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

Final (Draft) Report

S. Thenuja
IT16170780

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

Department of Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

September 2020

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

S. Thenuja
IT16170780

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

Department of Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

September 2020

DECLARATION

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

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

Name	**Registration Number**	**Signature**
S. Thenuja	IT16170780	

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

……………………………………..　　　　　　　　……………………..

Signature of the Supervisor:　　　　　　　　Date:

ABSTRACT

Groundwater quality is a comprehensive complication in all over the world. This proposed E-tongue is a smart intelligent tool to predict the safe consumption of groundwater. Forecasting groundwater quality parameters in the future based on the user location or gathered sample's location and which concern the seasonal variation of Sri Lanka which is one of the main features of this system. Moreover, people, who are in dry zone areas are struggling to consume safe groundwater for drinking. This proposed system mainly concentrates on dry zone areas and select groundwater sources for identifying water quality parameters for training and testing. In addition, this proposed model able to predict the groundwater quality parameters effectively and meet the requirements of consumers. It would be a favorable assistant to enhance decision making for a future sustainable quality of drinking water.

Keywords: - *water quality parameters, E-tongue, groundwater*

ACKNOWLEDGEMENT

I would like to take this opportunity to express my deep sense of gratitude and profound feeling of admiration to our project supervisor Ms. K.A.D.C.P.Kahandawaarchci. I would also like to extend our heartfelt gratitude to Ms. N.D.U.Gamage, the co-supervisor of the project, for sharing the experience with the project matters. Last but not least, we would like to gratefully acknowledge our Lecture-in-charge, Head of Research Groups/Projects Dr. Janaka Wijekoon for his guidance taught the course.

Thank you.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

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

1. INTRODUCTION

1.1 Background Study & Literature Survey

1.1.1 Background Study

Since groundwater demand has been rapidly increased for the past few decades all over Sri Lanka particularly in rural areas. Groundwater is considered as one of the main water sources for drinking. Due to usage of groundwater being continuously increased for major sources of rural water supplies primarily for Industrial, household, and drinking purposes. Figure 1 Drinking water coverage shows drinking water coverage in Sri Lanka. There are 94% improved drinking water access and the remaining 6% is unserved. Besides, 80% of rural domestic water supplies depend on groundwater which is from dug wells and tube wells.

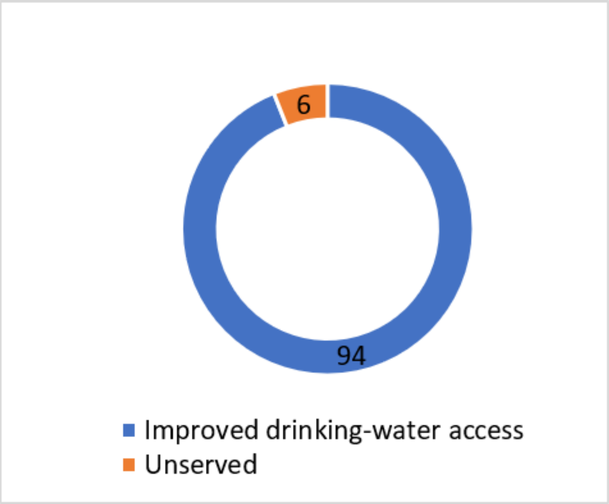


Figure 1 Drinking water coverage

Climate experienced in Sri Lanka for twelve months period is categorized into four seasonal variations as follow

- First Inter-monsoon Season (March - April)
- Southwest -monsoon Season (May - September)
- Second Inter-monsoon Season (October-November)
- Northeast -monsoon Season (December – February)

Groundwater consumption of dry zone areas is raised rapidly as well as it will be affected by seasonal variation [6]. Sri Lanka is mainly divided into three climatic zones such as wet zone, dry zone, and intermediate zone. Figure 2 Climatic zones in Sri Lanka shows the department of geography in Sri Lanka is divided into climatic zones [2]. Hence, there is around 75% comes under dry zone areas also the absence of rivers in that area therefore consumers depend on groundwater for drinking and agriculture.

This research has been conducted in the dry zone area specifically in the Jaffna district with 200 participants, 57.5% population has not contained adequate knowledge on the transmission and prevention of water-borne diseases, as a result, they are drinking uncertainty quality of water [6]. Consequently, 87% used unsafe water for drinking and domestic purposes. Up to 88% of water bone diseases arisen from unsafe water supplies and inadequate sanitation and hygiene [1].

World Health Organization (WHO) has its own water quality index to check the water quality in all around the world. The forecasting water quality index of the located place is more advantageous

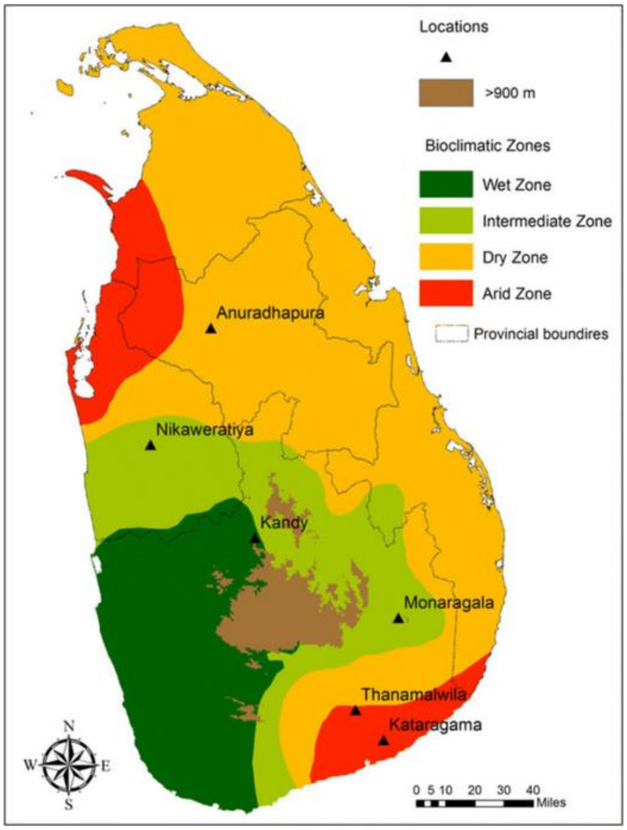


Figure 2 Climatic zones in Sri Lanka

to create awareness among the population. Besides, identifying the quality of water sources based on seasonal changes are most salient to prevent seasonal water diseases.

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

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

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

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

According to the study of the report which is issued in by National Water Supply & Drainage board in Sri Lanka, there are nearly 35% people can get the cleaned and controlled mineral content piped line water. The rest of the people generally get the from the tube wells or dug wells for their drinking and cultivation purpose. The concentration of the water quality parameters should be important before supplying the water to areas.

Hence, we are planning to develop a smart intelligence device to predict water quality in real-time as well as forecasting water quality parameters in the future. Accordingly, every people will be able to check the quality of water sources related to their area based on seasonal variations. The proposed system not only able to specify that the best quality of water sources that can be acquired from the nearby location in the future will be visualized on a heat map but also alert the people when the water quality index in low. Thus, it will be a necessary application for most of the people who are suffering from bad quality water.

1.1.2 Literature Survey

There are various kinds of researches have been conducted to gather requirements for predicting water quality parameters and identifying water quality sources based on seasonal variations and also to improve water quality forecasting in the future, therefore several interesting research papers, articles, and journals have been found out related to this domain even though extremely different features are identified while comparing with the proposed system. Several significant details are gathered via these papers.

1.1.2.1 Water quality prediction method was based on the LSTM Neural Network

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

1.1.2.2 An ANN application for water quality forecasting a single parameter

Shie-Yui Liong et al… were conducted research to predict and forecast a single quantitative characteristic of water bodies by using Artificial Neural Networks (ANN) [4]. It was utilized to forecast the dissolved oxygen level in the water. The ANN model can target both linear and non-linear relationships and then these relationships can be directly learned from the model which was being built using the data. The study in this domain particularly focuses on the Singapore Coastal water sources [6].

1.1.2.3 A Time series analysis of surface water quality

AbdollahTaheri Tizro1 et al… in 2014, had conducted another research in this domain [7]. Here, predicting the quality of river water by using the time series model and forecasting water quality for a particular river, which is surface water quality parameter, they have been mainly focused on water pollution and identifying water quality parameters that are going to be affected when the water was polluted. Although the result of this research to generate a model that includes water pollution factors and water quality parameters to prevent water pollution. Even though it will not be suitable for groundwater and that does not consider seasonal variations.

1.1.2.4 A hybrid neural network and ARIMA model for water quality time-series prediction

Durudu Omer Faruk et al… [8], This research depicts the time series of innovative models for water resources management that can be built using both linear and non-linear patterns [6]. Certainly, Non-linear relationships cannot be dealt with the ARIMA model and both linear and non-linear patterns cannot be handled by the neural network model. This approach proposes a hybrid ARIMA and neural network model along with a conjugated training algorithm [6]. Hence the model can be able to produce accurate predictions for the time-series data to the water quality predictions.

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

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

1.1.2.6 Time Series Forecasting of Water Quality of River Godavari

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

1.1.2.7 Forecasting of River Water Quality Parameters

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

1.1.2.8 Prediction of water quality parameters of Karoon River (Iran) by artificial intelligence-based models

S. Emamgholizadeh et al… [12] explore to predict the water quality parameters of Karoon River (Iran) by artificial intelligence-based models, in this research the machine learning model was created by using 17 years (1995 to 2011) past dataset to predict future values of dissolved oxygen (DO), biochemical oxygen demand (BOD) and chemical oxygen demand (COD) levels in the water has been demonstrated by using nine parameters such as PH, Ca, Mg, EC, Na, Turbidity, PO4, NO3 and NO2 [12].

Furthermore, another research conducted in time series analysis of water characteristics of streams in Eastern Macedonia – Thrace, Greece, this research has been done to predict maximum water temperature in the future [13]. The temperature of the water affects the maximum dissolved oxygen density in the water and speed of the chemical and biological reactions. Therefore, the machine learning model was developed by using the datasets of April 2013 – July 2016 to predict the temperature of the water in the future. The forecasting values of the water temperature will be helped scientists and responsible parties to figure out proper solutions about river pollution in the earliest stage and that will save the lives of many living things which depend on the water.

1.2 Research Gap

Numerous researches have been done in this domain and some products are currently available in the software market. Further, all the applications have certain limitations and it was focused on water quality prediction in real-time as well as specifically, concern about predicting water pollution factors. The main intention of this proposed system is to focus on forecasting water quality parameters in the future and alert the people via the mobile application if the water quality parameters are not fallen in the specific range for future. Lack of smart devices to predict the water quality and the forecasting the water quality parameters based on the given location have been created the research.

Before starting to implement the features of this system, the necessity of deep analysis of relevant systems or products that were in the market is required. Implementing a new application that has the same features, will be considered as a waste of money and also time-consuming. By analyzing the applications which are already available at the market, it is the most valuable to cut down the workload.

Table 1 shows the comparation of the other relevant product in the same domain with E- tongue. That is a kind a way to get a quick idea about the research gap and product features.

Features	Existing Products / Research				
	MeraBhujal [11]	Water Quality 4Thai [12]	Time series analysis of surface water quality [10]	A new forecasting model for groundwater quality based on short time series monitoring data [13]	E-Tongue
Location wise prediction	✓	✓	✓	✓	✓
Seasonal wise prediction	✗	✗	✓	✗	✓
Forecasting water quality parameter in future	✗	✗	✗	✗	✓
Visualization	✓	✓	✗	✗	✓
Predict consumable nearby water sources in the future	✗	✗	✗	✗	✓
Alert the consumer when water quality parameter having the danger values	✗	✓	✗	✗	✓

Table 1 Comparing with other applications

1.3 Research problem

Water quality prediction has more significance not only for the management of water sources but also for the prevention of spreading diseases such as diarrhea, cholera, Kidney Failure, and Typhoid [1]. People are advised to drink quality or purified water. Most of the people who are lived in the dry zone area, depending on the groundwater sources for drinking and also, they have the least number of water resources to fulfill their needs. Also, the seasonal variation impacts the quality of water [5]. Besides, the result of water quality deterioration is intensified water scarcity in the future. Hence, people are unable to forecast the water quality at an early stage.

A smart intelligence device that has the capability to predict water quality is proposed to develop a final year research project. It will be utilized to predict the water quality for the future and to visualize the water quality sources based on the seasonal changes on the heat map. If forecasting the water quality parameters of the groundwater are in the dry zone area, then it will favor for people who are suffering from bad quality of water based on seasonal variation. Through this system, the consumer will be alerted when the water goes under a low water quality index. Therefore, they will be able to take better precautions for purifying water.

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

- How are people forecasting water quality?
- How to identify the quality of water sources based on seasonal variations?
- How to predict the safe consumption of groundwater sources in the future?
- How to get overall groundwater quality in the dry zone area?

To overcome the above-mentioned problems, data are gathered from the Irrigation department and water supply department in dry zone areas. Then, machining learning algorithms have been developed to predict water quality parameters and to identify the quality of water sources based on seasonal variation and anticipating the quality of water in the future. Through this research, the awareness has been created among people who are affected by seasonal variation.

1.4 Research Objectives

1.4.1 Main Objectives

E-Tongue (A smart tool to predict the safe consumption of groundwater) is facilitated by forecasting water quality parameters in the future and predicting the quality of water sources on specific locations based on seasonal variations. This system provides an additional facility that is to identify the quality of water in real-time. Hence, it will grant the results with more accuracy and sustainable water quality parameters. The main purpose of this research is to concern about predicting water quality parameters.

1.4.2 Sub Objectives

In order to accomplish the primary goal of this proposed system, the consecutive sub-objectives need to be obtained.

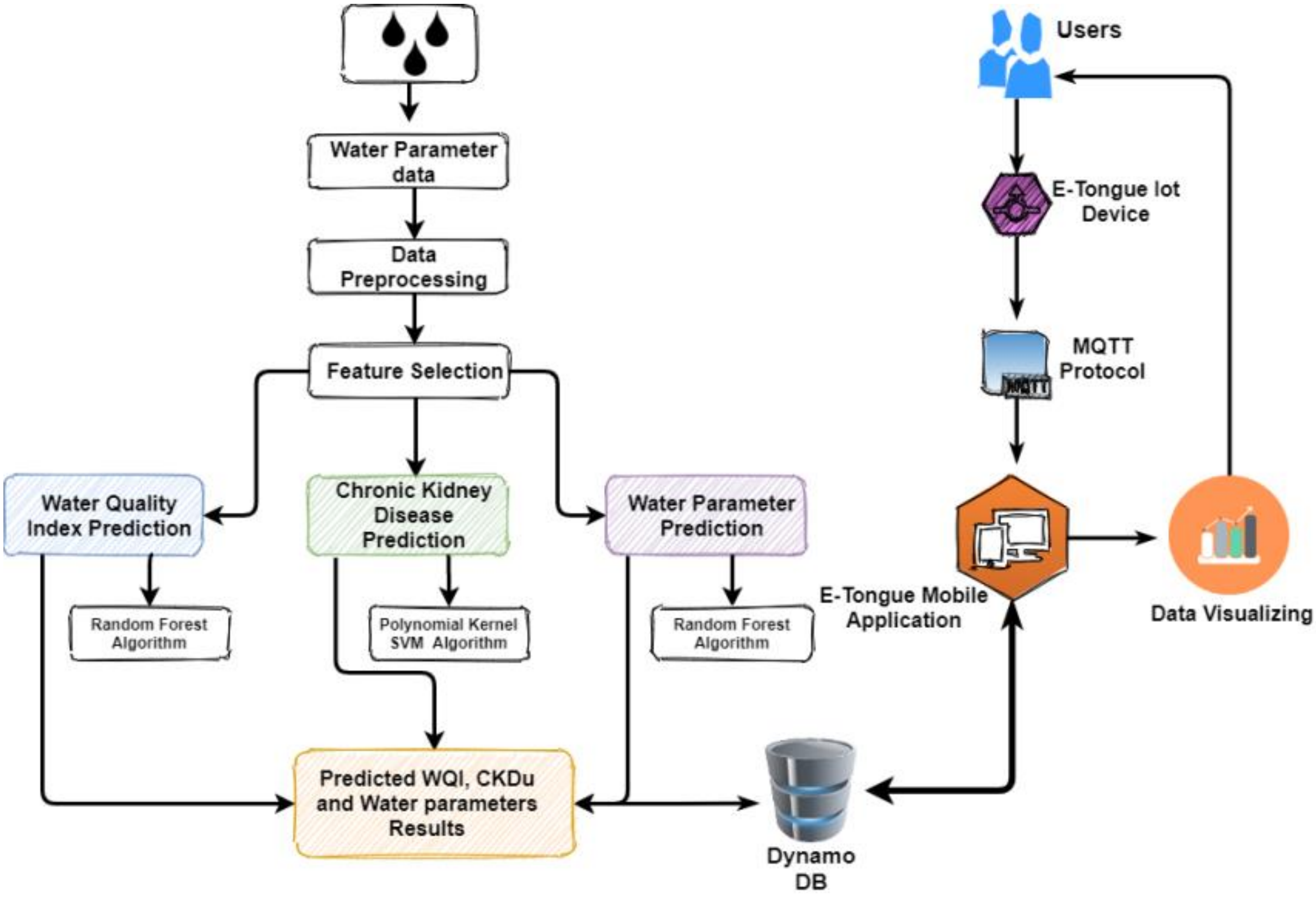
- Collect the specific set of requirements in order to continue the project plan.
- Gather data from the National Water Supply and Drainage Board in Sri Lanka.
- Exploration of the data set for data cleaning and understanding of the data.
- Extract features of the set of water quality parameters for forecasting water quality.
- Training the machine learning model by using different kinds of algorithms, then select the best suitable algorithm among them with high accuracy.
- Predict the water quality parameter for five years with high accuracy.
- Identifying water quality parameter which will be affected by seasonal changes.
- Predict the nearby quality of water sources in a specific area based on seasonal variation.
- Test the model by using the test data set.
- Implement graphical user interfaces and system logic based on the system specifications.
- Track user location and predict the water quality parameter of the certain location
- Alert the consumers when the water quality parameters are having bad water quality index values.
- Validate and test the implemented application whether the application has been completed the user requirements or not in order to achieve customer satisfaction.

2. METHODOLOGY

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

2.1 System Overview

It has many momentous research areas such as Machine Learning, Artificial Intelligence, IoT devices (Sensors), and Visualizing technologies. Research has conducted more studies on the above research areas. Hence, the gathered information will be used to achieve the main objectives and sub-objectives. Figure 3 is shown the system architecture diagram of the proposed system.



Water Parameter data
Data Preprocessing
Feature Selection
Water Quality Index Prediction
Chronic Kidney Disease Prediction
Water Parameter Prediction
Random Forest Algorithm
Polynomial Kernel SVM Algorithm
Random Forest Algorithm
Predicted WQI, CKDu and Water parameters Results
Dynamo DB
Users
E-Tongue Iot Device
MQTT Protocol
MQTT
E-Tongue Mobile Application
Data Visualizing

Figure 3 System overview diagram

2.1.1 Component Overview

According to the conclusion from the literature review, technology selection and software solutions are the most important part of this research. Forecasting water quality parameters in the future is one of the components in this proposed system. Consumers are able to know the water quality parameter before. Further, they able to identify the nearby quality of water sources based on seasonal variation. Generate the heat map to visualize the nearby quality of water sources. Figure 4 shows the system overview diagram of the component.

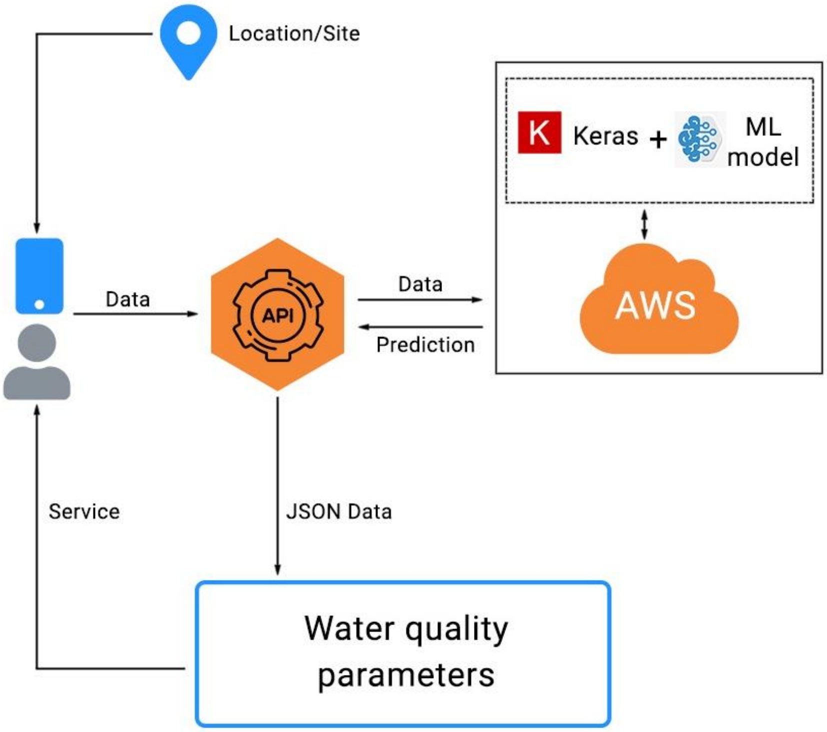


Figure 4 Component overview diagram

The result of this component will be forecasted water quality parameter value for the future. When the user locates their location or searches the known location, the application will be shown the water quality parameter for the future. Here, location is the main input for getting the output of this feature. This input sends to a model which tries to forecast water quality parameter by using historical water quality data through a Machine Learning algorithm. The trained ML model will be stored in the AWS. Figure 5 illustrates the flow diagram of the component.

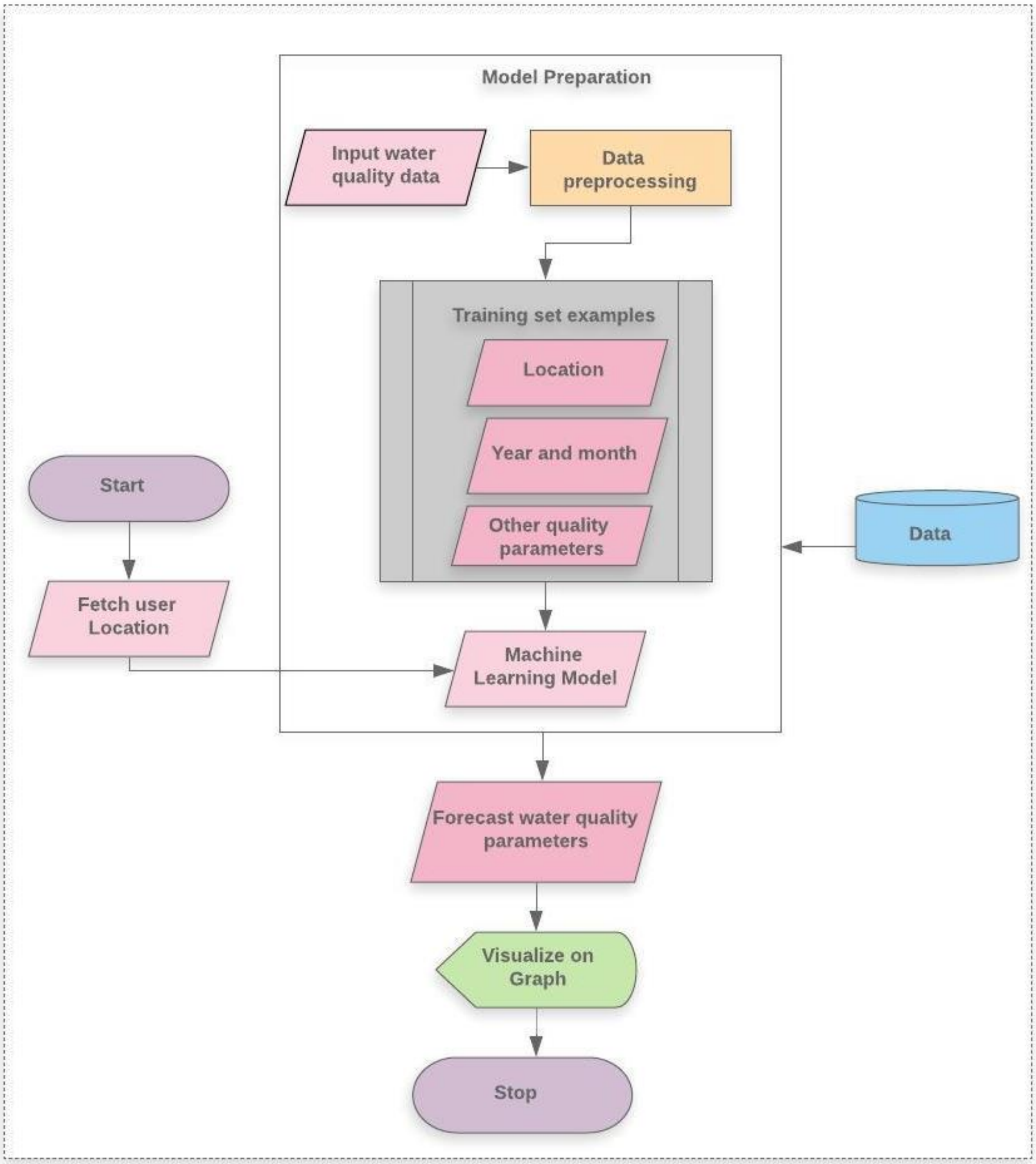


Figure 5 Flow chart of the component

According to the Figure 5 should possess below functionalities that need to be achieved to build the ML model.

- Preprocessing of the data
- Feature Engineering
- Hyper parameter tuning
- Ensemble methods

Use case Diagram

It is a diagram that is the simplest way to represent the user interaction of the system which illustrates the relationship between the system and the user with a different types of use cases. Below Figure 6 shows the user interaction of the component which is forecasting the water quality parameters.

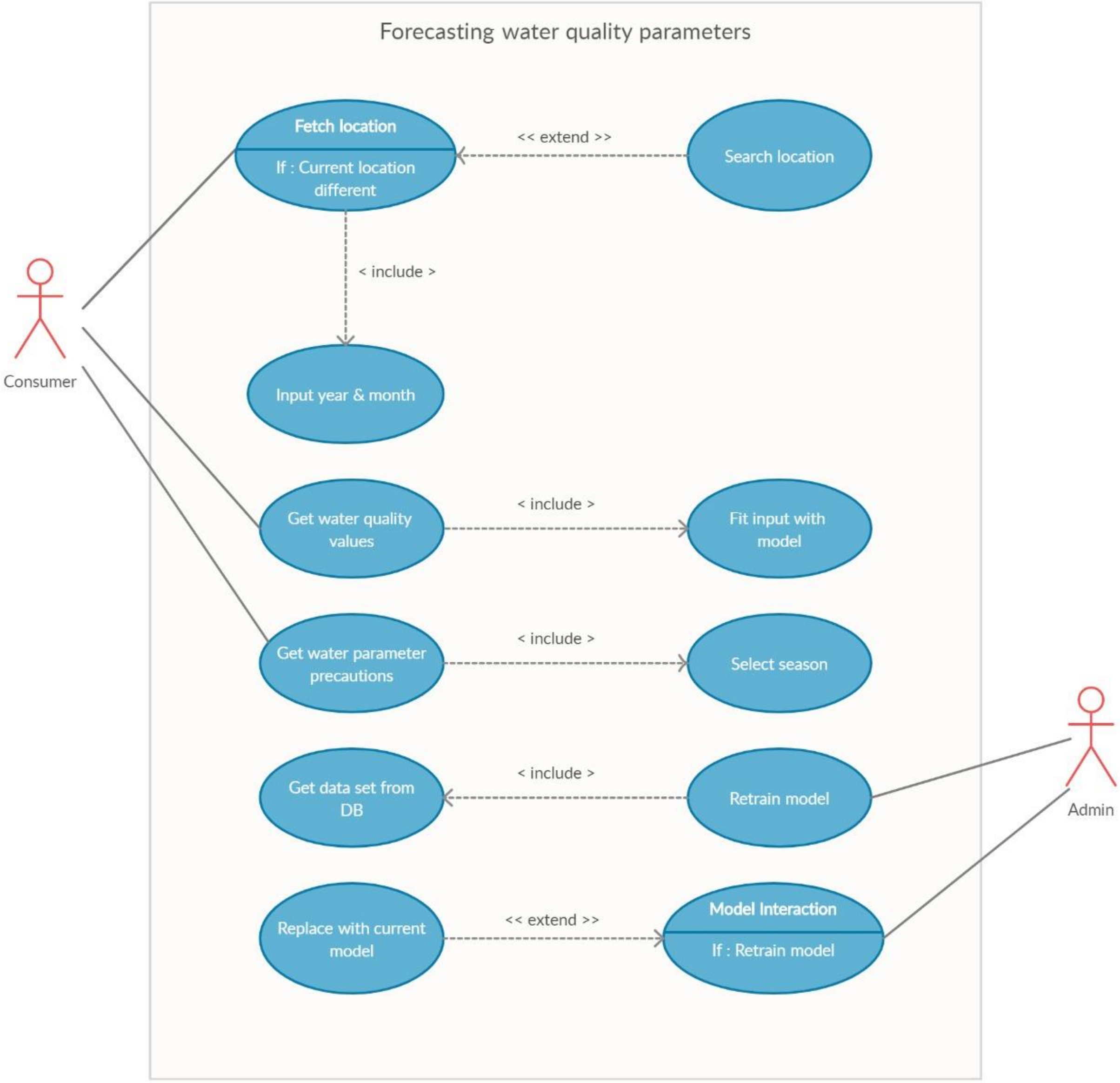


Figure 6 Use case diagram of forecasting water quality parameters

2.2 Development Process

Software Development Life Cycle (SDLC) is a process that is used by the software industry to develop any kind of software. So, we need to follow certain processes to achieve the main goal of our research product. According to this section we have to outline the problems which are induced to do the research on that area, define the functionalities that are going to solve the problem that was raised in the initial stage, explore the outcome that we have expected for the solution, and maintaining proper time management that was helped to us for one year of period to conduct the research. The "E- Tongue" project is developed under this life cycle model.

According to the Figure 7 Iterative Development Model shown model, we have to separate the requirements into a functional basis. During the iteration process, each development modules need to go through the requirements that were separated, design, and implementation and testing.

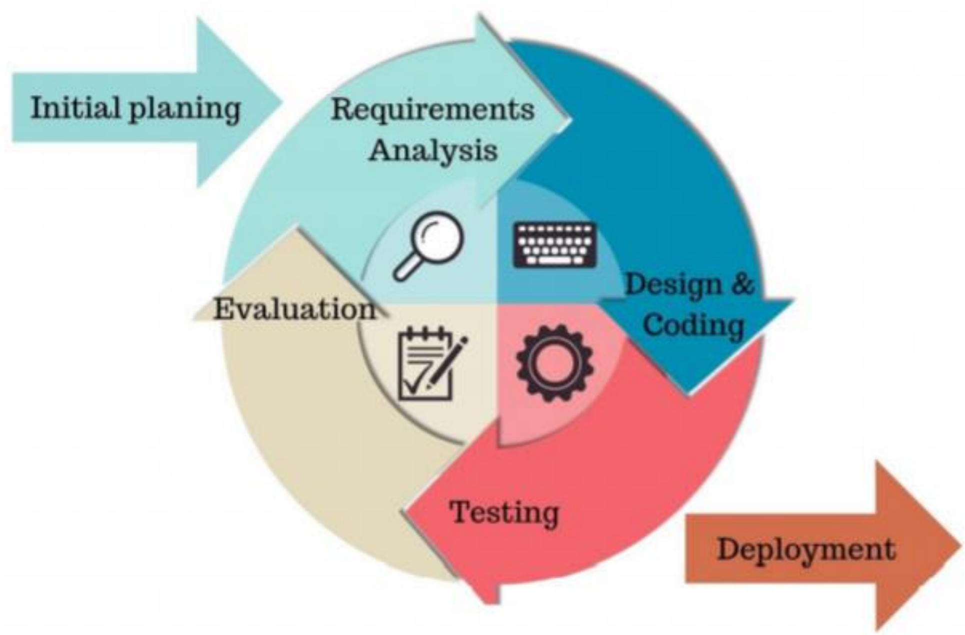


Figure 7 Iterative Development Model

2.3 Requirement Gathering and Data Collection

There are numerous research papers focused on the water quality prediction domain, After getting further analysis about that how to influence machine learning to come up with a solution and what type of system we are going to do that gets higher rates in the market. By analyzing and getting a better understanding of the solutions that is much suitable for Sri Lanka, especially in groundwater of dry zone areas. Those are the spotlights that we are going to implement our system.

2.3.1 Requirements

Functional Requirements:

- Provide a lightweight device to predict the safe consumption of the groundwater in the dry zone area.
- Forecasting the water quality parameters based on the location and noted with seasonal changes.
- Tracking the user location while using the system.

Nonfunctional Requirements:

- It takes low memory storage and less battery power consumption while using this application.
- Users should provide the location and other input parameters.

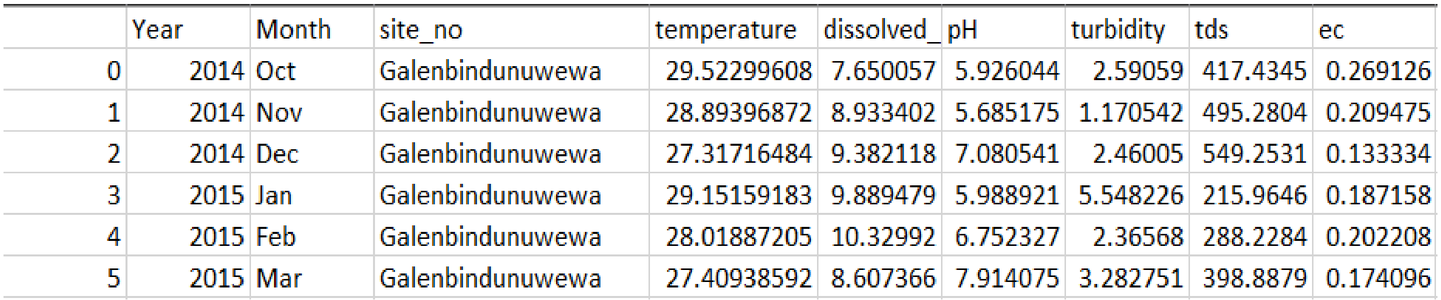
Site adaption requirement:

English is the essential language to operate the system at the initial release. The system should be updated with local languages Sinhala and Tamil in a future release. Device must be connected to the internet because all the backend ML model and DynamoDB were deployed into AWS also every component communicate via APIs, External Google API.

2.3.2 Data collection

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

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



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

Figure 8 First five rows of the dataset

2.4 Feasibility Study

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

2.4.1 Software Boundaries

- AndroidStudio

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

- Pycharm

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

- Google API

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

- DynamoDB

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

- Amazon Web Services (AWS)

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

- GitLab

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

2.4.2 Hardware Boundaries

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

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

2.4.3 Communication Boundaries

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

Memory constraints

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

For Back-end:

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

GPU - NVIDIA GeForce GTX 1050 or more.

Storage – 3GB or more.

For Front-end:

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

Storage – 1GB or more.

Operations

User able to perform below operations

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

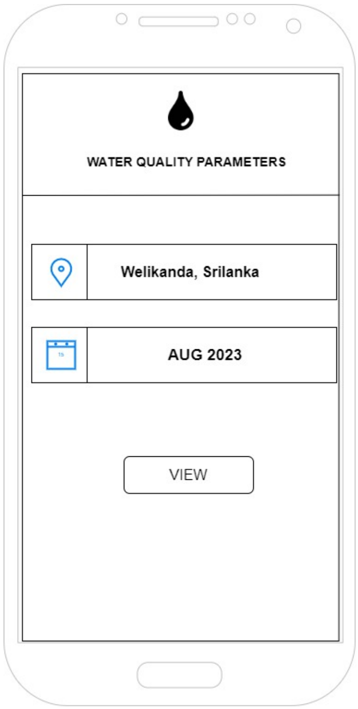
2.5 Designs

Forecasting water quality parameters is one of the components in the system which is designed to get the water quality parameter in advance.

- The following figure 11 shows the skeleton of the interface to fetch the user location or user able to search their desired location and pin the marker to get the location of the water source.
- Another following figure 9 illustrates the skeleton of the input data UI for forecasting water quality parameters.
- As a result of the following figure 12 show the skeleton of the output UI once the prediction did with input data, all the results are shown in a graph model.

Below figure 9,10, 11and 12 show the wire frames of the UI that has to implement in the implantation phase

Wireframes


WATER QUALITY PARAMETERS

Welikanda, Srilanka

AUG 2023

Figure 12 Wireframe of input parameter

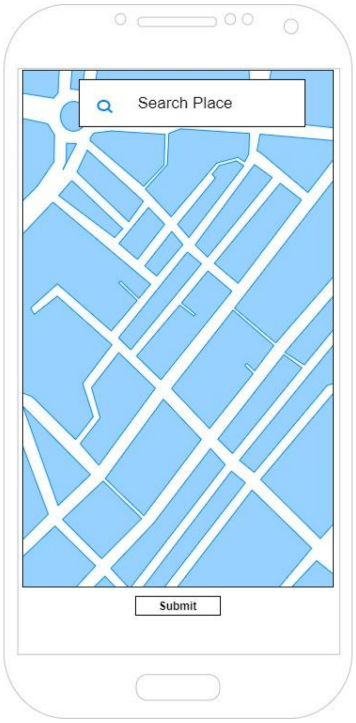
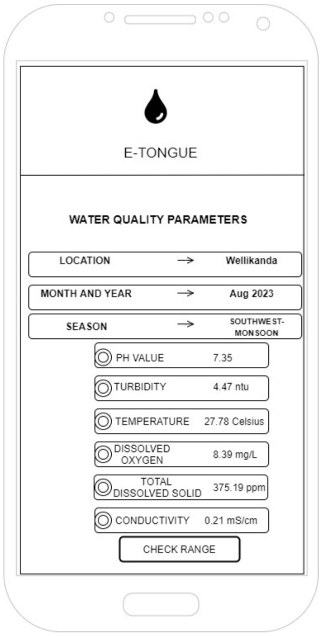


Figure 11 Wireframe of Search location


E-TONGUE

WATER QUALITY PARAMETERS

LOCATION → Wellikanda

MONTH AND YEAR → Aug 2023

SEASON → SOUTHWEST-MONSOON

○ PH VALUE 7.35

○ TURBIDITY 4.47 ntu

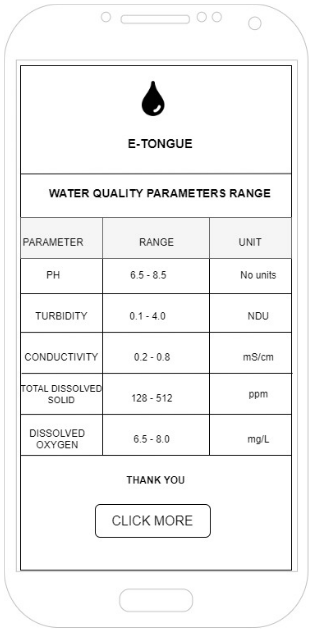
○ TEMPERATURE 27.78 Celsius

○ DISSOLVED OXYGEN 8.39 mg/L

○ TOTAL DISSOLVED SOLID 375.19 ppm

○ CONDUCTIVITY 0.21 mS/cm

Figure 9 Wireframe of forecasted parameters


E-TONGUE

WATER QUALITY PARAMETERS RANGE

PARAMETER	RANGE	UNIT
PH	6.5 - 8.5	No units
TURBIDITY	0.1 - 4.0	NDU
CONDUCTIVITY	0.2 - 0.8	mS/cm
TOTAL DISSOLVED SOLID	128 - 512	ppm
DISSOLVED OXYGEN	6.5 - 8.0	mg/L

THANK YOU

Figure 10 Wireframe of parameter range

2.6 Commercialization

“E-tongue” is a highly recommended system for people who are in the dry zone area, and they struggle to know the safe consumption of the groundwater. The main goal of the research group is to full fill the gap between the people and the National Water Supply & Drainage Board of Sri Lanka. Since there are several awarenesses were conducted regarding the importance of safe consumption of the groundwater. “E-tongue” has several facilities to the users such as, track the user location and storing the tested value, they able to get the WQI and safe consumption level, Forecast the water quality parameter level based on the location for a given year, predict Chronic Kidney Diseases (CKD) risk of the water. Therefore, it is going to be useful for both types of customers (standard level and business level).

“E-tongue” is a lightweight system, that is the major advantage when it comes to the market. Consumers able to the device anywhere and they can test the water from any place. It provides a few steps to connect to the system for usage. GPS tracking is used to track the consumer's location when they start to test the water. It suggests precaution when the water quality parameters are in the dangerous zone. In this system mainly target two kinds of purposes

1. New water sources

   This system helps to predict the water quality within a minute at the tested place instead of carrying water from the water sources and manual testing with several procedures. Therefore identifying water resources and tested them easily.
2. Existing water plants

   It will help to test the water quality of the existing water plants and forecast the water quality parameters of them in advance. Therefore, people can get proper awareness of the waterborne disease and they can do the precaution in advance.

2.7 Testing & Implementation

A. Implementation

Model Development

In order to predict the water quality parameters, we should have to follow prediction tasks that need to improve the model which is shown in Figure 13 Machine Learning model process.

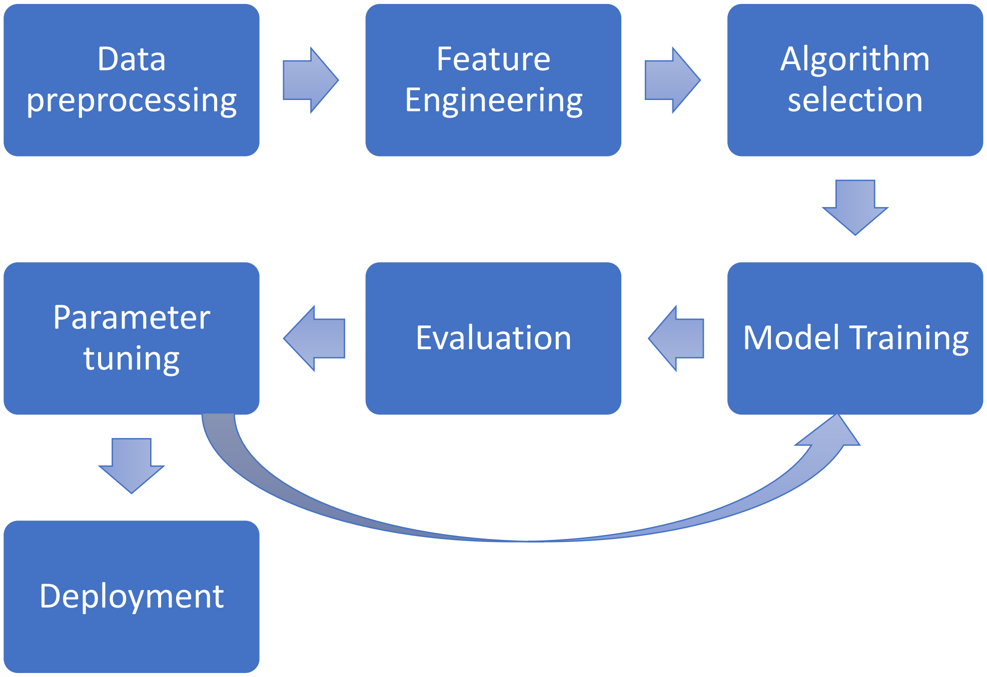
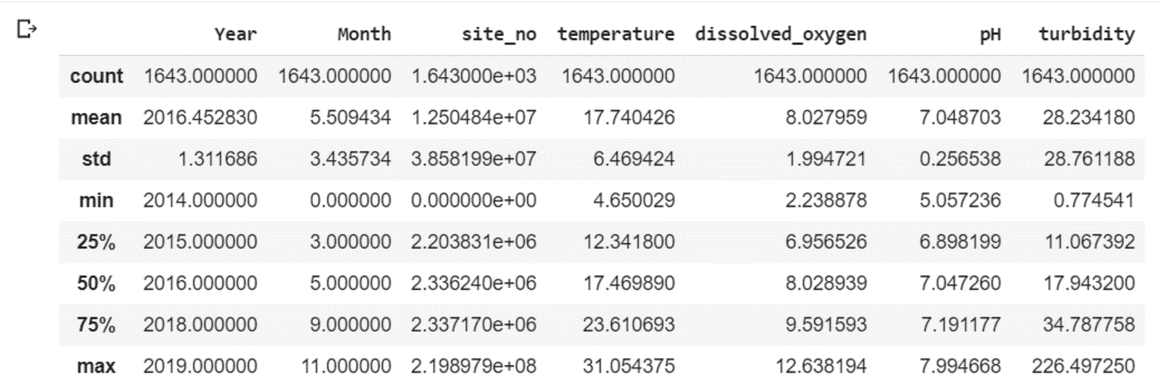


Figure 13 Machine Learning model process

- **Data pre-processing**

 Incomplete and inconsistent data will be affected the result of the machine learning prediction, therefore, we should be developed the machine learning model will good quality data then only we can expect quality output with the lowest error rate.

Data cleaning is a mechanism of identifying and removing inaccurate water quality parameters, corrupted, and noisy data from the original dataset. Python programming language provides flexible libraries for data cleaning and preprocessing the dataset. Figure 14 shows the cleaned data set.



	Year	Month	site_no	temperature	dissolved_oxygen	pH	turbidity
count	1643.000000	1643.000000	1.643000e+03	1643.000000	1643.000000	1643.000000	1643.000000
mean	2016.452830	5.509434	1.250484e+07	17.740426	8.027959	7.048703	28.234180
std	1.311686	3.435734	3.858199e+07	6.469424	1.994721	0.256538	28.761188
min	2014.000000	0.000000	0.000000e+00	4.650029	2.238878	5.057236	0.774541
25%	2015.000000	3.000000	2.203831e+06	12.341800	6.956526	6.898199	11.067392
50%	2016.000000	5.000000	2.336240e+06	17.469890	8.028939	7.047260	17.943200
75%	2018.000000	9.000000	2.337170e+06	23.610693	9.591593	7.191177	34.787758
max	2019.000000	11.000000	2.198979e+08	31.054375	12.638194	7.994668	226.497250

Figure 14 Preprocessed dataset

Below Figure 15 illustrates the feature distribution of the dataset. According to this graph get a clear understand of the feature selection for the model training. This heat map was generated against the feature indexes that are used to see the co-relations between the features of the given dataset such as year, month, site, temperature, dissolved oxygen, turbidity, and pH.

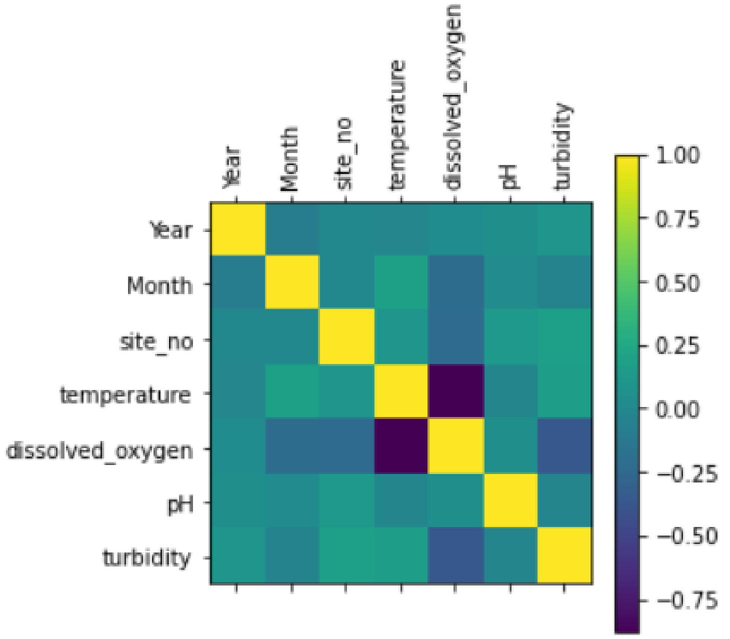


Figure 15 Visualization of the features

Applied the feature importance technique that is indicated by the ML model which will be provided many benefits to have a good model. To get a better understanding of the ML model’s logic that is important to focus on the features that we are going to select for our model.

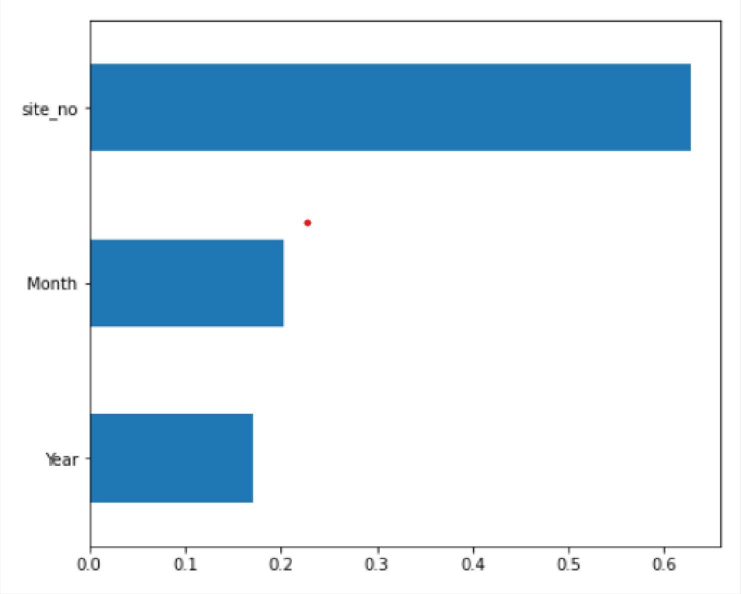


Figure 16 Feature importance graph

The above-mentioned way that we visualize the features for all variables in the data set. In this stage explore some of the ways to identify the important features that we are going to input for model training.

In order to train the model, we should select the features. As a result, shown in the Figure 16 Feature importance graph Can come up with the important feature that we are going to select. Year and month are literally less than the location when comparing.

- **Model Training and Building**

This is a vital phase of this research because the system's output highly depends on the output of this phase. Under this process, we have to divide the dataset into 2 components. 80% for training data and 20% for testing data. Hence, the machine learning model will be developed trained dataset that obtained multivariant parameters. pH, temperature, turbidity, and total dissolved oxygen are going to be forecast parameters. Trained the model by using five machine learning algorithms. Such as,

1. **Vector Auto Regression (VAR)**

Vector Auto Regression is one of the forecasting multi-parameter prediction algorithms that is used when more than one-time series variables which influence each other. The data set consists of time-series data for 53 months of 31different places water resources and water quality parameter features are turbidity, pH, temperature, and dissolved oxygen. Hence, All the four parameters are going to predict based on sites for a given time series.

$$x_{t,1} = \alpha_1 + \phi_{11}x_{t-1,1} + \phi_{12}x_{t-1,2} + \phi_{13}x_{t-1,3} + w_{t,1}$$

$$x_{t,2} = \alpha_2 + \phi_{21}x_{t-1,1} + \phi_{22}x_{t-1,2} + \phi_{23}x_{t-1,3} + w_{t,2}$$

$$x_{t,3} = \alpha_3 + \phi_{31}x_{t-1,1} + \phi_{32}x_{t-1,2} + \phi_{33}x_{t-1,3} + w_{t,3}$$

Each function will be a linear function that has lag 1 values for all the variables. In VAR(2) that has lag 2 values for all the variables that are added in the right-hand side of the equation. The above-mentioned equation VAR model was implemented at the model development phase.

According to the Figure 17 Parameter variance of VAR model to get a variance in between each parameter. At the end of the VAR model development process, plot the graph with predicted value for getting a better understanding.

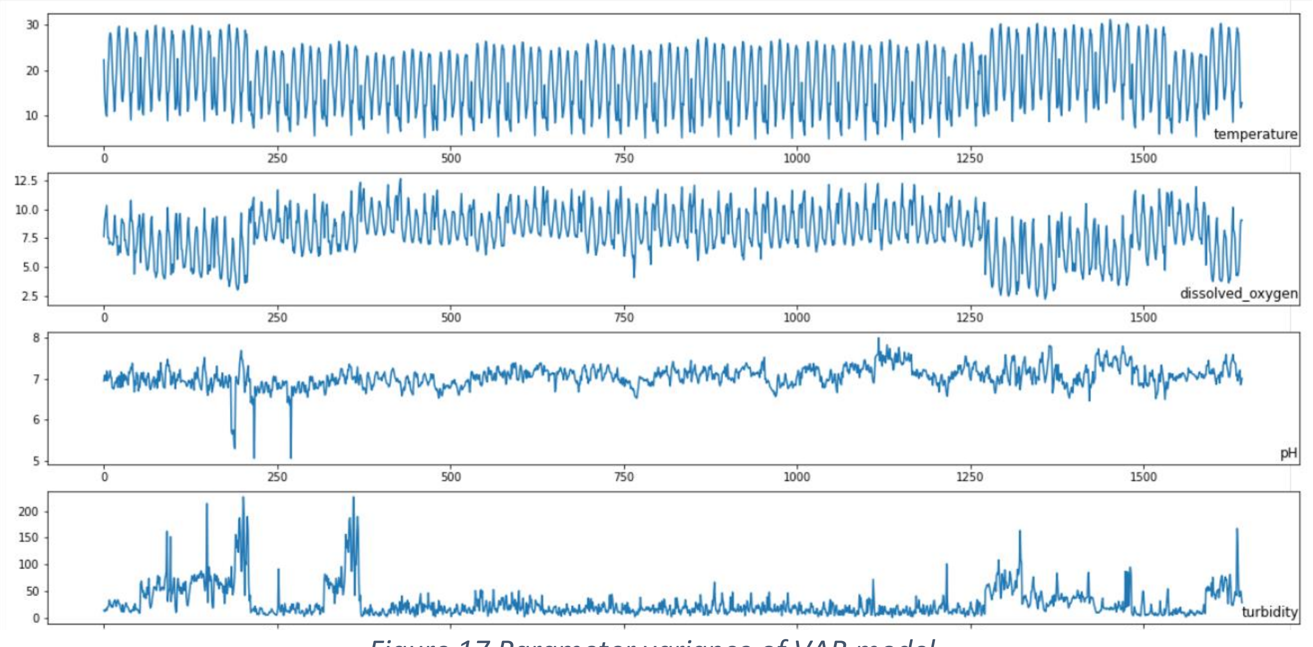


Figure 17 Parameter variance of VAR model

2. **Random Forest Regression (RFR)**

It predicts the water quality parameter levels based on a certain location or site which is used the previously measured value of the water quality parameters for certain locations or sites. Multivariate Random Forest Regression is the popular supervised machine learning algorithm for the multi-parameter regression problem.

RF consists of the ensemble of a simple tree. It creates the decision tree for the given problem based on learning techniques from the training dataset. Predicts the water quality parameter value based on the created decision tree. Figure 18 shows the generated single decision tree

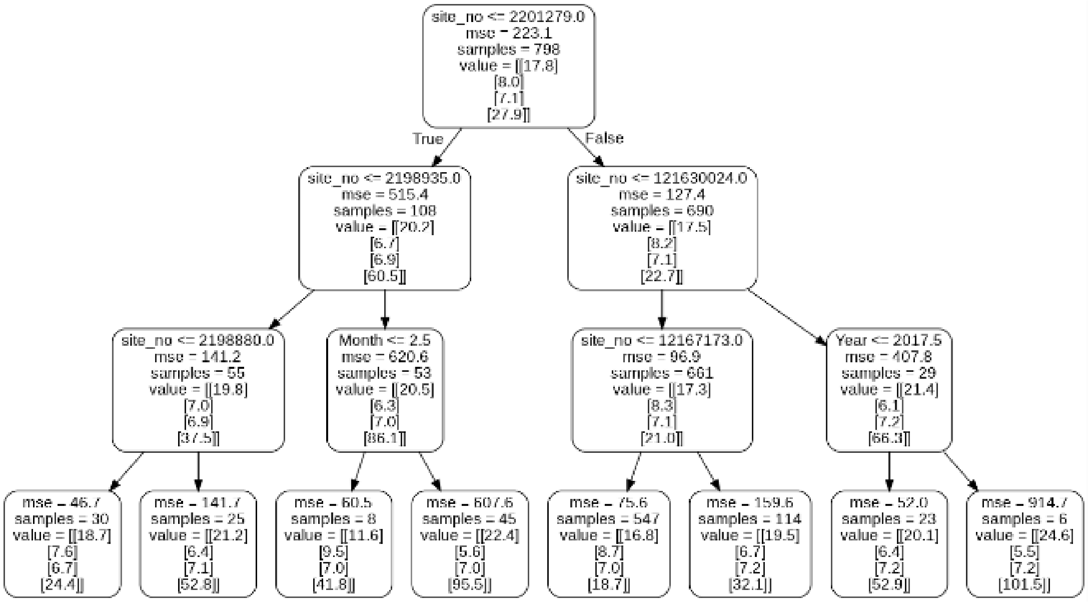


Figure 18 Single decision tree

In RFR each single decision tree responds based on the predictor values that are selected independently and the predictor values of an originally given dataset are the subset of the forest. Thus, the number of decision tree leads to predict the accuracy of the model. The equation for the optimal size of predictor input variables below.

$$\log_2 M + 1$$

M = number of input parameters (location or site, year, and month)

Prediction of the RFR is taken the average of each prediction of each tree is obtained. The following formula is used for RFR prediction.

$p = \frac{1}{k}\sum_{k=1}^{k} k^{\mathrm{th}}$ decision tree response

k = runs over the single decision tree in the forest.

3. Long-Short Term Memory (LSTM)

It is a Recurrent Neural Network (RNN) algorithm for forecasting time series data. Multivariate time series data observes more than one-time series variables. This component is focused on multiple parallel series because each independent input variables that are water quality parameters depends on the time series value. Predict each water quality parameter values that are input variables based on time series. A huge amount of data need to have for the training to obtain higher accuracy.

$$\tilde{c}_t = \tanh\left(w_c[h_{t-1}, x_t] + b_c\right)$$

above the equation is used for the input gate, here all the new information is going to store. $\tilde{c}_t$ represents a candidate for cell state at time-stamp t.

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

It tells the information to throw away from the state that called it as to forget the gate. c_t denotes the cell state or memory at time-stamp t.

$$h_t = o_t * \tanh\left(c^t\right)$$

The third equation is named it as output gate which is going to provide the activation of the final output of the algorithm at time-stamp t. All three equations were implemented at the model LSTM model development phase.

As a result of the below Figure 19 the graph showed the training and validation loss when end up with a training process. The training loss of the model is calculated when the moving average over one epoch. In this case, validation loss is higher than the training loss. It may cause insufficient data.

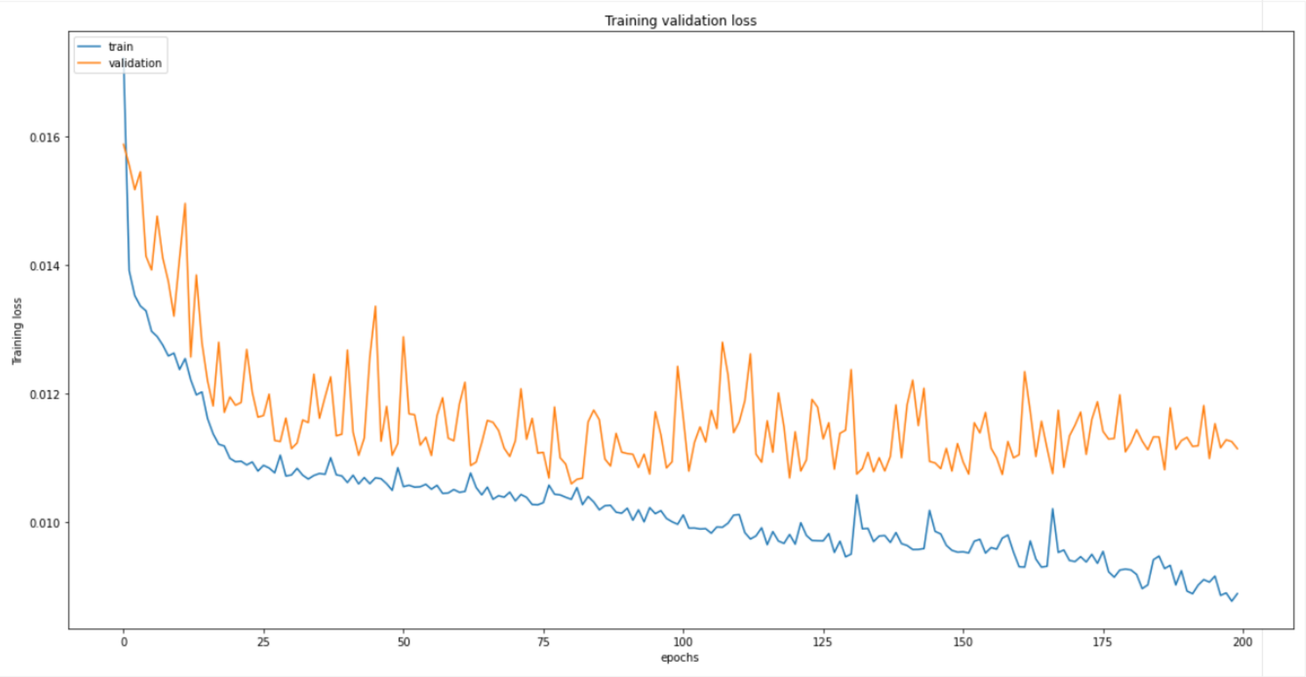


Figure 19 Training validation loss of LSTM

4. Support Vector Regression (SVR)

It is used for multiple regression problem that is involved one or more than one input and output variables. Forecasting multiple time-series data that is involved in multiple time series data predictions for given variables. The regression that is referred to the predictive model problem which is involved to predict a numerical value. It is predicting pH, temperature, turbidity and dissolved oxygen are the regression problem.

SVR function finds the coefficient that shout be minimized from that equation. The below equation is used to develop the SVR model in this component

$$L(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\alpha_i - \alpha_i^*\right)\left(\alpha_j - \alpha_j^*\right)G(x_i, x_j) + \varepsilon\sum_{i=1}^{N}\left(\alpha_i + \alpha_i^*\right) - \sum_{i=1}^{N} y_i\left(\alpha_i - \alpha_i^*\right)$$

Function of the value

$$f(x) = \sum_{n=1}^{N}\left(\alpha_n - \alpha_n^*\right)G(x_n, x) + b\,.$$

These equations were implemented to predict the water quality parameters at the model development phase.

As a graph shown in Figure 20 that will be generated at the end of the SVR model training process. pH value was predicted according to the given input. Each of the parameters is predicted separately and combined them on a single model. Timeseries prediction is used as the realistic way to predict the values when the data set has to have continuous values along with time series even the prediction with multiple variants.

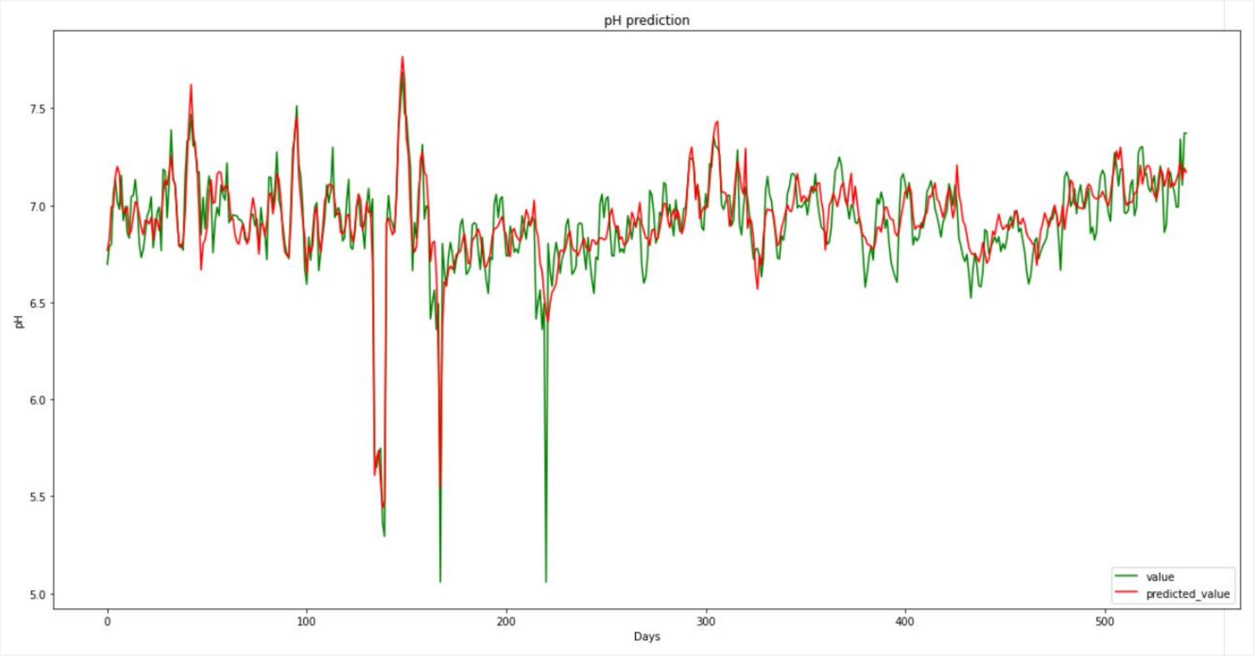


Figure 20 SVR prediction of pH

5. K-Nearest Neighbors (KNN)

This algorithm is used for classification and regression problems, but in this component, It is used for regression problems that are involved with multiple parameters. It should be considered the consistency of the data therefore it is involved in time series prediction as well. It is used to forecast water quality parameters. Its focus on efficient implementation with the complex dataset. The prediction of the model is based on the median or mean. KNN is used to calculate the average numerical value of the K-nearest neighbors. Another approach that has to be done in KNN is inversed the distance weighted average. Euclidean, Manhattan, and Minkowski are three different distance measurement techniques that valid only for continuous variables.

Distance functions

Euclidean $$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Manhattan $$\sum_{i=1}^{k}|x_i - y_i|$$

Minkowski $$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$$

Below the Figure 21 was plotted once the training is finished. That is shown in the original and predicted value of the water quality parameter. Each and every parameter is trained independently again with input data.

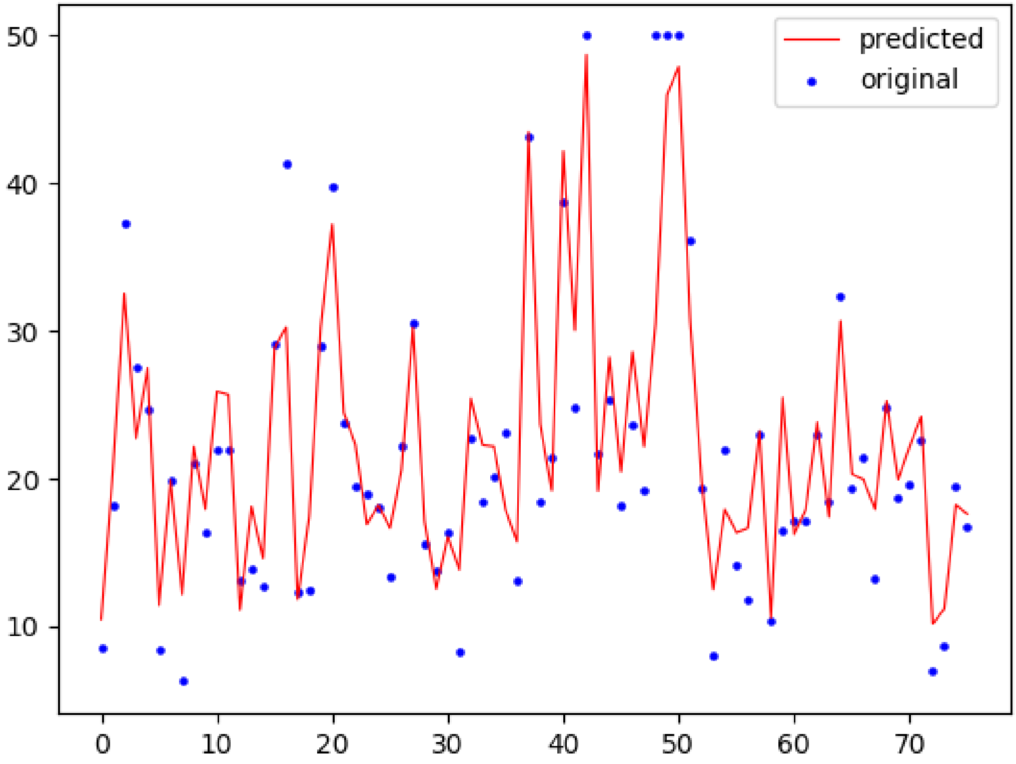


Figure 21 Plotted graph of KNN prediction

- **Model Validation**

  It is the mechanism of checking the correctness of the machine learning model output against the real data. Model validation can be done by measuring the accuracy of the results by using train datasets and test datasets. The advantages of model validations are early identification of bugs, cost reduction, and improving the quality of the model. There are plenty of machine learning model validation techniques available in the python environment. Each of the models was validated in this phase. According to this component description following validation methods are used for each model development mechanism.

  - ❖ Train and Test split
  - ❖ Time series cross-validation
  - ❖ K-fold cross-validation
  - ❖ Nested cross-validation

- **Model Testing**

  It can be done by testing the data used to train the model, testing the features of the model, and finally testing machine learning algorithms of the model.

  - o Testing the data which used to train the model – In this process checks whether datasets contains adversary dataset or not. If it has an adversary dataset it will make Data Poisoning therefore result will be varied. To get correct and accurate output adversary dataset should be removed from the original dataset in this process.
  - o Testing the features of the model – In this process, if a machine learning model contains redundant/irrelevant features it will make prediction bugs, therefore, those features will be removed to get better prediction results.

Testing machine learning algorithms – In this process which machine learning algorithm will be provided the best performance and accuracy with new data sets at regular intervals will be selected for the model

B. Testing

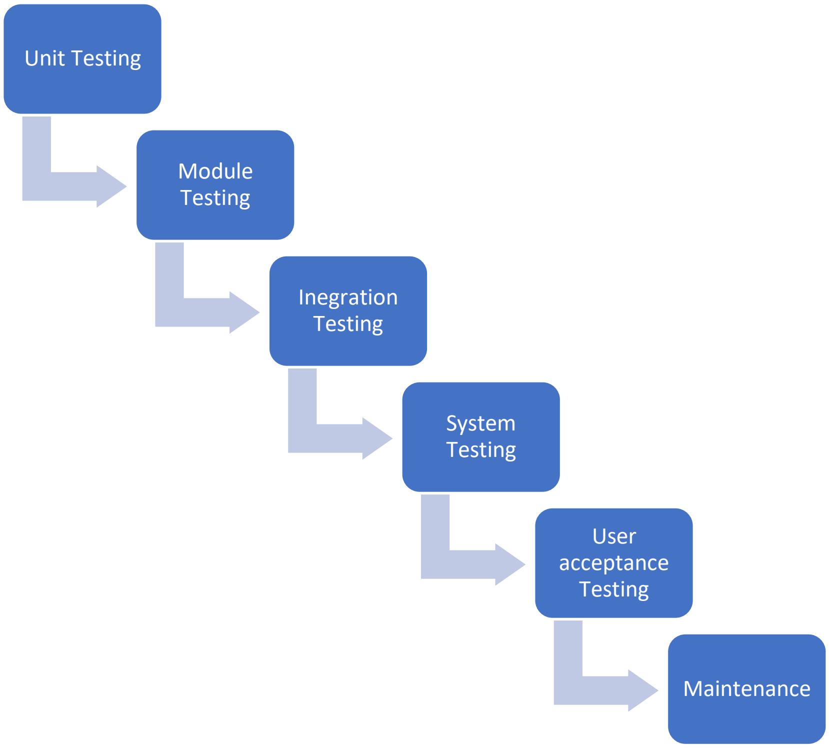


Figure 22 Types of software testing

Figure 22 is shown a prominent way to test the software before release at each development phase, testing the application, and fix the bugs. Through testing, applications become more quality and the best deliverable product. Software testing out could be very critical in the development life cycle to identify defects and errors that have been made during the development stages. And it is very essential to make sure of the quality of the product. it is required for a powerful performance of product or software application. it is required every phase that we are following in SDLC which has below testing categories.

1. Unit Testing

   In this phase, each of the individual units is tested and make sure every function is working correctly with bug-free. As a result of this component pass successfully that is noted as an error-free component and it will become ready to integrate with the main component. This testing part comes under white box testing.

2. Module Testing

   It is defined as another type of software testing, which is checked subroutine, a subprogram, and class. This is not going to test for your own module. Therefore, this was reviewed by other research team members.

3. Integration Testing

   Here, all the individual components or units are combined and tested as a group. The purpose of this testing is to be evaluating the compliance of each component or system which contains the functional requirements.

4. System Testing

   It is conducted when the whole system was integrated. The system should be evaluated with each functional requirement. The purpose of this testing is to be evaluating the end to end specifications. Once the bug was identified on this testing then It will debug by the team members.

5. User Acceptance Testing

   This testing was performed by the client or the end-user. It will be tested at the end of the testing phase. The feedback and user experience are given by target users that need to find the satisfaction of the user.

6. Maintenance

   The maintenance phase is the last phase of this SDLC model. Here, maintain the software updates, repairs, and fixes of the application are considered as some of the functionalities conducted in this phase.

The developed system is passed under all testing phases with error-free. The entire system should be divided when performing on the testing phase which will be a realistic way of testing.

- Mobile application – Frontend

  Test the android application which makes sure to perform without any failures.

- Server API - Backend

  The developed ML model is tested and adjust the parameter for increasing the accuracy of each model.

Mobile application – Frontend Testing

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

Table 2 Test case for fetch user current location

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

Table 3 Test case for search location

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

Table 4 Test case for check calendar

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

Table 5 Test case for negative case 1

Test case ID	005
Test case scenario	Check the submit button disabled when the location field empty.
Test steps	a. The user navigates to the forecast screen b. Keep the location field empty c. Select month and year
Test data	Month = December , Year = 2022
Expected result	The alert dialog should be shown as “Location need to enable”
Actual result	As expected
Pass/Fail	Pass

Table 6 Test case for negative case 2

Server API - Backend Testing

Test case ID	006
Test case scenario	Check forecast parameters
Test steps	a. The user navigates to the forecast page b. User enter location or site c. User select forecast year & month d. Submit

Test data	Location = Nochchiyagama, Sri Lanka
	Month = December , Year = 2022

Expected result	Display the water quality parameters’ level in the graph.
Actual result	As expected
Pass/Fail	Pass

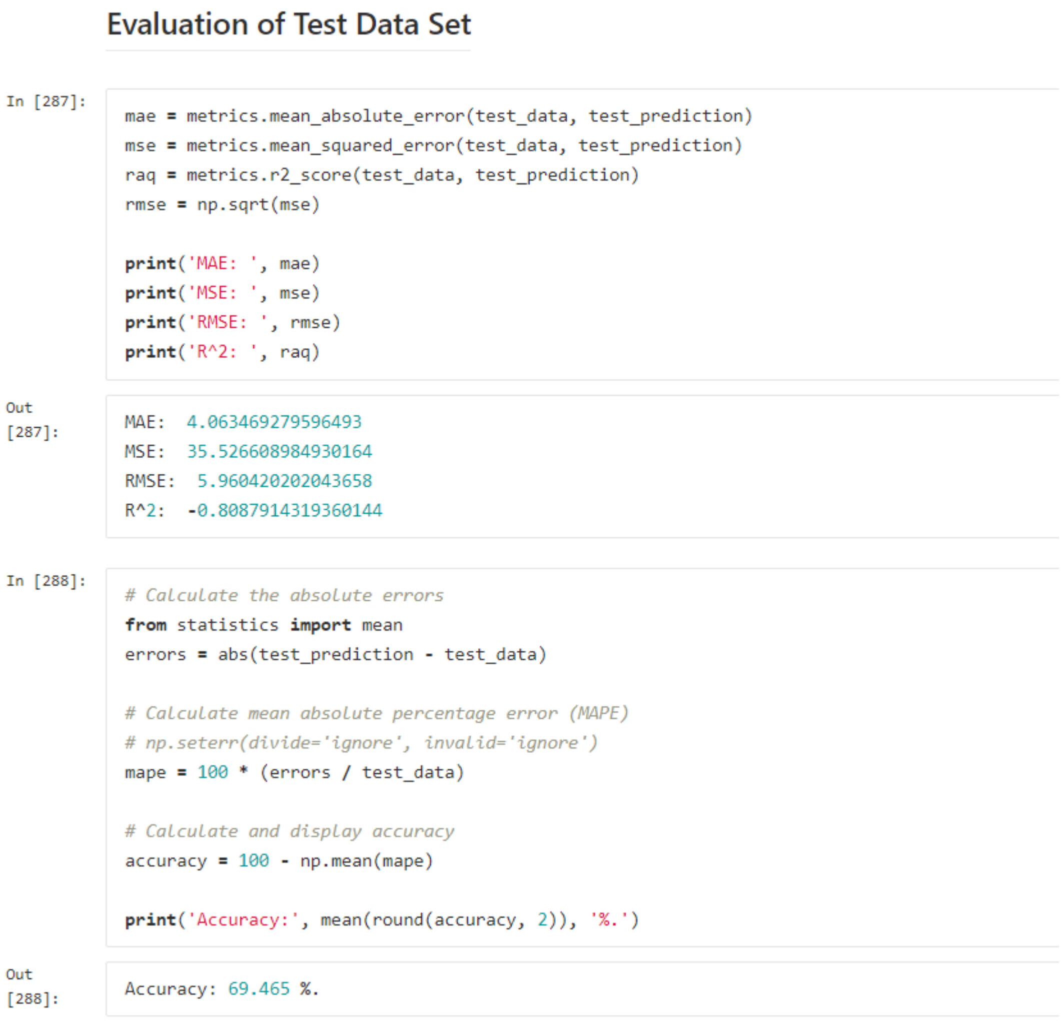
Table 7 Test case for backend API

3. RESULT & DISCUSSIONS

3.1 Results of each model

- **VAR**

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



Evaluation of Test Data Set

In [287]:

```
mae = metrics.mean_absolute_error(test_data, test_prediction)
mse = metrics.mean_squared_error(test_data, test_prediction)
raq = metrics.r2_score(test_data, test_prediction)
rmse = np.sqrt(mse)

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

Out [287]:

```
MAE:  4.063469279596493
MSE:  35.526608984930164
RMSE:  5.960420202043658
R^2:  -0.8087914319360144
```

In [288]:

```
# Calculate the absolute errors
from statistics import mean
errors = abs(test_prediction - test_data)

# Calculate mean absolute percentage error (MAPE)
# np.seterr(divide='ignore', invalid='ignore')
mape = 100 * (errors / test_data)

# Calculate and display accuracy
accuracy = 100 - np.mean(mape)

print('Accuracy:', mean(round(accuracy, 2)), '%.')
```

Out [288]:

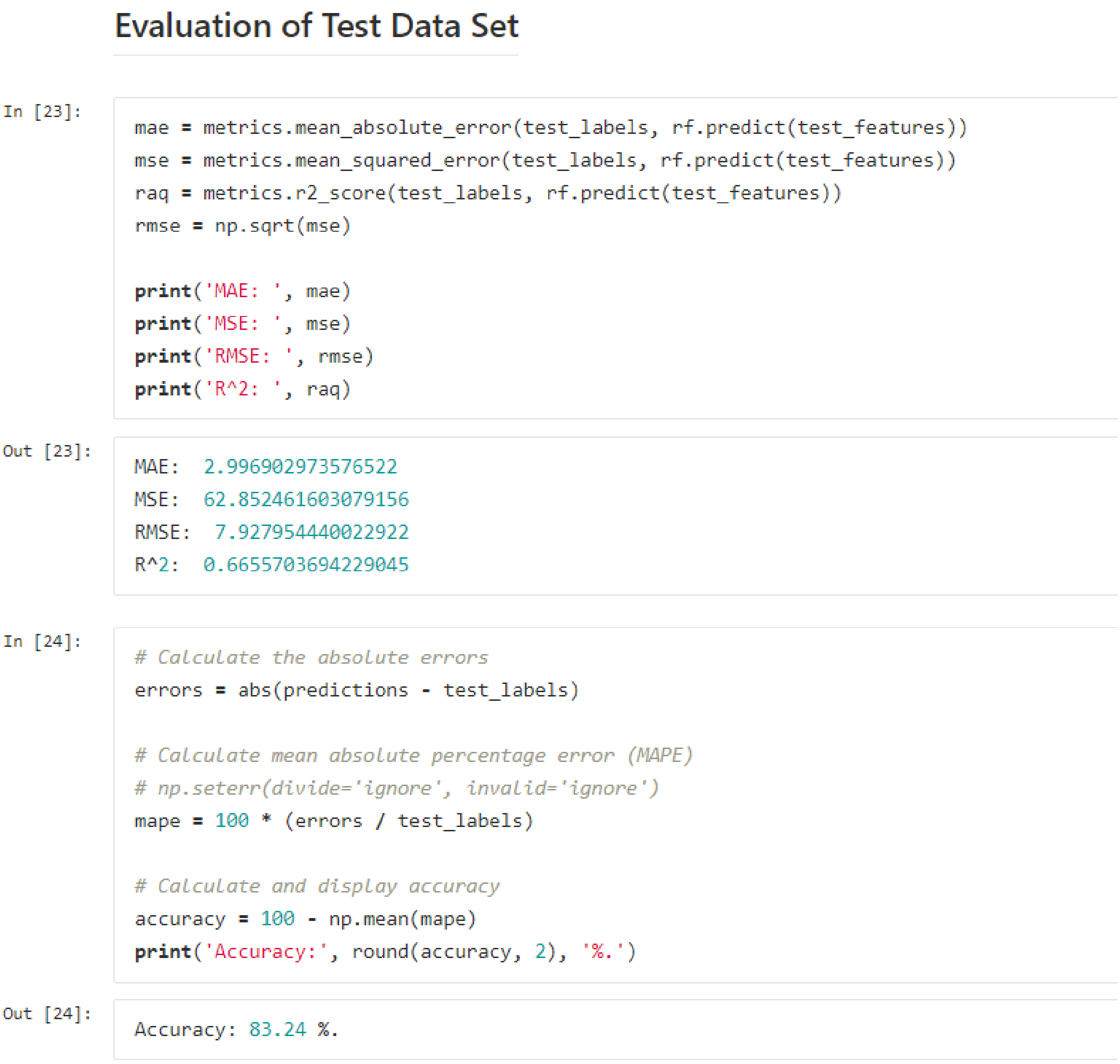
```
Accuracy: 69.465 %.
```

Figure 23 Evaluation of VAR

Figure 23 shows the result of the trained model which tested against the test data set. The model is trained and tested using Google Colab which has less computational power that runs on the cloud server.

- **RFR**

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



Evaluation of Test Data Set

In [23]:

```
mae = metrics.mean_absolute_error(test_labels, rf.predict(test_features))
mse = metrics.mean_squared_error(test_labels, rf.predict(test_features))
raq = metrics.r2_score(test_labels, rf.predict(test_features))
rmse = np.sqrt(mse)

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

Out [23]:

```
MAE:  2.996902973576522
MSE:  62.852461603079156
RMSE:  7.927954440022922
R^2:  0.6655703694229045
```

In [24]:

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

# Calculate mean absolute percentage error (MAPE)
# np.seterr(divide='ignore', invalid='ignore')
mape = 100 * (errors / test_labels)

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

Out [24]:

```
Accuracy: 83.24 %.
```

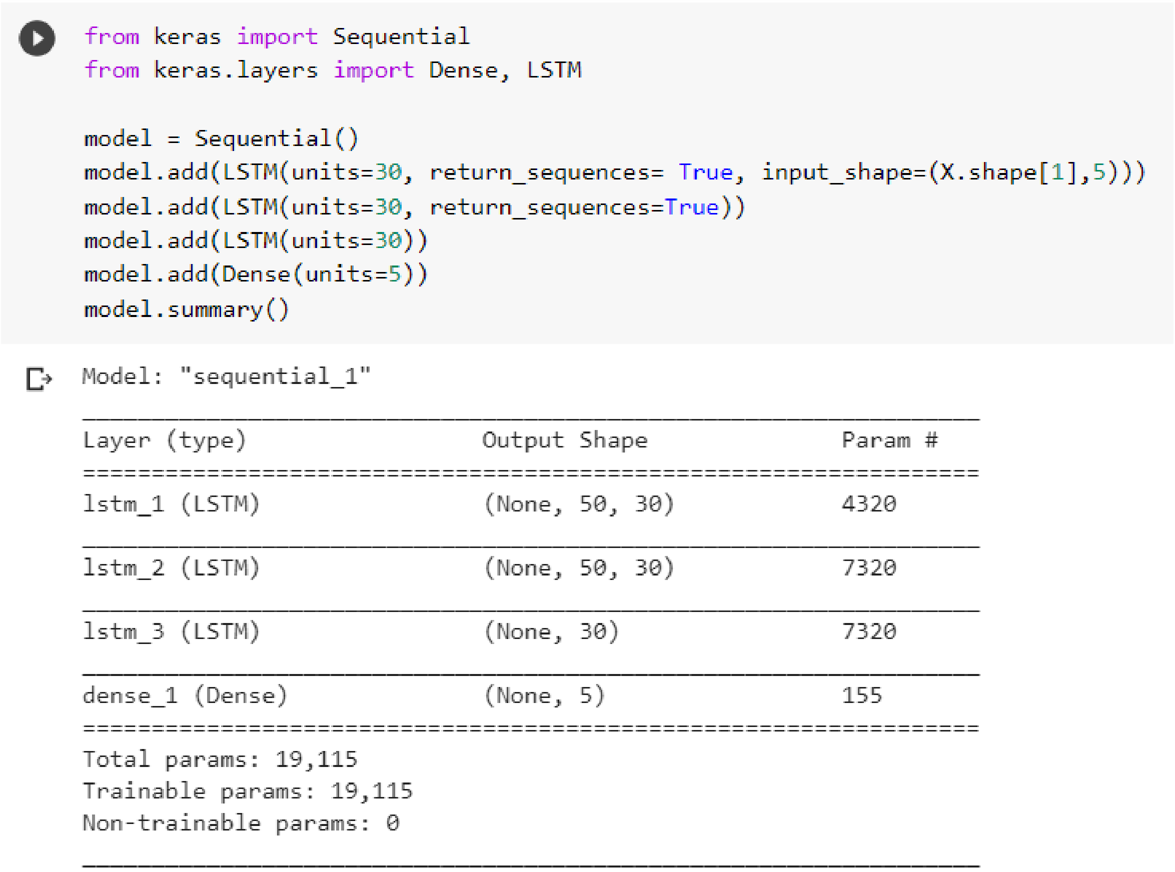
Figure 24 Evaluation of RFR

The below figure 24 shows the evaluation of the trained model against the test data set. That provides the accuracy of the model.

- **LSTM**

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

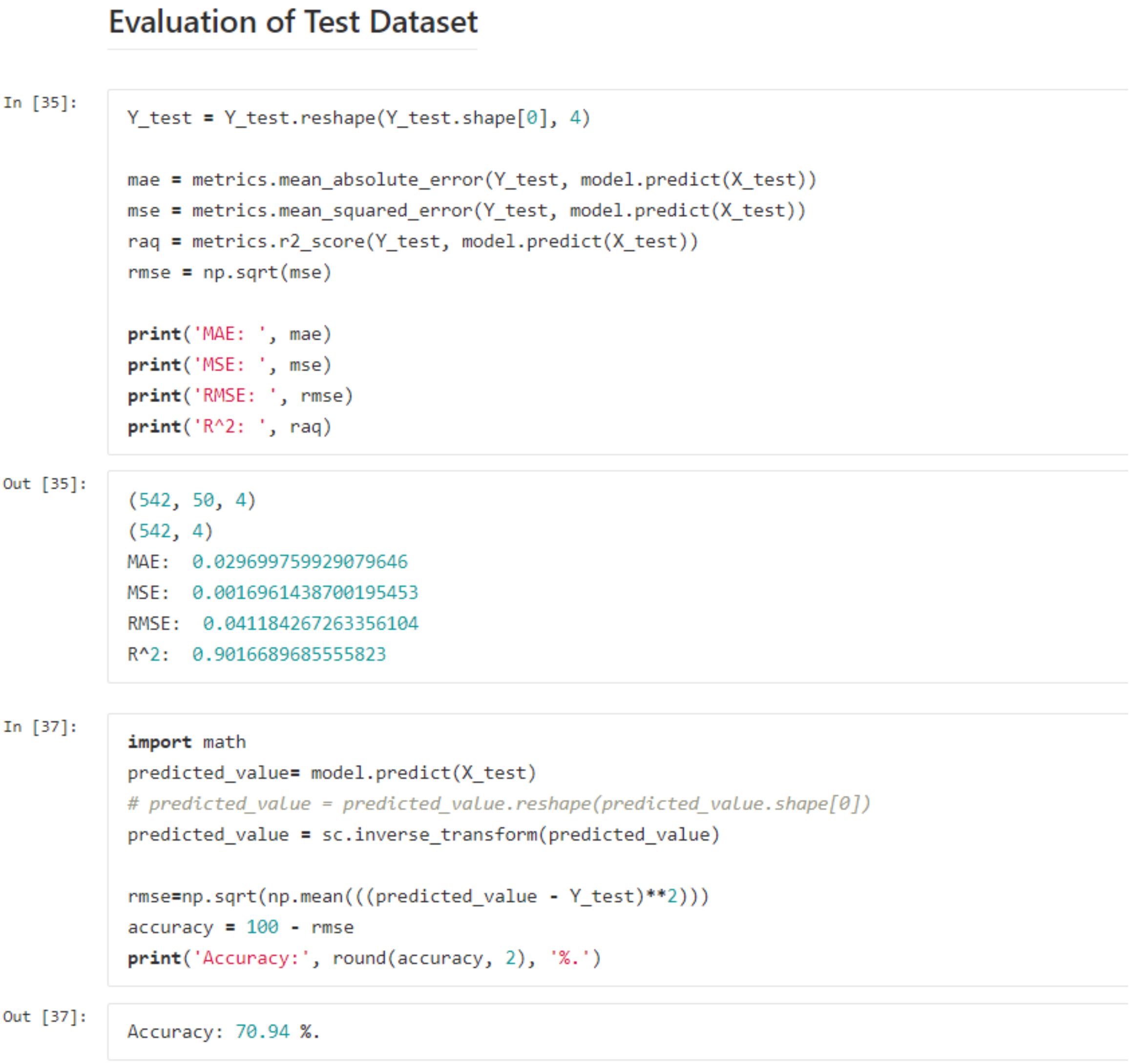
The following figure 25 illustrates the model summarization of LSTM which has types of layers and dense. Figure 26 shows the accuracy of the model which is evaluated against the test dataset.



```python
from keras import Sequential
from keras.layers import Dense, LSTM

model = Sequential()
model.add(LSTM(units=30, return_sequences= True, input_shape=(X.shape[1],5)))
model.add(LSTM(units=30, return_sequences=True))
model.add(LSTM(units=30))
model.add(Dense(units=5))
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
lstm_1 (LSTM)                (None, 50, 30)            4320      
_________________________________________________________________
lstm_2 (LSTM)                (None, 50, 30)            7320      
_________________________________________________________________
lstm_3 (LSTM)                (None, 30)                7320      
_________________________________________________________________
dense_1 (Dense)              (None, 5)                 155       
=================================================================
Total params: 19,115
Trainable params: 19,115
Non-trainable params: 0
_________________________________________________________________
```

Figure 25 Model summarization of LSTTM



Evaluation of Test Dataset

In [35]:

```
Y_test = Y_test.reshape(Y_test.shape[0], 4)

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

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

Out [35]:

```
(542, 50, 4)
(542, 4)
MAE:  0.029699759929079646
MSE:  0.0016961438700195453
RMSE:  0.041184267263356104
R^2:  0.9016689685555823
```

In [37]:

```
import math
predicted_value= model.predict(X_test)
# predicted_value = predicted_value.reshape(predicted_value.shape[0])
predicted_value = sc.inverse_transform(predicted_value)

rmse=np.sqrt(np.mean(((predicted_value - Y_test)**2)))
accuracy = 100 - rmse
print('Accuracy:', round(accuracy, 2), '%.')
```

Out [37]:

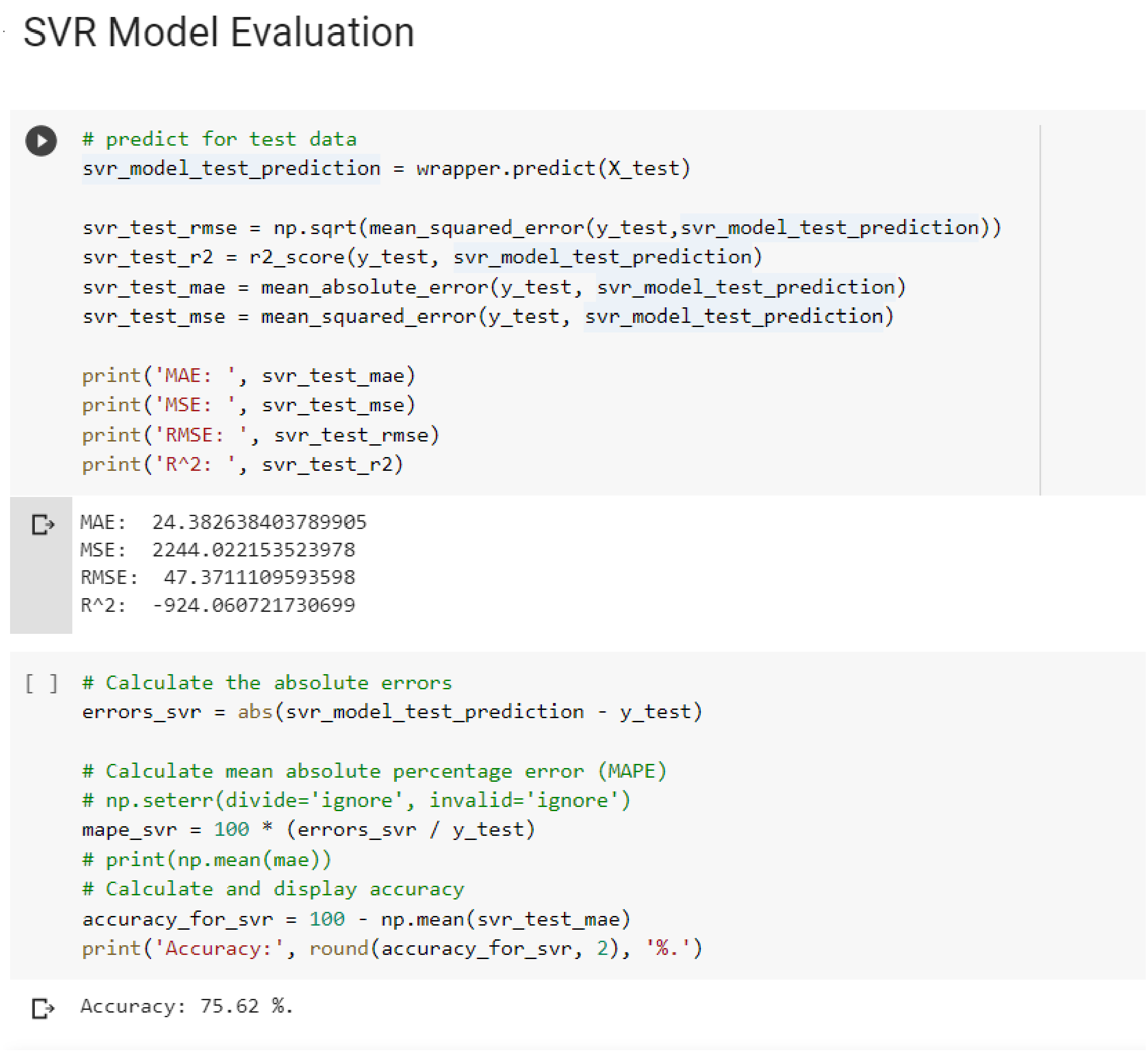
```
Accuracy: 70.94 %.
```

Figure 26 Evaluation of LSTM

- **SVR**

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

Below the figure 27 shows the model evaluation of SVR. The trained model was evaluated against the test data set that provides the accuracy of the test data. Therefore, this model gives 75.27% accuracy.



SVR Model Evaluation

```
# predict for test data
svr_model_test_prediction = wrapper.predict(X_test)

svr_test_rmse = np.sqrt(mean_squared_error(y_test,svr_model_test_prediction))
svr_test_r2 = r2_score(y_test, svr_model_test_prediction)
svr_test_mae = mean_absolute_error(y_test, svr_model_test_prediction)
svr_test_mse = mean_squared_error(y_test, svr_model_test_prediction)

print('MAE: ', svr_test_mae)
print('MSE: ', svr_test_mse)
print('RMSE: ', svr_test_rmse)
print('R^2: ', svr_test_r2)
```

```
MAE:  24.382638403789905
MSE:  2244.022153523978
RMSE:  47.3711109593598
R^2:  -924.060721730699
```

```
[ ] # Calculate the absolute errors
    errors_svr = abs(svr_model_test_prediction - y_test)

    # Calculate mean absolute percentage error (MAPE)
    # np.seterr(divide='ignore', invalid='ignore')
    mape_svr = 100 * (errors_svr / y_test)
    # print(np.mean(mae))
    # Calculate and display accuracy
    accuracy_for_svr = 100 - np.mean(svr_test_mae)
    print('Accuracy:', round(accuracy_for_svr, 2), '%.')
```

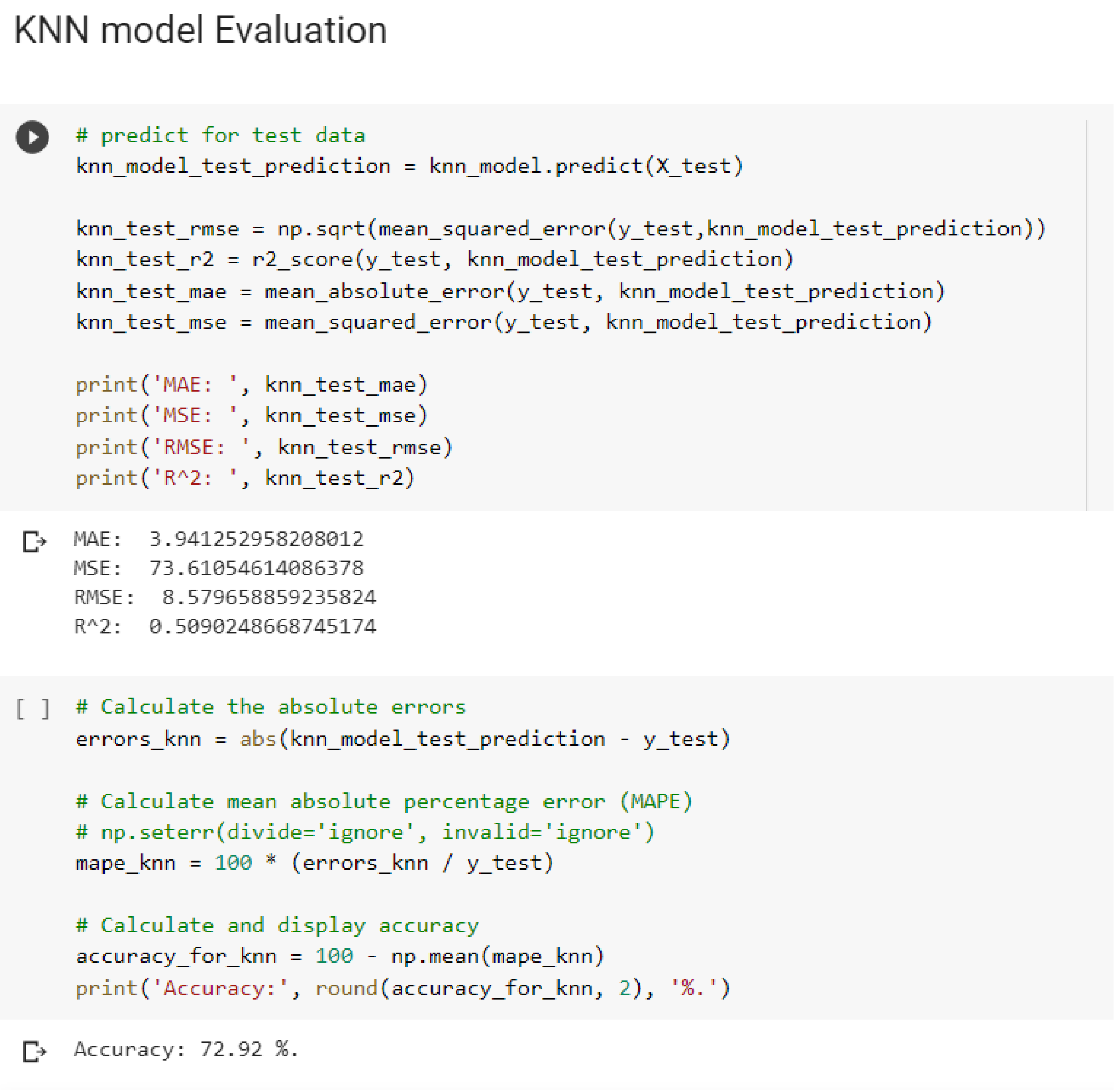
```
Accuracy: 75.62 %.
```

Figure 27 Evaluation of SVR

- **KNN**

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

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



KNN model Evaluation

```
# predict for test data
knn_model_test_prediction = knn_model.predict(X_test)

knn_test_rmse = np.sqrt(mean_squared_error(y_test,knn_model_test_prediction))
knn_test_r2 = r2_score(y_test, knn_model_test_prediction)
knn_test_mae = mean_absolute_error(y_test, knn_model_test_prediction)
knn_test_mse = mean_squared_error(y_test, knn_model_test_prediction)

print('MAE: ', knn_test_mae)
print('MSE: ', knn_test_mse)
print('RMSE: ', knn_test_rmse)
print('R^2: ', knn_test_r2)
```

```
MAE:  3.941252958208012
MSE:  73.61054614086378
RMSE:  8.579658859235824
R^2:  0.5090248668745174
```

```
[ ] # Calculate the absolute errors
    errors_knn = abs(knn_model_test_prediction - y_test)

    # Calculate mean absolute percentage error (MAPE)
    # np.seterr(divide='ignore', invalid='ignore')
    mape_knn = 100 * (errors_knn / y_test)

    # Calculate and display accuracy
    accuracy_for_knn = 100 - np.mean(mape_knn)
    print('Accuracy:', round(accuracy_for_knn, 2), '%.')
```

```
Accuracy: 72.92 %.
```

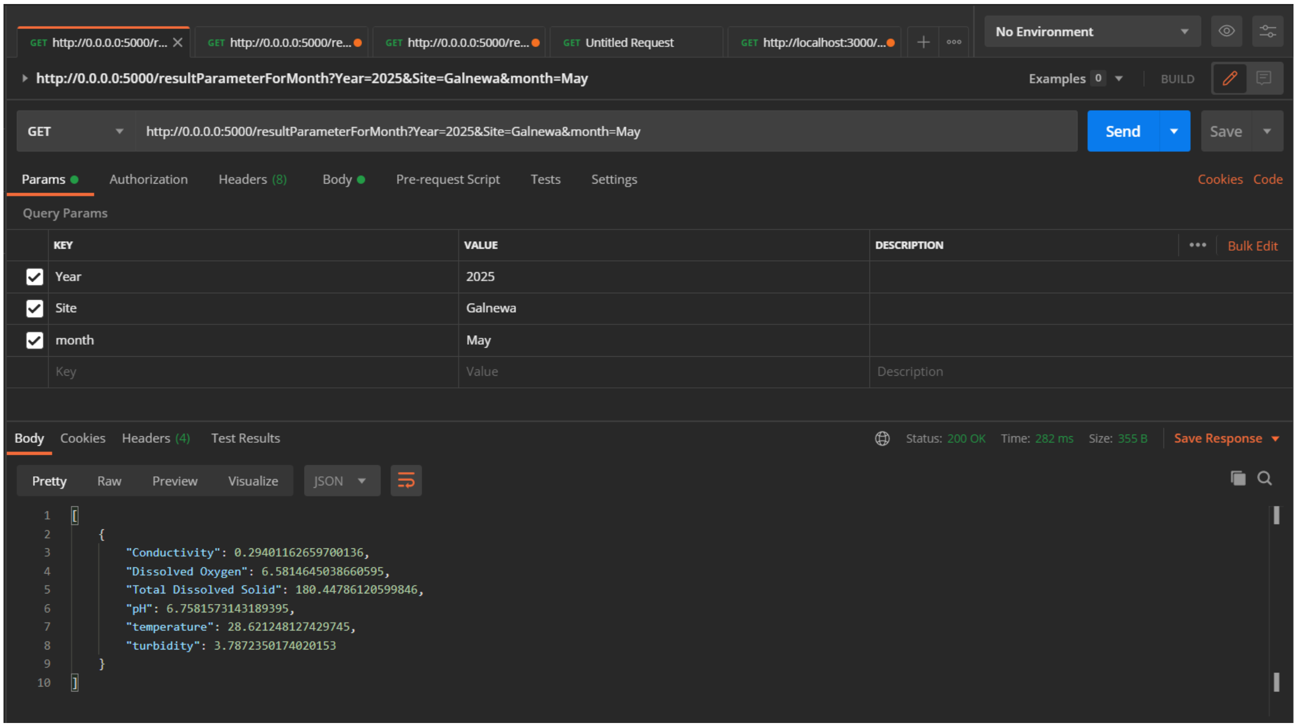
Figure 28 Evaluation of KNN

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

Model Deployment

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

Flask is an efficient python server that is using microservice that allows us to build RESTful APIs that need to communicate between the back end and front end via HTTP protocols with minimum configuration. Figure 29 shows the sample output of the GET method of this component. Google map API is used to fetch the user location and the user able to search the location via that service.



http://0.0.0.0:5000/resultParameterForMonth?Year=2025&Site=Galnewa&month=May

GET http://0.0.0.0:5000/resultParameterForMonth?Year=2025&Site=Galnewa&month=May

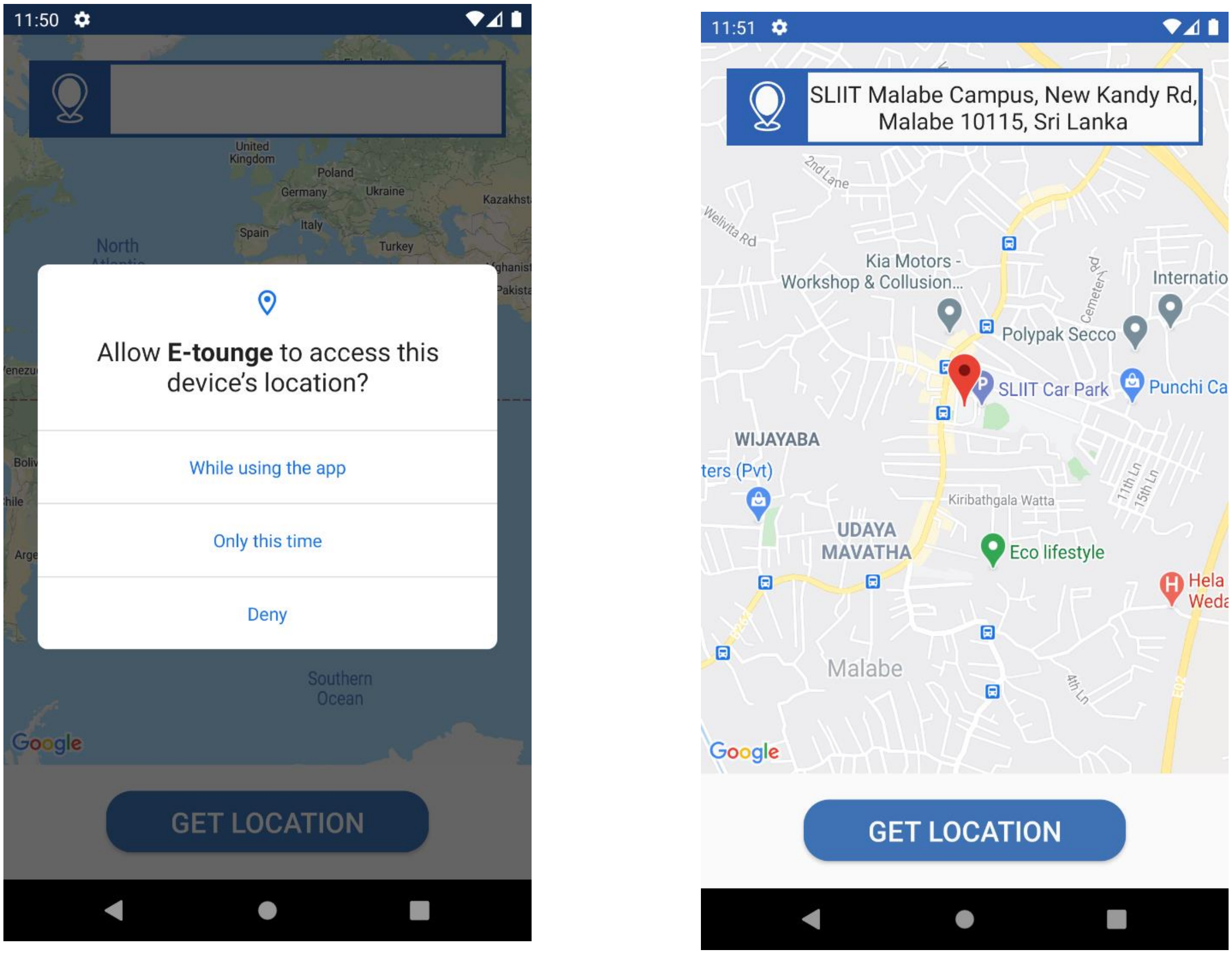
	KEY	VALUE	DESCRIPTION
☑	Year	2025	
☑	Site	Galnewa	
☑	month	May	
	Key	Value	Description

Status: 200 OK Time: 282 ms Size: 355 B Save Response

```
[
    {
        "Conductivity": 0.29401162659700136,
        "Dissolved Oxygen": 6.5814645038660595,
        "Total Dissolved Solid": 180.44786120599846,
        "pH": 6.7581573143189395,
        "temperature": 28.621248127429745,
        "turbidity": 3.7872350174020153
    }
]
```

Figure 29 API output on Postman

User Interface of Mobile Application



11:50

Allow **E-tounge** to access this device's location?

While using the app

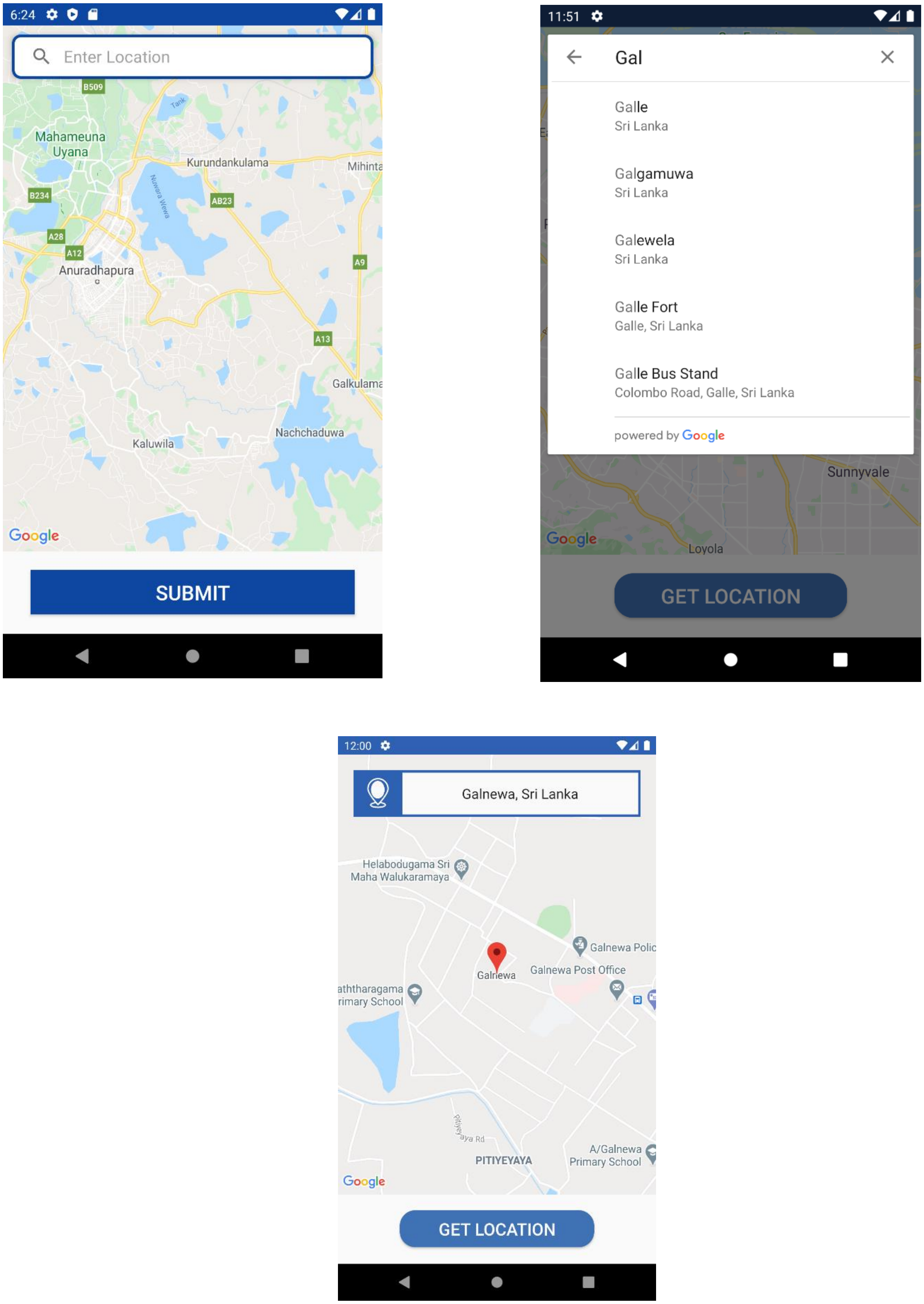
Only this time

Deny

11:51

SLIIT Malabe Campus, New Kandy Rd, Malabe 10115, Sri Lanka

Figure 30 UI of fetch user location



Gal

Galle
Sri Lanka

Galgamuwa
Sri Lanka

Galewela
Sri Lanka

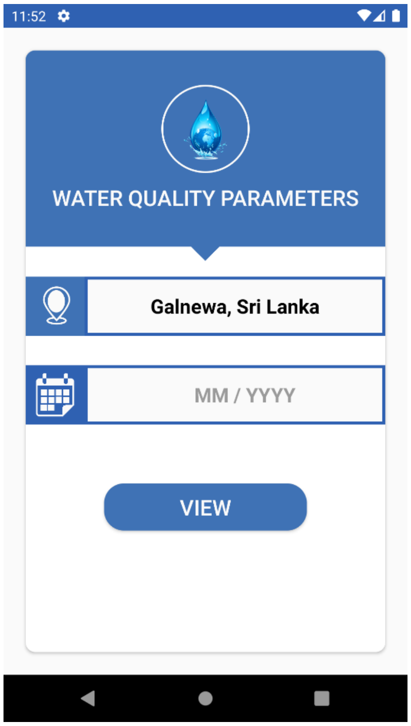
Galle Fort
Galle, Sri Lanka

Galle Bus Stand
Colombo Road, Galle, Sri Lanka

powered by Google

Galnewa, Sri Lanka

Figure 31 UI of Search location



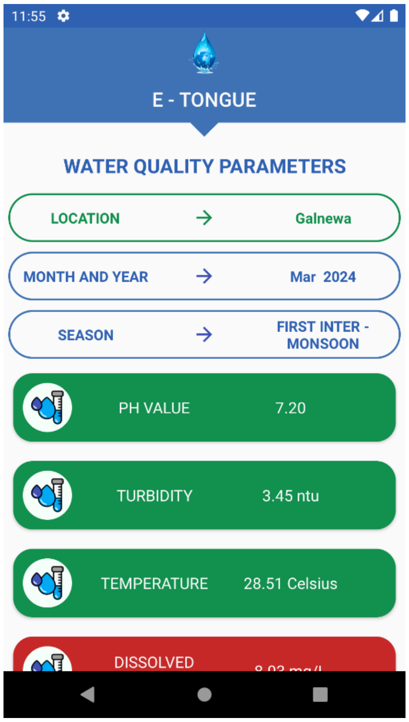
WATER QUALITY PARAMETERS

Galnewa, Sri Lanka

MM / YYYY

VIEW

Figure 32 UI of input parameter



E - TONGUE

WATER QUALITY PARAMETERS

LOCATION → Galnewa

MONTH AND YEAR → Mar 2024

SEASON → FIRST INTER - MONSOON

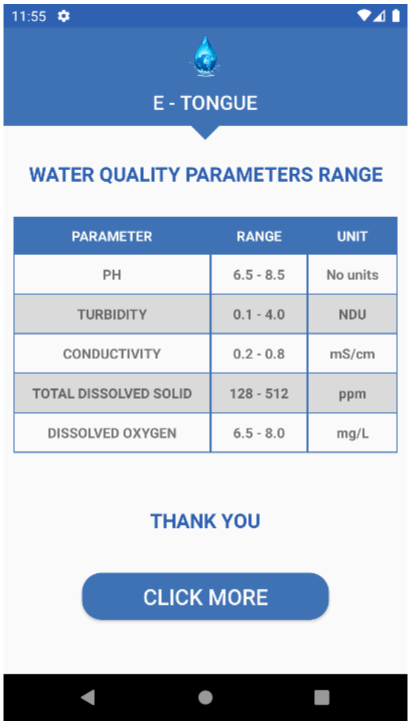
PH VALUE 7.20

TURBIDITY 3.45 ntu

TEMPERATURE 28.51 Celsius

DISSOLVED 8.03 mg/l

Figure 33 UI of forecasted parameters



E - TONGUE

WATER QUALITY PARAMETERS RANGE

PARAMETER	RANGE	UNIT
PH	6.5 - 8.5	No units
TURBIDITY	0.1 - 4.0	NDU
CONDUCTIVITY	0.2 - 0.8	mS/cm
TOTAL DISSOLVED SOLID	128 - 512	ppm
DISSOLVED OXYGEN	6.5 - 8.0	mg/L

THANK YOU

CLICK MORE

Figure 34 Specific parameter details

3.2 Research Finding

The basic idea of this research project is to develop an android app which able to help the National Water Supply and Drainage Board as well as the stand users who are well known to use the smart application. This application provides the result of forecasting water quality parameter value for a given location.

Analyzing the technology area

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

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

Choosing the best algorithm

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

3.3 Discussion

This part discusses the accuracy level of the five model and proof the finalized model selection through the training and testing accuracy. Earlier stage, get the accuracy of the training data then it will be validated against the test data the gives the accuracy of the model. Thus, gathering the accuracy of each model and listed them into one table for easier discussion.

Algorithm	**Accuracy (%)**	**Error Percentage (%)**
Vector Auto Regression (VAR)	69.47	30.53
Random Forest Regression (RFR)	83.24	16.24
Long Short-Term Memory (LSTM)	70.94	29.04
Support Vector Regression (SVR)	75.52	24.48
K- Nearest Neighbors (KNN)	72.92	27.08

Table 8 Accuracy of each model

According to the Table 8, Random Forest Regression (RFR) provides the lowest error rate (16.24%) and highest accuracy (83.24%) which will be best enough to predict the parameters. Therefore, RFR was selected for a future prediction. In order to improve the performance of the model by parameter tuning with several ways that give higher accuracy of the model.

RFR algorithm fills all the mandatory requirements for the prediction of the system. That has the capability of forecasting water quality parameters with a smaller number of input sample data. Therefore, it should be most suitable in Sri Lanka. The result of this component has to be forecast the water quality parameter for a given location, year, and month. It will be a major goal of the research component.

Now, predict few numbers of water quality parameters such as pH, temperature, turbidity, total dissolved solids, and electric conductivity but other quality parameters also interact with the quality of the data water. Therefore, collecting other parameter data from more dry zone areas and do the model training with a comparatively higher dataset than now. LSTM is the best algorithm for a huge amount of multivariable time series data. Improve the accuracy by change the algorithm and parameter tuning in the future.

3.4 Summary of the student contribution

Member	Component	Tasks
Thenuja S	Forecasting water quality parameter	• Feasibility study to perceive the requirement of the aspect. • Identify the dataset which will be used in machine learning model. • Data preprocessing to reduce the redundant data and increase the accuracy rate. • Identify the most suitable algorithm in order to produce accurate results. • Test the developed machine learning model with a dataset that has known results in order to obtain the accuracy. • Deploy a model into REST full web service. • Connect with Google Map API to fetch user location. • Alert the user via mobile app if the parameter range is different in future.

4. CONCLUSION

This individual thesis of “E – Tongue: A smart tool to predict the safe consumption of groundwater” is especially focused on **Forecasting Water Quality Parameters**. E – Tongue is a mobile application, which is well suitable for people, who are in the dry zone area. In Sri Lanka, people depend on common water resources and water supply for their routine life and they struggle to consume safe consumption water. To avoid these problems, the implemented system provides the facility to forecasting water quality parameters of groundwater in specific common water sources and it considering the monsoon seasonal changes of Sri Lanka.

E – Tongue android mobile app initialize with fetch the consumer location or their desired location to forecast the parameter level. All the prediction has to be done in the backend. Client services communicate with the backend through the API calls. Google Map API is a third-party API that integrates with this system. ML model is used to predict the parameter values for the future month that deployed onto the cloud service. That will be centralized so consumers are able to access from anywhere at any time. The system has few barriers to handle it, mobile should be connected with the internet and enable the device location to fetch their current location.

RFR model assists with the past 4 years data that not enough for initial LSTM model prediction. E – Tongue has severed another facility to predict the water quality in real-time. That tested real-time data is stored in the database which will be a reliable way to do the prediction of water quality parameter in the future with time series algorithm.

This implemented system is initiated with the main scope of serving people who are in the dry zone area. Meanwhile, that could be a motivation to provide service for the system commercialization aspect. It is going to bring much interest to the department of the National Water Supply & Drainage Board and those totally dependent on groundwater for drinking. As a result of the main objective of this research project to consumers that is accomplished.

5. REFERENCES

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

6. APPENDICES

Model training

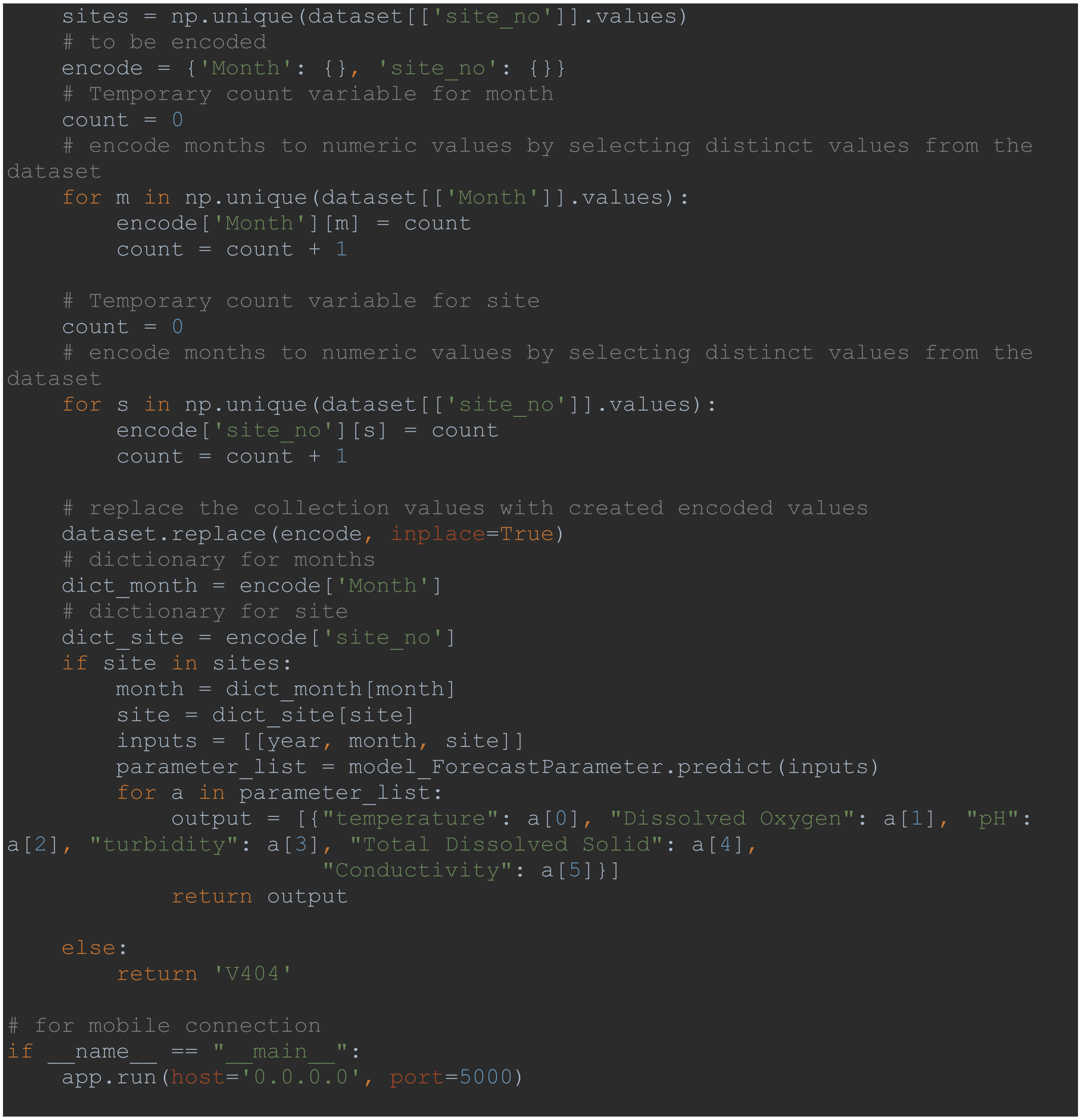
```
from pandas import read_csv
import numpy as np
from sklearn.ensemble import RandomForestRegressor
# Using Skicit-learn to split data into training and testing sets
from sklearn.model_selection import train_test_split
# Import the model we are using
import pickle


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

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

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

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

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


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

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

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


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

# Saving label names for later use
labels_list = list(labels.columns)
```

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

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

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

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

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

random_state=42)

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

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

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

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

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

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

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

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

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

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

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

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


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


    else:
        return 'We will be reach soon'


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

app.py

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



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

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

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


# predicting water quality parameter for a given month of a year
def predictWaterQualityParameterForAMonth(year, month, site):
    dataset = read_csv('datasets/dataset_forecastparameter.csv', index_col=0)
    output = {"temperature": 0, "Dissolved Oxygen": 1, "pH": 2, "turbidity":
3, "Total Dissolved Solid": 4, "Conductivity": 5}
```



```
    sites = np.unique(dataset[['site_no']].values)
    # to be encoded
    encode = {'Month': {}, 'site_no': {}}
    # Temporary count variable for month
    count = 0
    # encode months to numeric values by selecting distinct values from the
dataset
    for m in np.unique(dataset[['Month']].values):
        encode['Month'][m] = count
        count = count + 1

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

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

    else:
        return 'V404'

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