application can be used by any android user to access the necessary component that the user wishes to access. The user has options to choose from water quality identification to CKDu outbreak and forecasting water quality parameters based on season. The user will be able to see suggestions about the purification methods that could be used industrially to ensure the safeness of water. these suggestions are an introduction to our country where the purification methods are much subtle and not extreme.

In a scenario where a user wants to dig a well or a tube well, if we analyze the real life scenario, the water resource board or any other relevant authority will visit the place, take sample from the place by injecting a small hole and test them in the laboratory to give a final report as whether the water from that source is drinkable and useable. Our product can be used to identify the quality of the water then and there on real time to make a final decision.

2.3 Implementation and Testing

2.3.1 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.

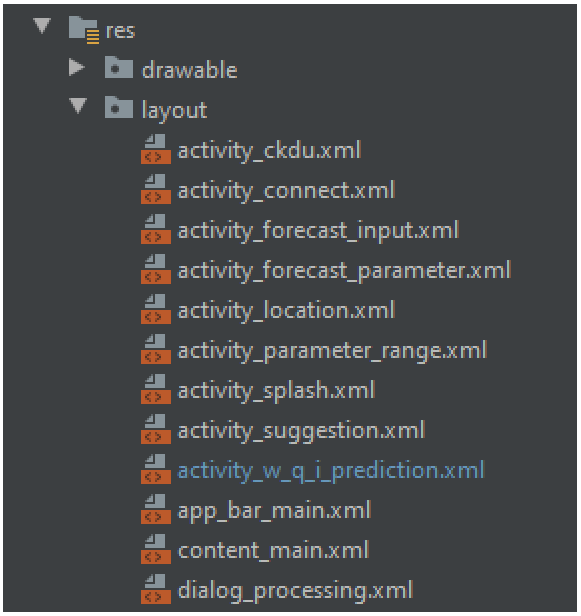


Figure 2.33: Layout folder in Android Studio

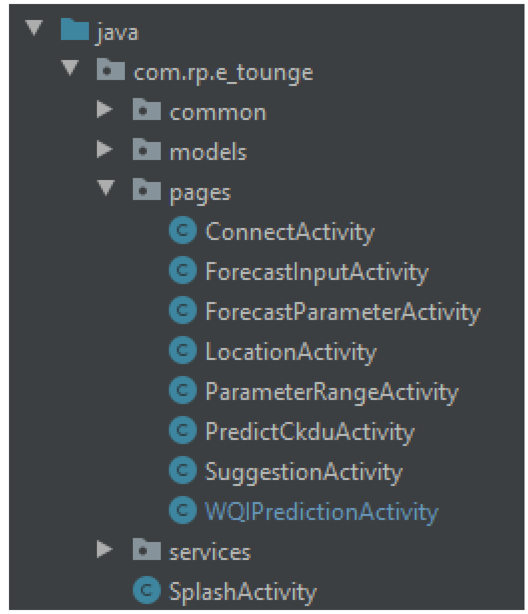


Figure 2.34: Activities in Android Studio

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.

```
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 2.35: API service calling class in Android Studio

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 2.36: Database access file in NodeJS

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 2.37: Readings receiving method call in Android Studio

2.3.2 Testing

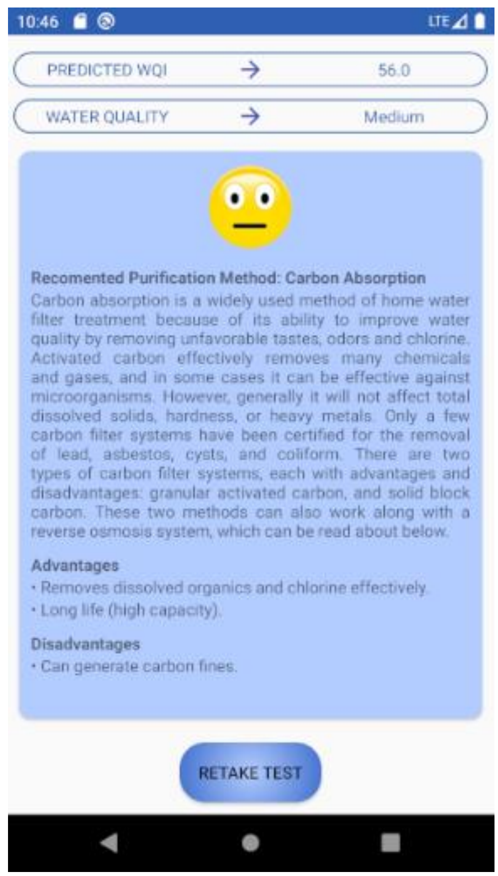
<u>Functionality Test</u>

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.

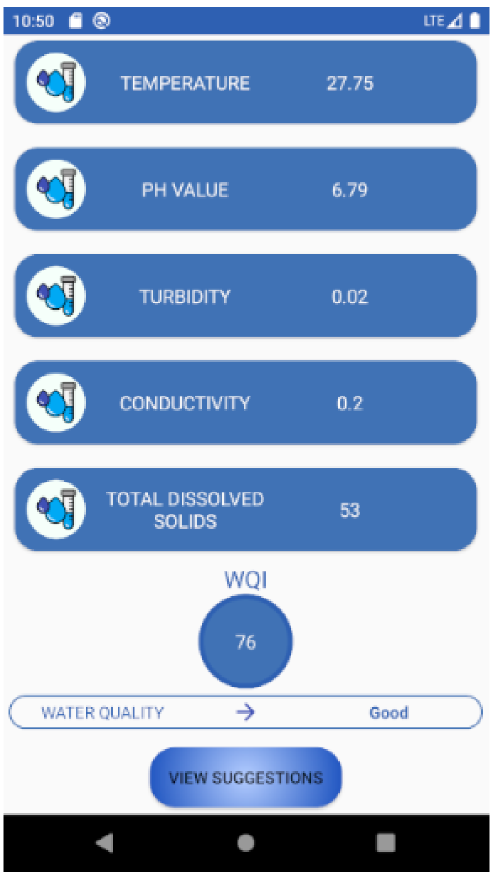
Status: Pass

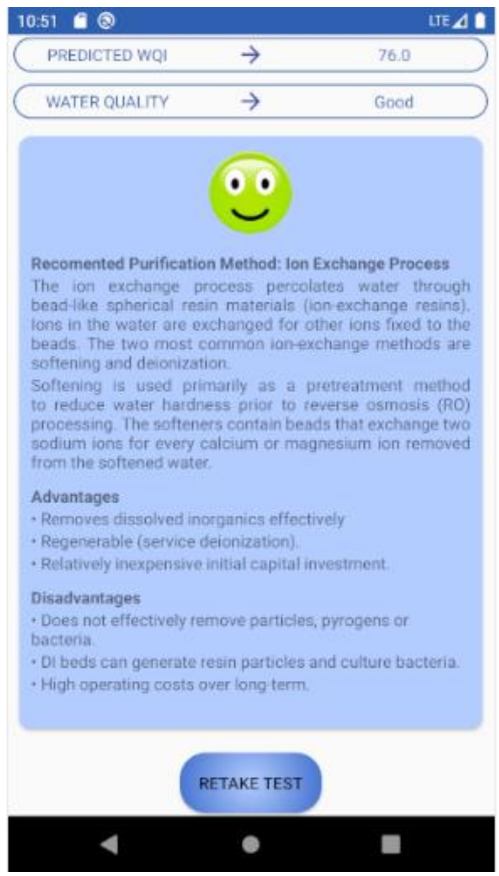
Test Case 02: Predicting the WQI of a tap water sample collected from the university premises.
Pre-condition: Model trained with data obtained from NWSDB
Input parameters:

- 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.

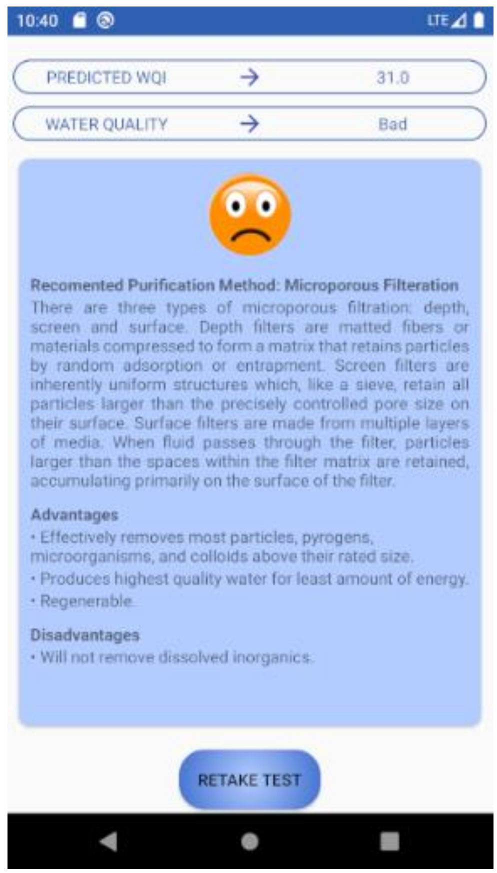
Status: Pass

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

UI Test: Sprint 1

Test Case Name	Description	Test Steps	Expected Results	Sprint 1 - July		
				Actual results	Status	Tested By
UI Load of WQI prediction page	Verifying the page landing.	1. Run the application 2. Observe the	The WQI prediction page should be loaded without any	The WQI prediction page lands without any delays.	Pass	Thenuja
Check Text Quality of WQI prediction page	Verifying the visibility of text, usage of colors, Font size, Font style etc of visible areas of the WQI prediction page.	1. Run the application 2. Observe the page landing.	The lables and texts should be visible, attractive and user friendly.	Some texts are not visible and the color selections are not attractive and user friendly.	Fail	Thenuja
Check the sensor readings are being displayed in the WQI prediction page	Verify whether the sensor readings are clearly displayed in the WQI prediction page	1. Run the application 2. Load the main page 3. click on the WQI	Readings must be retrieved and displayed in the UI	Readings are not getting retrieved and displayed successfully	Fail	Thenuja
Check the exceeded values are being highlighted and notified	Verify whether the sensor readings are analyzed and the readings which exceeds the data are getting highlighted	1. Run the application 2. Load the main page 3. click on the WQI prediction page	Readings must be retrieved and displayed in the UI	Readings are not compared and not highlighted	Fail	Thenuja
Check the predicted WQI is displayed in the page	Verify whether the predicted WQI value is displayed in the page	1. Run the application 2. Load the main page 3. click on the WQI	The predicted WQI must be retrieved from the backend and must get displayed	WQI is not displayed since the system is not connected with the backend	Fail	Thenuja

Figure 2.38: Manual UI testcases of sprint 1

UI Test: Sprint 2

Test Case Name	Description	Test Steps	Expected Results	Sprint 2 - September		
				Actual results	Status	Tested By
UI Load of WQI prediction page	Verifying the page landing.	1. Run the application 2. Observe the	The WQI prediction page should be loaded without any	The WQI prediction page lands without any delays.	Pass	Thenuja
Check Text Quality of WQI prediction page	Verifying the visibility of text, usage of colors, Font size, Font style etc of visible areas of the WQI prediction page.	1. Run the application 2. Observe the page landing.	The lables and texts should be visible, attractive and user friendly.	Some texts are visible and the color selections are not attractive and user friendly.	Pass	Thenuja
Check the sensor readings are being displayed in the WQI prediction page	Verify whether the sensor readings are clearly displayed in the WQI prediction page	1. Run the application 2. Load the main page 3. click on the WQI	Readings must be retrieved and displayed in the UI	Readings are getting retrieved and displayed successfully	Pass	Thenuja
Check the exceeded values are being highlighted and notified	Verify whether the sensor readings are analyzed and the readings which exceeds the data are getting highlighted	1. Run the application 2. Load the main page 3. click on the WQI prediction page	Readings must be retrieved and displayed in the UI	Readings are compared and highlighted	Pass	Thenuja
Check the predicted WQI is displayed in the page	Verify whether the predicted WQI value is displayed in the page	1. Run the application 2. Load the main page 3. click on the WQI	The predicted WQI must be retrieved from the backend and must get displayed	WQI is getting displayed after the backend connection	Pass	Thenuja

Figure 2.39: Manual UI testcases of sprint 2

Unit Test

Unit testing is a simplest form of testing process used by developers to ensure the fit of usage of smallest testable modules and their behavior. In Android Studio we have implemented unit testing where the Junit library is used. The testing snippet shown below ensures the launch of the WQIPredictionActivity without any failures.

```java
public class WQIPredictionActivityTest {

    @Rule
    public ActivityTestRule<WQIPredictionActivity> wqiPredictionActivityActivityTestRule =
            new ActivityTestRule<WQIPredictionActivity>(WQIPredictionActivity.class);

    private WActivity wActivity = null;

    @Before
    public void setUp() throws Exception {
        wActivity = wqiPredictionActivityActivityTestRule.getActivity();
    }

    @Test
    public void testLaunch() {
        View view = wActivity.findViewById(R.id.wqiValue);
        assertNotNull(view);
    }

    @After
    public void tearDown() throws Exception {
        wActivity = null;
    }
}
```

Figure 2.40: Unit testing in Android Studio

3 RESULTS AND DISCUSSION

3.1 Results

3.1.1 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.

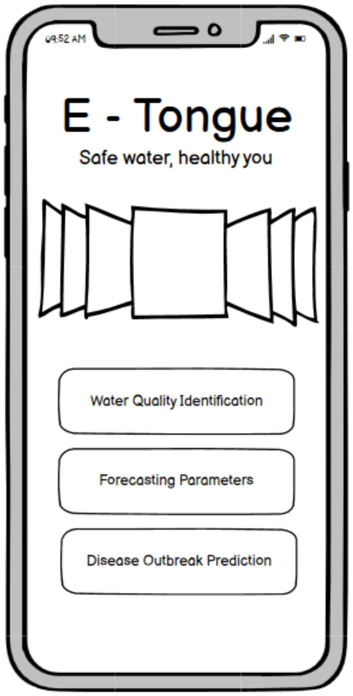
The selection of a best algorithm must be done in an impartial process. We have used the same dataset which we have obtained from the National Water Supply and Drainage Board, Sri Lanka. The platform that we have trained and the hardware components of the trained computer are maintained consistent for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

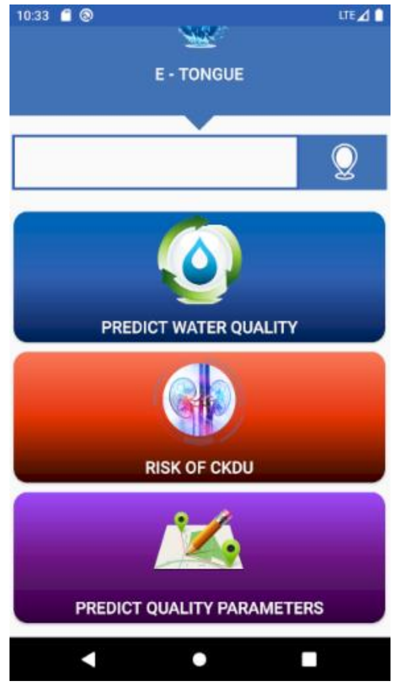
Algorithm Used	MAE	MSE	RMSE	R-SQUARED
Linear Regression	1.3082	3.6730	1.9165	0.9249
k Nearest Neighbors	1.9203	17.2976	4.1590	0.6424
Random Forest Regression	0.6673	1.6477	1.2836	0.9782
Artificial Neural Network	7.5060	90.2375	9.4993	0.2381
Ridge Regression	1.3084	3.6738	1.9167	0.9249
Lasso Regression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

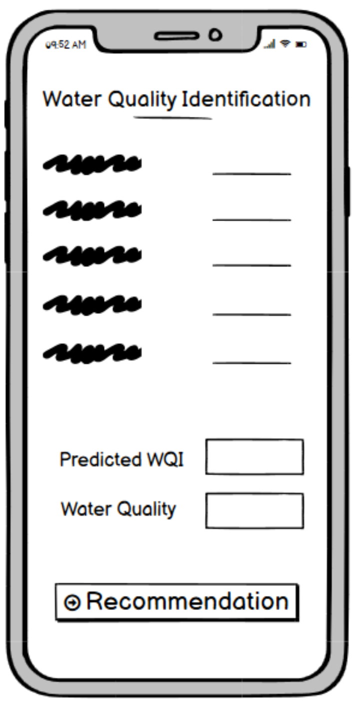
Table 3.0: Algorithm performance evaluation

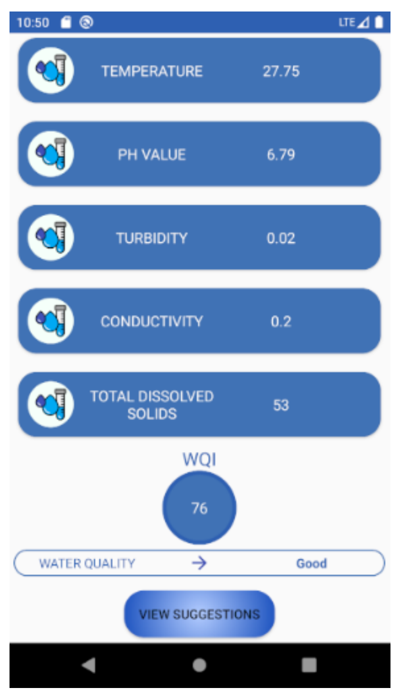
In a regression problem, the R-Squared value defines the coefficient of determination. Based on the observed results, the Random Forest Regression model has highest R^2 and lowest Root Mean Square Error. As a result of the model selection process, RFR was selected as the model to be implemented into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

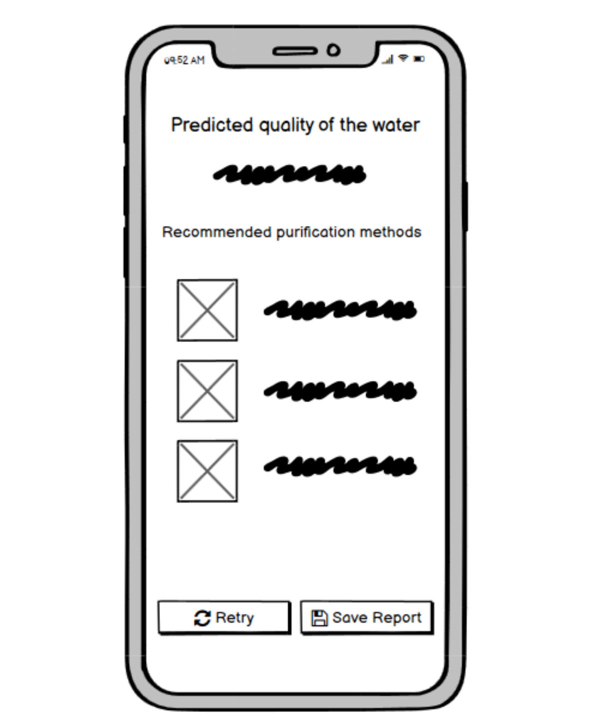
3.1.2 Implementing the UIs











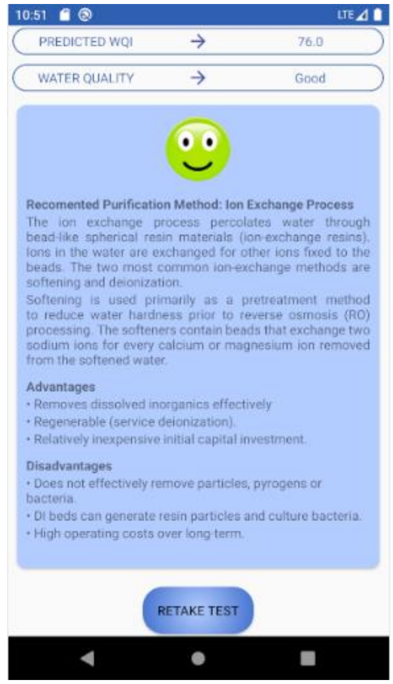


Table 3.1: Comparison of wireframes with the implemented UIs

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.

3.2 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.

By considering the results from the conducted literature survey, the main 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.

With the help of the industrial experts, specially from the National Water Board and Drainage System, suitable methods of purifying water have been introduced which were used in the mobile application as a suggestive offering to the users based on the predicted water quality. The fashioned device totally operates in a cloud-based environment where the readings from the device are sent to the cloud where the machine learning models were stored and the mobile application is designed to access data and information from the cloud database that ensures performance, availability and security.

Clear advancements could be seen from the implementation of this procedure in terms of accuracy, in addition to obtaining results in stage regression aligned to the official water quality identification methods followed by the National Water Board and Drainage System and the water resource Authority.

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.

It is important that the introduction of this term must have an impact on the authorities as well as the public to give them a clear picture of the actual water they are dealing with. To fill this gap, this smart system is designed and implemented in order to predict the WQI using a set of sensors where the users can use in multiple scenarios that requires a water quality identification in real time.

This specific component requires regression algorithms to train models. With the abundance of regression algorithms in the industry at time, it is vital to select one specific algorithm which can function flexible to the scenario. Training models requires data and the most difficulty that was faced during this study period was to find an appropriate dataset that could be used to train the models for comparison. As an initial step, a testing dataset which had fair similarities to the required dataset was obtained and several models were trained. Random Forest Regression algorithm was selected to proceed with the system implementation as it is highly reliable on a regression type scenario and it follows an ensemble learning pattern. Once the model is selected, the National Water Supply and Drainage Board agreed to help the study by providing the required data. After conducting multiple surveys and paying few visits to the authorities which conducts studies and researches on water bodies, the final model retraining process started with preparing and modifying the obtained dataset.

The hardship that we faced at this stage was the calculation of WQI as this index was never used in any part of the system where the authorities define the quality of the water sample they test on. The manual WQI calculation required a total time period of 3 days as this process involves complex mathematical equations and graph references. Although the data preparation stage consumed a considerable amount of time, the outputs that were obtained from the model after retraining the RFR model was not in an acceptable range. The WQI calculated through the model after connecting to the device had a difference range of ±9 which didn't show the accuracy that we required. After a deep discussion and analysis that we found out that the reason is due to the calibration issue that was present. After recalibrating the device and manually testing different water samples obtained from Malabe, Kegalle and Jaffna areas, the difference range was brought down to ±2. Once the front-end application was developed, we were able to connect the E-Tongue application and run tests entirely as users to identify water quality of different ground water samples.

3.4 Summary of the student contribution

Member	Component	Tasks
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.

Table 3.2: Summary of the student contribution

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 particular 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] Umair Ahmed, Rafia Mumtaz , Hirra Anwar , Asad A. Shah, Rabia Irfan and José GarcíaNieto, Efficient Water Quality Prediction Using Supervised Machine Learning, Water 2019, 11, 2210; doi:10.3390/w11112210.

[2] Sarath Gunatilake, Channa Jayasumana, Glyphosate, Hard Water and Nephrotoxic Metals: Are They the Culprits Behind the Epidemic of Chronic Kidney Disease of Unknown Etiology in Sri Lanka?, International Journal of Environmental Research and Public Health 11(2):2125-47

[3] Waterborne Diseases by Dr Manjula Rathnabharathie cover story 02, Volume 29, Number 03 Vidurava. Available: http://dl.nsf.ac.lk/bitstream/handle/1/9962/VIDU_29%283%29_19.pdf?sequence=2&isAllowed= y

[4] M.L.Fowsul Ameer, Water-Borne Diseases and Their Challenges in the Coastal of Ampara District in Sri Lanka. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjereqntdnrAhXEfH0KHeIhABUQFjACegQIBRAB&url=https%3A%2F%2Fwww.researchgate.net%2Fpublication%2F314239558_Water-Borne_Diseases_and_the_Their_Challenges_in_the_Coastal_of_Ampara_District_in_Sri_Lanka&usg=AOvVaw28Ye_KISuA8UrAiy9qguJ4

[5] Sri Lankan Journal of Infectious Diseases - Contaminated Public well water as a source of sporadic outbreak of enteric infection in Northern Sri Lanka.

[6] Senaka Rajapakse, Mitrakrishnan Chrishan Shivanthan, and Mathu Selvarajah, Chronic kidney disease of unknown etiology in Sri Lanka, Int J Occup Environ Health. 2016 Jul; 22(3): 259–264.

[7] Jayatilake N, Mendis S, Maheepala P, Mehta FR, Chronic kidney disease of uncertain aetiology: prevalence and causative factors in a developing country. CKDu National Research Project Team.

[8] Peiris-John RJ, Wanigasuriya JK, Wickremasinghe AR, Dissanayake WP, Hittarage A, Exposure to acetylcholinesterase-inhibiting pesticides and chronic renal failure.

[9] Linkoping University, Department of Physics, Chemistry and Biology (IFM), Available: http://www.ifm.liu.se/applphys/biosensors-and-bioelectro/group-members/fredrik-winquist/

[10] Abhishruti Bhuyan, Bipan Tudu, Rajib Bandyopadhyay, Arunangshu Ghosh, and Sanjeev Kumar, ARMAX Modeling and Impedance Analysis of Voltammetric E-Tongue for Evaluation of Infused Tea, IEEE SENSORS JOURNAL, VOL. 19, NO. 11, JUNE 1, 2019.

[11] Palash K. Kundu, Madhusree Kundu, A comparative study on tea samples classification by E-Tongue using Cross correlation based PCA , PCA and Shamon's Nonlinear mapping method, 2012 7th International Conference on Electrical and Computer Engineering.

[12] Palash K. Kundu, Madhusree Kundu, Classification of Tea Samples Using SVM as Machine Learning Component of E-Tongue, 2016 International Conference on Intelligent Control Power and Instrumentation (ICICPI).

[13] Shafi, U.; Mumtaz, R.; Anwar, H.; Qamar, A.M.; Khurshid, H. Surface Water Pollution Detection using Internet of Things. In Proceedings of the 2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT), Islamabad, Pakistan, 8–10 October 2018; pp. 92–96.

[14] Sakizadeh, M. Artificial intelligence for the prediction of water quality index in groundwater systems. Model. Earth Syst. Environ. 2016, 2, 8.

[15] Nabeel M. Gazzaz, Mohd Kamil Yosoff, Ahmad Zaharin Aris, Hafizan Juhair, Mohammad Firzul Ramil - Artificial neural network modeling of the water quality index for Kinta River (Malaysia) using water quality variables as predictors, Marine Pollution Bulletin 64 (2012) 2409–2420.

[16] Umair Ahmed, Rafia Mumtaz , Hirra Anwar , Asad A. Shah, Rabia Irfan and José GarcíaNieto, Efficient Water Quality Prediction Using Supervised Machine Learning, Water 2019, 11, 2210; doi:10.3390/w11112210.

[17] Tests with laboratory details, National Water Supply & Drainage Board. Available: https://drive.google.com/file/d/1D7IGpSsKe1Hs8OTwuFWr0n5PgGB7yOwC/view

[18] Handbook for water consumers – Ministry of water Supply & Drainage, national water Supply and Drainage Board

[19] Shweta Tyagi, Bhavtosh Sharma, Prashant Singh, Rajendra Dobhal, Water Quality Assessment in Terms of Water Quality Index. Available: http://pubs.sciepub.com/ajwr/1/3/3/

[20] Monitoring the Quality of Surface Waters (WQI Calculator), Available: https://www.water-research.net/index.php/water-treatment/water-monitoring/monitoring-the-quality-of-surfacewaters

[21] "Android Studio", En.wikipedia.org, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Android_Studio. [Accessed: 18- September- 2020].

6 APPENDICES

Appendix A – source code (RFR model training in Python)

```
# 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)
```

```python
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))
```

Appendix B – source code (java class: WQIPredictionActivity)

```
package com.rp.e_tounge.pages;

import androidx.appcompat.app.AppCompatActivity;

import android.animation.ArgbEvaluator;
import android.animation.ObjectAnimator;
import android.annotation.SuppressLint;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.util.Log;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.Animation;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.PopupWindow;
import android.widget.TextClock;
import android.widget.TextView;

import com.google.gson.GsonBuilder;
import com.rp.e_tounge.R;
import com.rp.e_tounge.common.Utils;
import com.rp.e_tounge.models.PredictWaterParameters;
import com.rp.e_tounge.models.SensorData;
import com.rp.e_tounge.models.SensorDataResponse;
import com.rp.e_tounge.models.WQI;
import com.rp.e_tounge.services.ApiClient;
import com.rp.e_tounge.services.ApiServices;

import java.util.List;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class WQIPredictionActivity extends AppCompatActivity {

    float temperature, ph, turbidity, conductivity, tds;
    TextView Wqitextview;
    TextView qualityText;
    TextView phText, tempText, turbText, condText, tdsText;
    ApiServices apiServices;
    PopupWindow popupWindow;
    LayoutInflater layoutInflater;
    LinearLayout linearLayout;
    Context context;
```

```
    double wqi;
    Retrofit retrofitClient;
    public static final String EXTRA_TEXT = "androidx.appcompat.app.WQI.EXTRA_TEXT";
    public static final String EXTRA_NUMBER =
"androidx.appcompat.app.WQI.EXTRA_NUMBER";
    //public String T;
    private static SensorData sensorData;
    TextView textViewValue;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_w_q_i_prediction);

        Wqitextview = (TextView) findViewById(R.id.wqiValue);
        float wqi = Float.parseFloat(Wqitextview.getText().toString());
        context = WQIPredictionActivity.this;
        textViewValue = (TextView) findViewById(R.id.qValue);


        Button button = findViewById(R.id.suggestionButton);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openSuggestionActivity();
            }
        });

        //getWQIPrediction();

        phText = (TextView) findViewById(R.id.phId);
        tempText = (TextView) findViewById(R.id.tempId);
        turbText = (TextView) findViewById(R.id.tbId);
        condText = (TextView) findViewById(R.id.ecId);
        tdsText = (TextView) findViewById(R.id.tdsId);


        float temperatureB = Float.parseFloat(tempText.getText().toString());
        float phB = Float.parseFloat(phText.getText().toString());
        float turbidityB = Float.parseFloat(turbText.getText().toString());
        float conductivityB = Float.parseFloat(condText.getText().toString());
        float tdsB = Float.parseFloat(tdsText.getText().toString());


        if (turbidityB <= 0 || turbidityB > 5) {
            blinkTurbidity();
            turbText.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Utils.showMessageDialog("Turbidity level has exceeded the range!
\nAcceptable range: 0.1 - 4.0 ntu", context);
                }
            });
        }
```

```
        if (temperatureB < 20 || temperatureB > 30){
            blinkTemperature();
            tempText.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Utils.showMessageDialog("Temperature level has exceeded the
range! \nAcceptable range: 27 - 32 Celsius", context);
                }
            });
        }

        if (phB < 6.5 || phB > 8.5) {
            blinkPH();
            phText.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Utils.showMessageDialog("pH level has exceeded the range!
\nAcceptable range: 6.5 - 8.5", context);
                }
            });
        }

        if (conductivityB <= 0.2 || conductivityB > 0.8) {
            blinkConductivity();
            condText.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Utils.showMessageDialog("Electrical Conductivity level has
exceeded the range! \nAcceptable range: 0.2 - 0.8 ms/cm", context);
                }
            });
        }

        if (tdsB < 0 || tdsB > 500) {
            blinkTDS();
            tdsText.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Utils.showMessageDialog("Total Dissolved Solids (TDS) level has
exceeded the range! \nAcceptable range: 128 - 512 ppm", context);
                }
            });
        }


        //pop up window to view the WQI reference range
        linearLayout = (LinearLayout) findViewById(R.id.wqiLayout);
        Wqitextview.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                layoutInflater = (LayoutInflater)
getApplicationContext().getSystemService(LAYOUT_INFLATER_SERVICE);
                ViewGroup container = (ViewGroup)
layoutInflater.inflate(R.layout.popup_menu_wqi, null);
```

```
                popupWindow = new PopupWindow(container, 900, 900, true);
                popupWindow.showAtLocation(linearLayout, Gravity.TOP, 0, 500);

                container.setOnTouchListener(new View.OnTouchListener() {
                    @Override
                    public boolean onTouch(View view, MotionEvent motionEvent) {
                        popupWindow.dismiss();
                        return true;
                    }
                });
            }
        });

        getSensorData();

    }

    public void openSuggestionActivity() {
        //textView 2 line code was here
        Wqitextview = (TextView) findViewById(R.id.wqiValue);
        float wqi = Float.parseFloat(Wqitextview.getText().toString());

        qualityText = (TextView) findViewById(R.id.qValue);
        String result = qualityText.getText().toString();

        Intent intent = new Intent(this, SuggestionActivity.class);
        intent.putExtra(EXTRA_NUMBER, wqi);
        intent.putExtra(EXTRA_TEXT, result);
        startActivity(intent);
    }

    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("Direction", "Response Request " + response.raw().request());
                Log.d("Body", "Response body " + response.body());
                if (!response.isSuccessful()) {
                    Utils.showMessageDialog("Something Went Wrong,Please try again
later", context);
                    Log.d("Code", "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();
            }

            @Override
            public void onFailure(Call<List<WQI>> call, Throwable t) {
                Utils.hideProgressDialog();
                Utils.showMessageDialog("Something Went Wrong,Please try again 
later", context);
                System.out.println(t.getMessage());
            }

        });
    }

    //methods to blink and highlight the text view if it exceeds the range
    @SuppressLint("WrongConstant")
    public void blinkTemperature() {
        ObjectAnimator animator = ObjectAnimator.ofInt(tempText, "backgroundColor", 
Color.RED);
        animator.setDuration(500);
        animator.setEvaluator(new ArgbEvaluator());
        animator.setRepeatMode(Animation.REVERSE);
        animator.setRepeatCount(Animation.INFINITE);
        animator.start();

    }

    @SuppressLint("WrongConstant")
    public void blinkPH() {
        ObjectAnimator animator = ObjectAnimator.ofInt(phText, "backgroundColor", 
Color.RED);
        animator.setDuration(500);
        animator.setEvaluator(new ArgbEvaluator());
        animator.setRepeatMode(Animation.REVERSE);
        animator.setRepeatCount(Animation.INFINITE);
        animator.start();

    }

    @SuppressLint("WrongConstant")
    public void blinkTurbidity() {
        ObjectAnimator animator = ObjectAnimator.ofInt(turbText, "backgroundColor", 
Color.RED);
        animator.setDuration(500);
        animator.setEvaluator(new ArgbEvaluator());
        animator.setRepeatMode(Animation.REVERSE);
        animator.setRepeatCount(Animation.INFINITE);
        animator.start();

    }

    @SuppressLint("WrongConstant")
    public void blinkConductivity() {
        ObjectAnimator animator = ObjectAnimator.ofInt(condText, "backgroundColor", 
Color.RED);
```

```java
        animator.setDuration(500);
        animator.setEvaluator(new ArgbEvaluator());
        animator.setRepeatMode(Animation.REVERSE);
        animator.setRepeatCount(Animation.INFINITE);
        animator.start();

    }

    @SuppressLint("WrongConstant")
    public void blinkTDS() {
        ObjectAnimator animator = ObjectAnimator.ofInt(tdsText, "backgroundColor",
Color.RED);
        animator.setDuration(500);
        animator.setEvaluator(new ArgbEvaluator());
        animator.setRepeatMode(Animation.REVERSE);
        animator.setRepeatCount(Animation.INFINITE);
        animator.start();

    }

    void getSensorData() {
        try {
            Utils.showProgressDialog(context);
            retrofitClient = ApiClient.getClientForNode();
            apiServices = retrofitClient.create(ApiServices.class);

            Call<SensorDataResponse> call = apiServices.getSensorData();
            call.enqueue(new Callback<SensorDataResponse>() {
                @Override
                public void onResponse(Call<SensorDataResponse> call,
Response<SensorDataResponse> response) {
                    Utils.hideProgressDialog();
                    if (!response.isSuccessful()) {
                        Utils.showMessageDialog("Something Went Wrong,Please try
again later", context);
                    } else {
                        sensorData = response.body().items.get(0);
                        temperature = Float.parseFloat(sensorData.temp);
                        tds = Float.parseFloat(sensorData.tds);
                        ph = Float.parseFloat(sensorData.ph);
                        turbidity = Float.parseFloat(sensorData.turb);
                        conductivity = Float.parseFloat(sensorData.ec);

                        phText.setText(Float.toString(ph));
                        tempText.setText(Float.toString(temperature));
                        turbText.setText(Float.toString(turbidity));
                        condText.setText(Float.toString(conductivity));
                        tdsText.setText(Float.toString(tds));

                        getWQIPrediction();

                        Log.w("2.0 getFeed > Full json res wrapped in pretty printed
gson  => ", new GsonBuilder().setPrettyPrinting().create().toJson(response.body()));
                    }
                }
```

```java
                @Override
                public void onFailure(Call<SensorDataResponse> call, Throwable t) {
                    Utils.hideProgressDialog();
                    Utils.showMessageDialog("Something Went Wrong with retrieving
sensor data,Please try again later", context);
                }
            });

        } catch (Exception e) {
            Utils.hideProgressDialog();
            e.printStackTrace();
            Utils.showMessageDialog("Something Went Wrong,Please try again later",
context);
        }

    }

    void setData() {
        if (wqi >= 80 && wqi < 100) {
            textViewValue.setText("Excellent !");
            textViewValue.setTextColor(Color.parseColor("#008000"));
        } else if (wqi >= 65 && wqi < 80) {
            textViewValue.setText("Good");
            textViewValue.setTextColor(Color.parseColor("#ace600"));
        } else if (wqi >= 55 && wqi < 65) {
            textViewValue.setText("Medium");
            textViewValue.setTextColor(Color.parseColor("#e6b800"));
        } else if (wqi >= 30 && wqi < 55) {
            textViewValue.setText("Bad");
            textViewValue.setTextColor(Color.parseColor("#e65c00"));
        } else if (wqi >= 0 && wqi < 30) {
            textViewValue.setText("Very Bad");
            textViewValue.setTextColor(Color.parseColor("#cc0000"));
        }
    }
}
```