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

Final (Draft) Report

P.M.S.S.B. Gunarathna
IT16168800

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

P.M.S.S.B. Gunarathna

IT16168800

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

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

Name	Student ID	Signature
P.M.S.S.B. Gunarathna	IT16168800	

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

…………………………………….. ……………………..

Signature of the Supervisor: Date:

ACKNOWLEDGEMENT

I might want to accept this open door to stretch out my appreciation to the individuals who have helped me through this research. Initially, I thank my supervisor Mrs.Chathurangika Kahandawaarachchi and co-supervisor Ms.Narmadha Gamage for the patient guidance, consolation and advice they gave me over the span of this research. I would also like thank our lecture in charge Mr.Janaka Wijekoon and CDAP team for providing the guidance to reach the successful ending of the project.

Furthermore, my much gratitude goes to National Water Supply and Drainage Board and National Building Research Organisation of Sri lanka, National Hospital of Anuradhapura for the direction, advices and providing the required datasets for the study.

Much obliged to you my companions, for your important proposals and direction in the exploration. It truly made my life simpler during the end phases of my research. At long last, on account of everybody in my family who energized me during this Academic four years.

Keywords: Water quality, SVM, water parameters, Water Quality Index (WQI), CKD, CKDu

ABSTRACT

Smart solutions for water quality detecting are picking up significance with headway in communication innovation.21.4% of population is Sri Lanka satisfy their domestic water supply needs from groundwater assets. These water resources are spent without checking the safety of the water and they can be face to major outbreaks in future. Unawareness of the safety and quality of the water that consumes can leads to major health problems. Chronical kidney diseases and water borne diseases has been a major health concern in Sri Lanka that causes due to consumption of water that does not meet the water quality standards. There is less amount of solutions to identify the quality and the risk of exposure to CKDu or water borne diseases. The current solution for the issue is to propose a real time and user-friendly method to detect the quality of water and predict the risk of exposure to CKDu and water borne diseases. "E-tongue" the proposed smart device is implemented to detect the quality of the water in real time using cloud platform and predict the risk of CKDu and water borne diseases using machine learning approach. The end results are visualize using an android mobile application. The ultimate goal of the solution is to beware the people who are consuming the groundwater about its safeness, quality and risk of health concerns.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CKDu	Chronic Kidney Disease of unknown aetiology
PH	potential of hydrogen
TDS	Total Dissolved Solids
CKD	Chronic Kidney Disease
ANOVA	ANalysis Of VAriance
SVM	Support Vector machine
EC	Electric Conductivity
AWS	Amazon Web Services
API	Application Programmers Interface
BOD	Biological Oxygen Demand
WQI	Water Quality Index
ML	Machine Learning
IoT	Internet of Things
WHO	World Health Organization

1. INTRODUCTION

Water is one of key component that is discovered sufficient in nature and is imperative to living kind, but it is elusive a decent source that is alright for utilization, particularly for humanity. The nature of water is a general depiction of the organic, substance and physical qualities of water regarding proposed utilizes and a lot of guidelines. Consequently the assessment of nature of water can be characterized as the evaluation of the organic, substance and physical properties of water in agreement to characteristic quality and planned employments.

Ground water is one of the major water resources that is extensively used in Sri Lanka for domestic, industrial, commercial and other purposes. Among 21.44 million population of Sri Lanka 5.3 million people, in percentage of 21.4 meet their rural domestic water supply needs from tube wells and dug wells by accessing ground water.

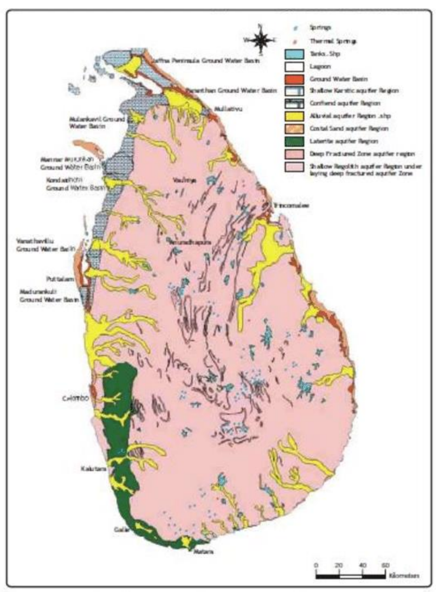


Figure 1.1.Different types of aquifers in sri lanka

Through the national water supply and drainage board and water asset specialists, Sri Lankan government has taken walks in satisfying the necessities of the individuals to ensure the water is provided to however much as the network as could reasonably be expected, major percentage of citizens still do not have access to the safe water coverage.

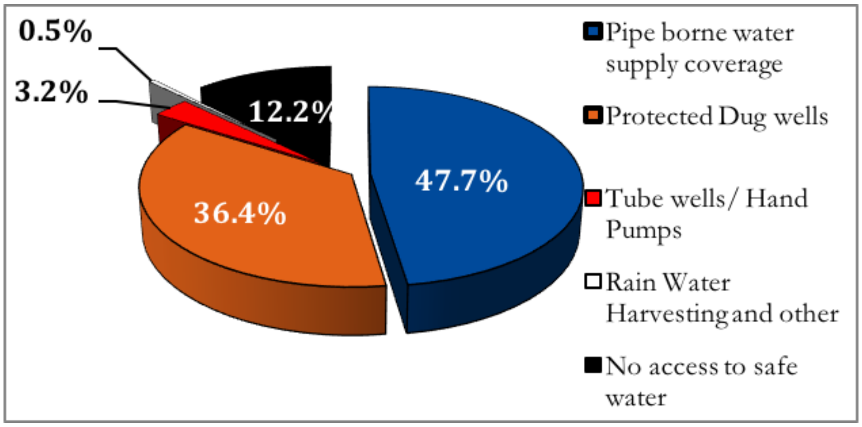


Figure 1.2.Access to safe water coverage in sri lanka

Uncontrolled use of ground water and natural poor quality or contamination can lead to major health concerns such as kidney failures, cardio vascular disease, dental fluorosis and CKDu. Unawareness of quality of the consuming ground water is one of the major causes for these scenarios. Chronic Kidney Disease of unknown aetiology (CKDu) is one of the greatest problems in north central province (NCP) which recorded highest mortality and morbidity rates and recognized reasons were unknown. Although there are solutions to determine the quality of ground water sample, those solutions need a period of time to deliver the results. Therefore, a solution that will help to overcome the challenge of determining water quality real time is demanded.

After thorough background study on the it is decided to develop a smart tool that can predict the safe consumption of the ground water consists with IoT based device and an mobile application.

1.1.Background & Literature Survey

One of the major organs that helps human body to function in a correct way is kidneys. It plays out the capacity of cleansing the blood. Poisons and fluid squanders that were found in blood is separated and emptied out of the body by the assistance of kidneys. Among the illnesses that are identified with kidneys, Chronic Kidney Disease (CKD) is one normal, yet more perilous malady which has been destroying lives of numerous individuals around the globe.

Over the most recent twenty years, ceaseless kidney infection of obscure etiology (CKDu) has developed as a critical supporter of the weight of constant kidney malady (CKD) in rustic Sri Lanka. It is described by the nonappearance of distinguished reasons for CKD. The commonness of CKDu is 15.1–22.9% in some Sri Lankan regions [1].

Sri Lanka has a populace of roughly 20 million and farming is a significant part of the economy. Chronic kidney disease (CKD) is a significant weight on the medical care arrangement of Sri Lanka. Diabetes, hypertension, and the different types of glomerulonephritis are very much perceived etiologies. With expanding predominance of non-transferable maladies, specifically diabetes and hypertension, the weight of CKD is relied upon to rise.5 Since the 1990s, another CKD, where no undeniable reason is recognizable, has been depicted in Sri Lanka. This new condition has brought about an ascent in the frequency of CKD in rustic Sri Lanka and has been suitably named ceaseless kidney malady of obscure etiology (CKDu). Alternate proposed names for this condition incorporate chronic agricultural nephropathy (CAN) and CKD of multifactorial origin (CKD-mfo). Similar substances with obscure reason for CKD exist in different pieces of the world. Ecological operators and conditions, for example, substantial metals and modern synthetic concoctions, have been connected to the advancement of CKDu in different pieces of the world.

The causes for the CKDu is still unknown. Multiple factors are assumed to be the causes for the CKD. Studies discovered that despite the fact that CKDu is brought about by many recognized components, the spots which has high presence of CKDu is fundamentally because of the utilization of tainted water. For the spreading of CKDu in areas like Anuradhapura and Polonnaruwa agricultural and metal squanders have been recognized as the significant reasons since the individuals are generally relying upon ground water sources which has direct drainage from from all these previously mentioned defilements.

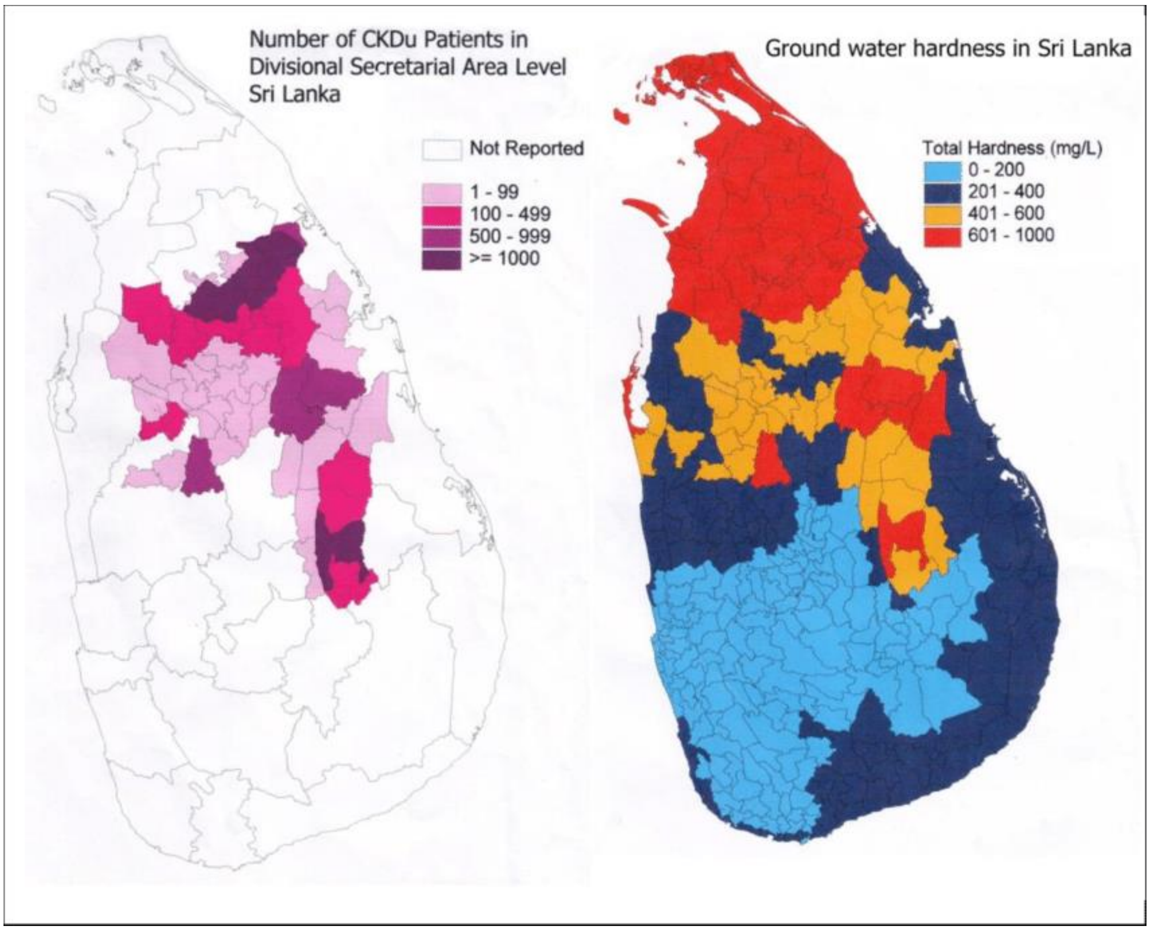


Figure 1.4.1.1.geographical distribution of CKDu patients and ground water

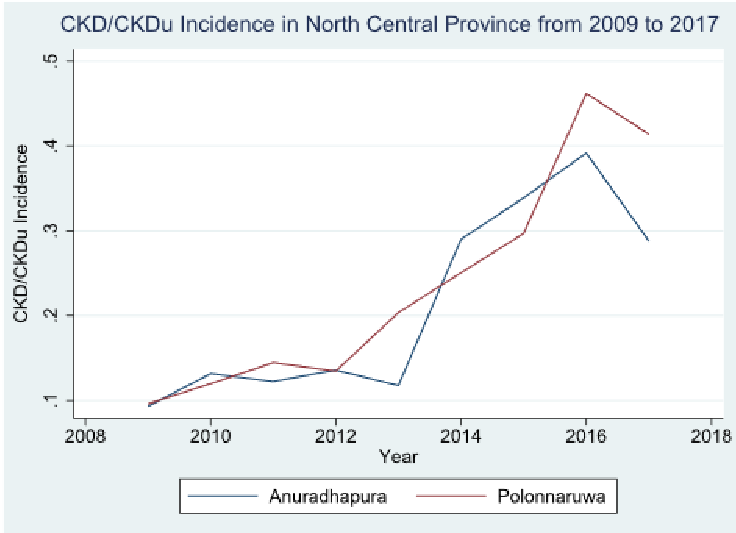


Figure 1.4.1.2.CKDu distribution of Ampara

The research was carried on Ulagalla cascade in Anuradhapura. According to the study it suggested that cumulative levels of heavy metals may aggravating the CKDu. Study that carried on Giradurukotte, Sri Lanka [2] thirty-two water samples were selected representing CKDu prevalent and non-prevalent communities and experimented the water quality parameter such as PH, Electrical conductivity, TDS. Based on measured water quality parameters twenty five percent of ground water bodies were identified as doubtful, whereas all the natural surface water bodies were identified as suitable for drinking purposes.

Higher TDS content was recorded in CKDu non-prevalent areas compared to prevalent areas. Also, a significant difference ($p<0.05$) was observed in groundwater and surface water for TDS and higher TDS values were recorded in groundwater. Average fluoride content of shallow wells and surface water bodies in both areas were varied from 0.29 mg/l to 1.36 mg/l and 0.17 mg/l to 0.88 mg/l respectively. Based on results of ANOVA, a significant difference ($p>0.05$) was not observed in CKDu prevalent and non-prevalent areas. But a significant difference ($p<0.05$) was observed in drinking water sources while, groundwater recorded the higher fluoride content and high calcium concentrations were recorded in groundwater bodies located in CKDu non-prevalent areas.[3]

Prof. Fredrik Winquist from the department of applied physics, Linkoping University, Sweden is the inventor of the voltammetry based electronic tongue concept voltammetry based electronic tongue concept[4]. By embracing the concept of E-tongue significant amount of research project have conducted such as analytical gustatory tool, wine discrimination , classification of tea samples with usage of complex mathematical models.

S.Revathy and collogues has carried out research to predict Chronic Kidney Disease using machine learning algorithms. According to the study multiple classification algorithms were trained to obtain the result. Random Forest Classifier and Support Vector Machines algorithms are being the main algorithms that were trained and the algorithm that obtained the highest accuracy is Random Forest Classifier with 99% of accuracy.[7]

The research carried out by Ross.S.Kleman and members on using machine learning algorithms to predict risk for development of Calciphylaxis in patients with CDK has found that modeling calciphylaxis risk with random forests learned from binary feature data produces strong models, and in the case of predicting calciphylaxis development among stage 4 CKD patients.[8]

The study conducted by W.G.S.D.Gunarathna, K.D.M. Perera and K.A.D.C.P. Kahandawaarachci on Performance Evaluation on Machine Learning Classification Techniques for Disease Classification and Forecasting through Data Analytics for Chronic Kidney Disease (CKD), they have used machine learning classification algorithms. Classification models have been worked with various order calculations will foresee the CKD and non-CKD status of the patient. These models have applied on as of late gathered CKD dataset downloaded from the UCI storehouse with 400 information records and 25 credits. Consequences of various models are thought about. From the correlation it has been seen that the model with Multiclass Decision woodland calculation performed best with a precision of 99.1%.[9]

1.2.Research Gap

As it explained in the background and literature survey, numerous researches have been conducted in the domain of CKDu. Based on the area of research we can categorize as the research studies that conducted to find the cause of the CKDu and the research studies that conducted to predict the CKDu patients with existing bio medical data. And also, there are various water quality measuring tools available in the industry. But there is not satisfied solution that can predict the risk of CKDu based on the water quality.

Prior to beginning to actualize the development of this solution, the need of profound examination of significant frameworks or items that were in the market is required. Executing another application that has similar highlights, will be considered as a misuse of cash and furthermore tedious. By dissecting the applications which are now accessible at the market, it is the most significant to chop down the remaining task at hand.

Table 1.4.1.1.Comparison of Existing Products

Features	**Existing Research/products**				
	TDS	Irrigation water quality calculator	Zen-Test	Water quality monitoring for rural areas	E-tongue
Basic sensors of water quality parameters	✓		✓	✓	✓
Get real time values from sensors	✓		✓	✓	✓

Use wi-fi to upload data to cloud					✓
Analyze past data to give solution				✓	✓
Display sensor values real time			✓		✓
Real time prediction of future water quality					✓
Cloud services to store sensor data and obtain results				✓	✓
Real time prediction for risk of exposure to CKDu					✓

Above mentioned details created major research gap that opens path to develop a smart device that has the functionality of predicting possible risk of exposing to CKDu using machine learning algorithms.

1.3.Research Problem

This section elaborates the research problem and specific research questions that this individual component addressed. After thorough study of above-mentioned research gap, research problem of this component can address as follow.

- How to predict possible risk of exposing to Chronical Kidney Disease of unknown etiology using machine learning algorithms.

The specific research questions that has to address while resolving the above-mentioned main research problem has listed below.

- How to find the connection between CKDu and water quality parameters.
- How to find the dataset that fulfills the requirements
- How to create the machine learning model and train the model
- How to select the best algorithm for prediction
- How to deploy trained machine model
- How to visualize predicted outcome to end user

1.4.Objectives

Following sub sections discuss about the main objective and the specific objectives that need to be achieved for the completion of the main objective.

1.4.1. Main Objective

The main objective of this individual component is to predict the risk of exposure to CKDu by consuming the given water sample resource for a longer time period.

1.4.2. Specific Objectives

- Identify the dataset that will be used in machine learning model.
- Identify the most suitable algorithm I order to produce the results.
- Test the mathematical model with dataset that has known results in order to obtain the accuracy.
- Implement data streaming pipelines to ingest the data to mathematical model and obtain results.
- Develop an android mobile application and web application in order to visualize the outputs.

2. METHODOLOGY

In this section the methodology of the component has elaborated in a comprehensive approach.

2.1. System Overview

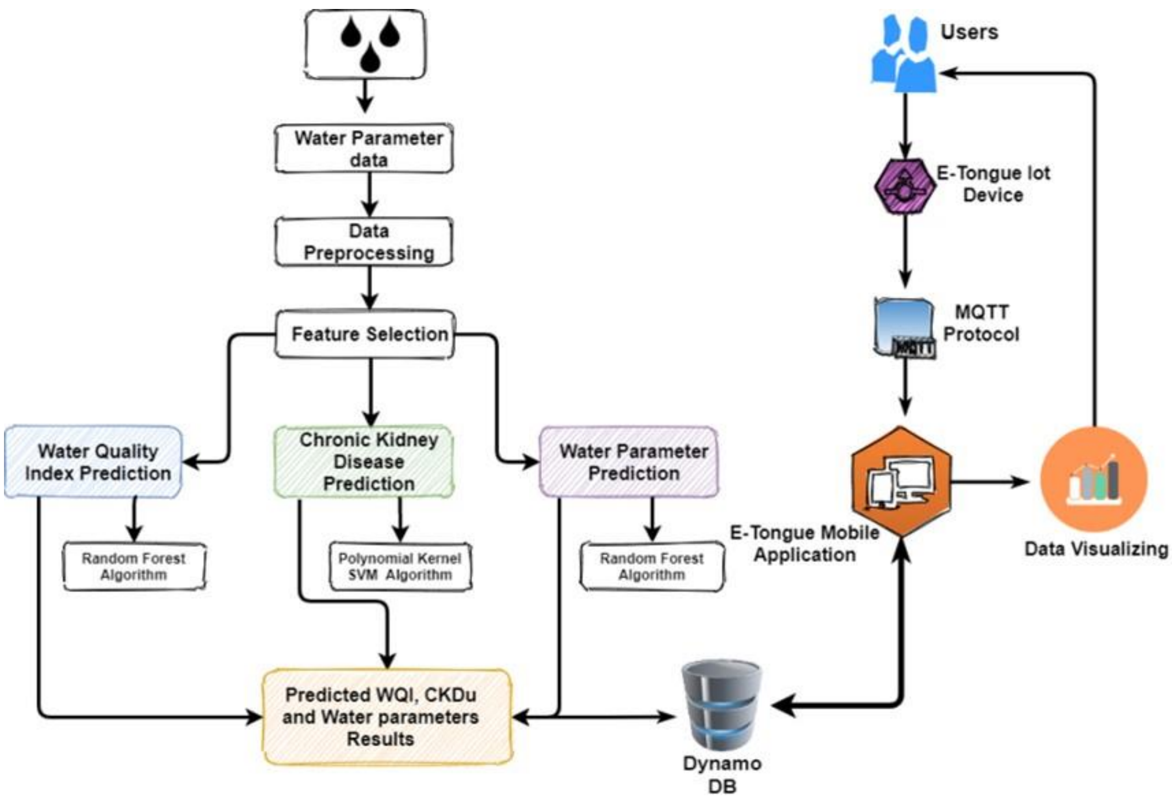


Figure 1.4.2.1.System Overview Diagram

As it mentioned in the system overview diagram E-tongue consists with three main components.

- IoT based device to collect water quality parameter values of a given water sample
- Backend service which generates the predictions according to the collected water sample and stores results to the database
- Android mobile application which visualize the generated outputs and collected senor values to the users.

IoT based device gathers the sensor data from a given water sample for 3 minutes with 12 seconds intervals. The collected sensor data is stored in the AWS Dynamo DB instance. From the backend sensor data that is specific to the respective device id is retrieved. The average of the retrieved data will be feed to the models for the prediction.

From the backend services of the system it provides the outputs for the main functionalities of the system.

- Identify the water quality of the given water sample by predicting the WQI value.
- Predict the possible risk of exposure to CKDu based on the location and the water quality.
- Forecasting water quality parameters.

The prediction processes of the main functionalities are concluded using supervised learning techniques which are used on machine learning. The end result of the predictions can be accessed by the mobile application which is connected with the backend services.

2.2. Resources Needed

The specific datasets were required for the prediction algorithms. Therefore, in order to predict risk of CKDu dataset of water quality parameter values of different location of North Central province in Sri Lanka with respective location details is collected form the Ground Water Department, Ratmalana, Sri Lanka. The dataset of the patient location details is obtained from the National Hospital of Anuradhapura.

2.3. Flow of Project

2.3.1. Component Overview

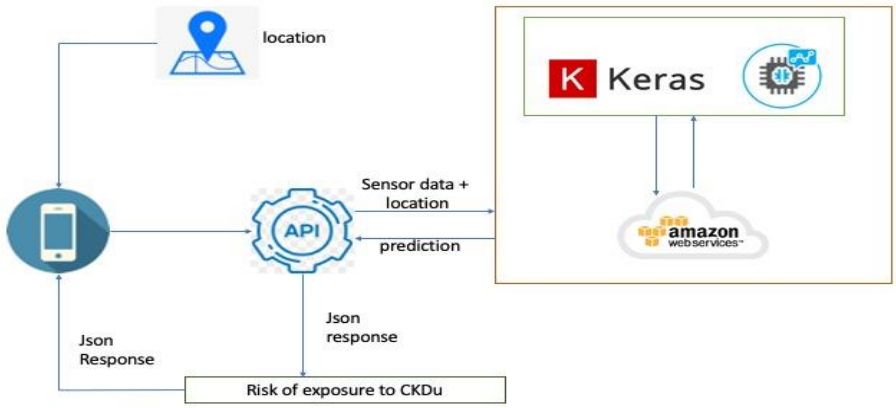


Figure 2.3.1.1.Component overview diagram

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

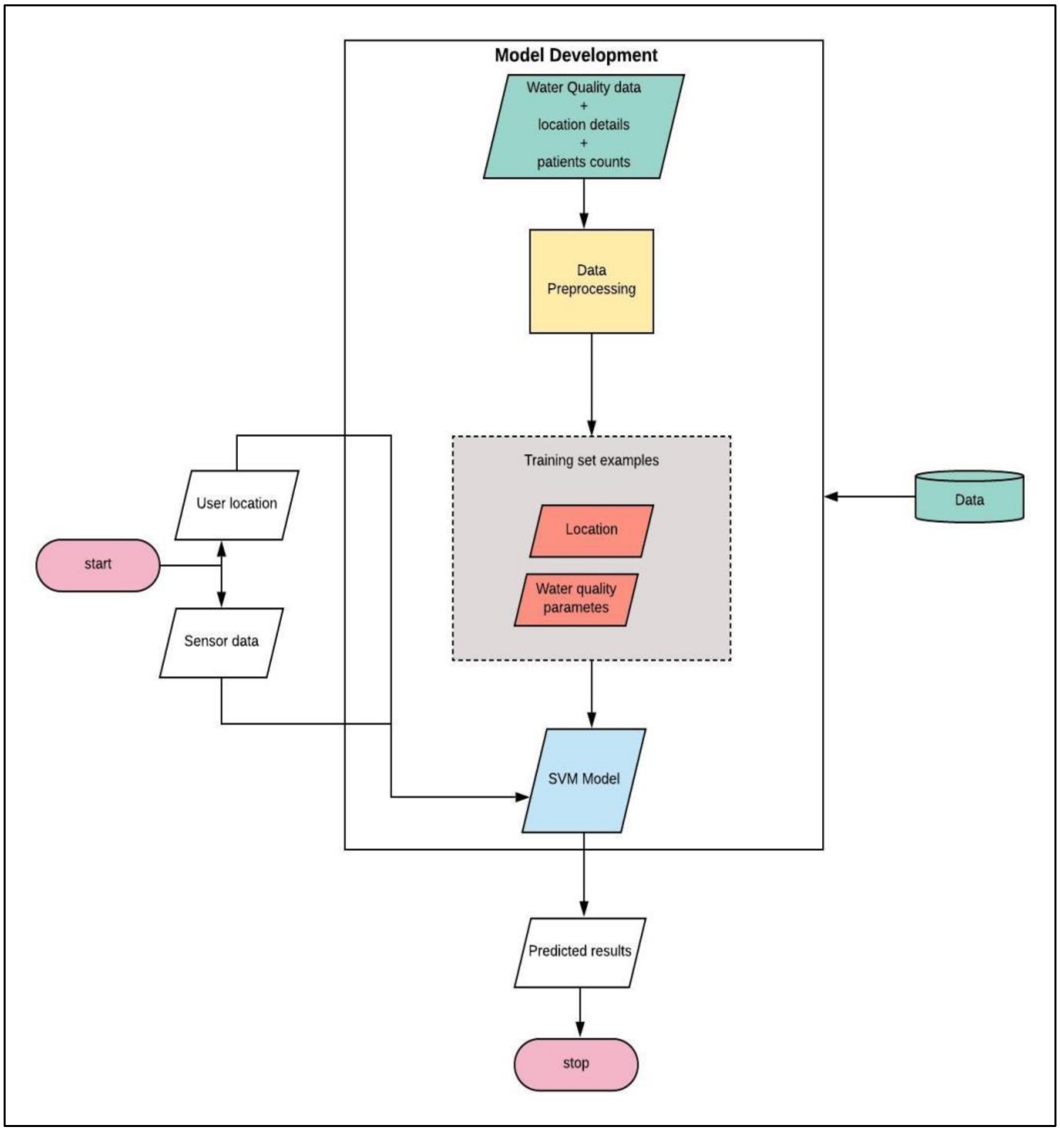


Figure 2.3.1.2.Flow diagram of the component

2.3.2. Data gathering for model training process.

The dataset that is utilized for model preparing measure assumes critical function on model training process. For machine learning models to understand how to perform various actions, training datasets must first be fed into the machine learning algorithm, followed by validation datasets (or testing datasets) to ensure that the model is interpreting this data accurately.

For the prediction process of CKDu, the required dataset of water quality parameters with the location details is obtained from the National Water Supply and Drainage Board head office located in Ratmalana and the location of the discovered CKDu patients are obtained from the National Hospital of Anuradhapura.

2.3.3. Model training using classification algorithms.

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

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

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

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

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

Where d is the degree of the polynomial.

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

To predict the risk of CKDu the sensor data needs to be feed to the machine learning model. In order to retrieve the sensor data from the AWS dynamo DB instance the RESTful web API is developed using Node JS as a programming language.
In order to get the predicted results of the machine learning model and to feed the data to the model API is created using Flask web framework for python. Both of the API services are accessible through mobile application using the Retrofit client.

2.3.5. Develop mobile application to visualize the results.

As end user access the mobile application to view the results implementation of the mobile application plays the crucial part in the methodology. Wireframes were created as initial step of the mobile application implementation.

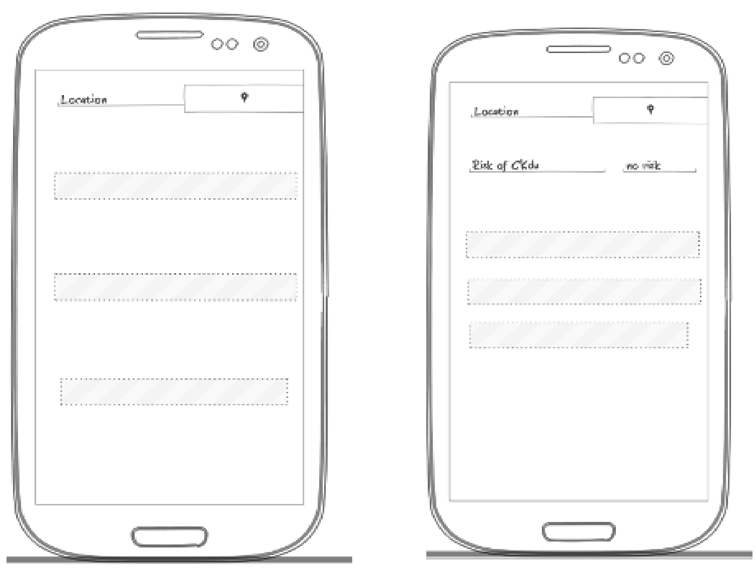


Figure 2.3.5.1.Wireframes of Mobile part1

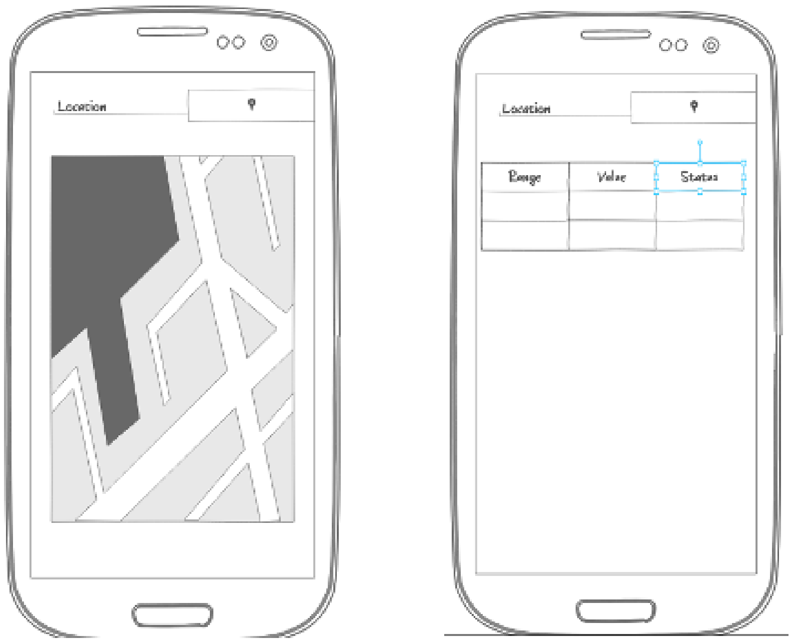


Figure 2.3.5.2.Wireframes of Mobile part2

Flow of the mobile application for this component can be concluded by the following steps.

- Connect the mobile application with device using special device id.
- Once device get the reading user can access the output.
- User has to select the location before choosing the predict the risk of CKDu.
- Once the results obtained from the backend servers the results will be displayed with the values of the sensor data.
- If parameter values are not in the correct range user is notified.

2.4. Commercialization Aspects of the Product

"E-tongue" is a strongly suggested framework for individuals who are in the dry zone territory, and they battle to know the protected utilization of the groundwater. The fundamental objective of the research study is to full fill the hole between the individuals and the National Water Supply and Drainage Board of Sri Lanka. Since there are a few awarenesses were directed with respect to the significance of safe utilization of the groundwater.

CKDu is has become weighty burden to Sri Lanka. The foundations for the CKD are not all that conspicuous to distinguish. As it talked about in the writing overview numerous factor are thought to be the reason for the CKDu and water quality being one of them. However, there are no devices that could distinguish the danger of CKDu dependent on the water quality. E-tongue facilitates the function of predicting the possible risk of exposing to CKDu in a given water sample and location. This solution can be applied to the both industrial and individual user to identify the quality of the water. And it can also be used on following scenarios.

- measuring water quality of new water sources.
- measure the water quality of the existing water resources with change of time.

2.5. Testing & Implementation

2.5.1. Implementation

2.5.1.1.Model development

For prediction of the CKDu the machine learning model need to be developed. The life cycle of the machine learning model development is shown in Figure 2.5.1.1.

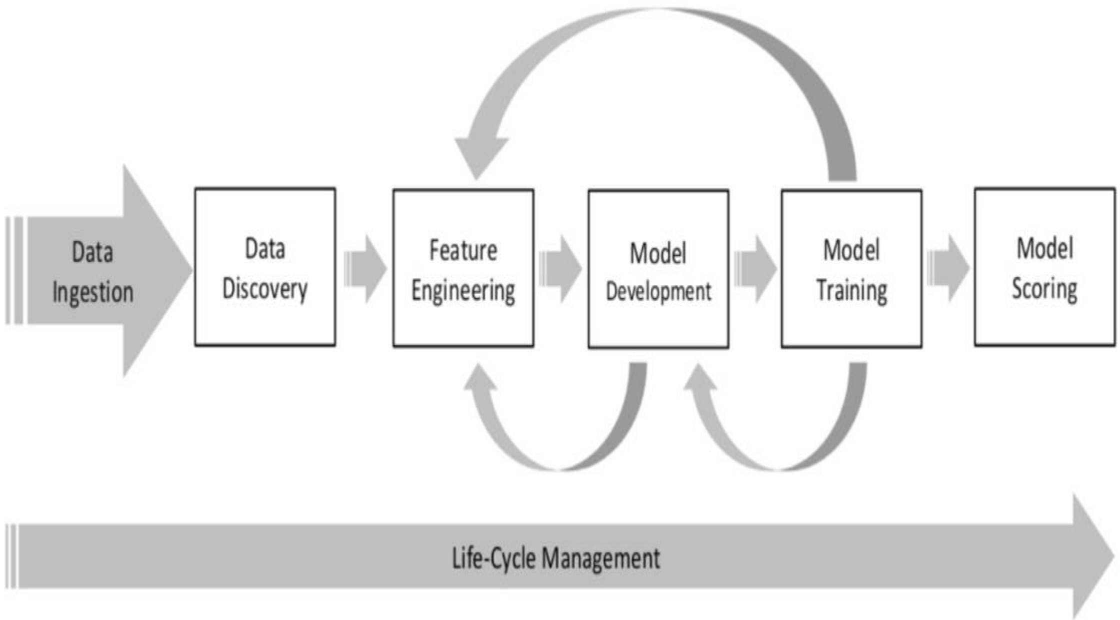


Figure 2.5.1.1.Life Cycle of model training

Feature engineering

All the machine learning algorithms use set of input data to create outputs or predictions. Input data are compromise features which are usually in the form of structured columns. But algorithms require features with some specific characteristics to work in a proper manner. Feature engineering offer following options on the datasets.

- Preparing the proper input dataset which is compatible with the machine learning algorithm requirements.
- Improve the performance of the using machine learning model

	Year	Month	temperature	dissolved_oxygen	pH	turbidity	hardness	patients	status	tds	ec
count	1643.000000	1643.000000	1643.000000	1643.000000	1643.000000	1643.000000	1643.000000	1643.000000	1643.000000	1643.000000	1643.000000
mean	2016.452830	6.566038	22.879805	8.027959	7.048703	28.234180	118.983898	3.295800	0.245283	205.463169	0.235942
std	1.311686	3.596214	6.916625	1.994721	0.256538	28.761188	34.280067	4.911534	0.430386	85.747693	0.065843
min	2014.000000	1.000000	4.650029	2.238878	5.057236	0.774541	60.035979	0.000000	0.000000	55.146489	0.120075
25%	2015.000000	3.000000	17.766066	6.956526	6.898199	11.067392	88.885932	0.000000	0.000000	132.995039	0.180201
50%	2016.000000	7.000000	25.518597	8.028939	7.047260	17.943200	119.667195	0.000000	0.000000	206.508660	0.237581
75%	2018.000000	10.000000	28.781889	9.591593	7.191177	34.787758	147.485759	6.000000	0.000000	276.989056	0.291732
max	2019.000000	12.000000	30.764247	12.638194	7.994668	226.497250	179.846586	15.000000	1.000000	354.536870	0.349956

Figure 2.5.1.2.Dataset after Feature engineering

Figure 2.5.1.3 shows the dataset after it gone through with the feature engineering.Andthe Figure 2.5.1.4 shows the correlation of the parameters.

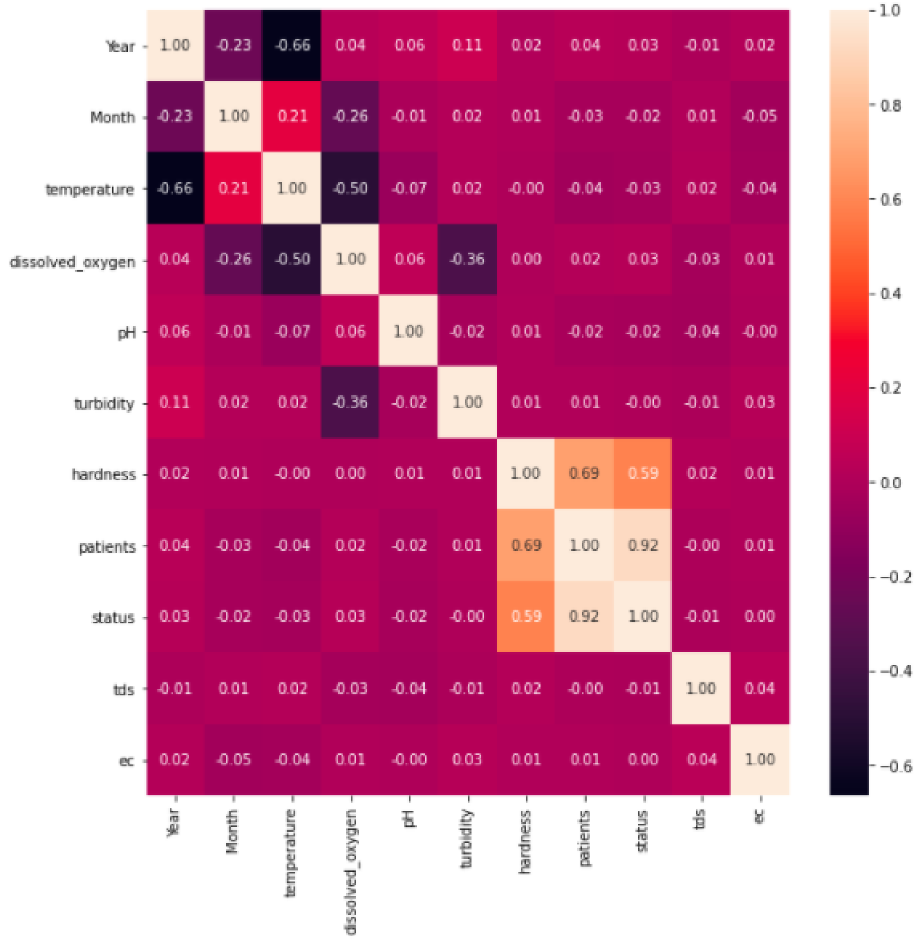


Figure 2.5.1.5.Correlation of parameters

Model training and building

The key component of the implementation process is the creation of the machine learning model and train it. Since it needed to find the algorithm that gives the best performance dataset is trained with the number of classification algorithms. As the standard way of the machine learning model training process dataset is divided into two main parts before model training.

1. Training set

   Train and tune the model (using cross validations)

2. Test set
    - Excluded from training set
    - Validate the model build

```
# Splitting the data into sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
print('Train Features :', X_train.shape)
print('Train Labels:', y_train.shape)
print('Test  Features:', X_test.shape)
print('Test  Labels:', y_test.shape)
```

```
Train Features : (1314, 7)
Train Labels: (1314,)
Test  Features: (329, 7)
Test  Labels: (329,)
```

Figure 2.5.1.6.Slit dataset

Above Figure2.5.1.1.6 shows the dividing process of the dataset.

The divided dataset is trained against the following classification machine learning algorithms with following petameters as input parameters and risk of CKDu as output.

- Ph
- Temperature
- Turbidity
- TDS
- EC
- Location

(a)Random forest classifier

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Figure 2.5.1.1.7 shows the process of the Random forest classifier.

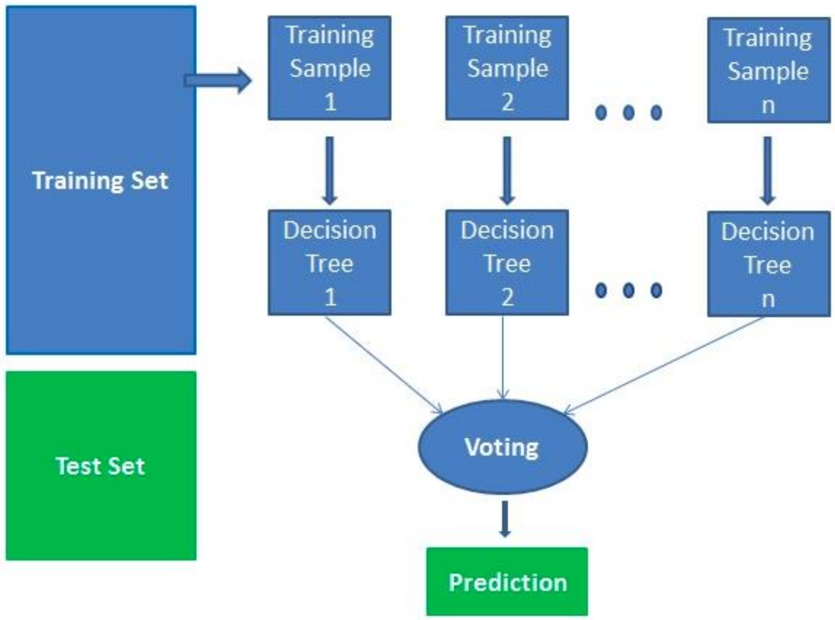


Figure 2.5.1.7.Random forest classifier

The dataset is trained against the random forest classifier to predict the risk of CKDu.

```
# Splitting the data into sets
x_trains, x_tests, y_trains, y_tests = train_test_split(data, labels, test_size = 0.2, random_state = 40)
print('Train Features :', x_trains.shape)
print('Train Labels:', y_trains.shape)
print('Test  Features:', x_tests.shape)
print('Test  Labels:', y_tests.shape)
model = RandomForestClassifier(n_estimators=1000,max_features=8,bootstrap=True,criterion='entropy',n_jobs=-1)
model.fit(x_trains, y_trains)
```

```
Train Features : (985, 8)
Train Labels: (985,)
Test  Features: (658, 8)
Test  Labels: (658,)
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='entropy', max_depth=None, max_features=8,
                       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=-1, oob_score=False, random_state=None, verbose=0,
                       warm_start=False)
```

Figure 2.5.1.8.Model trainingpart1

```
model = RandomForestClassifier(n_estimators=1000,max_features=8,bootstrap=True,criterion='entropy',n_jobs=-1)
model.fit(x_trains, y_trains)
```

```
#Predict test values
yprediction = model.predict(x_tests)
```

```
from sklearn import metrics
print("Classification Report of Risk of water borne disease : \n\n",metrics.classification_report(yprediction, y_tests,
target_names = ["No Risk","Risk"]))
```

```
Classification Report of Risk of water borne disease :

              precision    recall  f1-score   support

     No Risk       0.46      0.49      0.47       303
        Risk       0.54      0.50      0.52       355

    accuracy                           0.50       658
   macro avg       0.50      0.50      0.50       658
weighted avg       0.50      0.50      0.50       658
```

Figure 2.5.1.9.Model training part2

(b)Polynomial Kernel SVM

Among Support Vector Machine that that represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models, Polynomial kernel SVM is widely used.

Intuitively, the polynomial kernel looks not only at the given features of input samples to determine their similarity, but also combinations of these. In the context of regression analysis, such combinations are known as interaction features. The (implicit) feature space of a polynomial kernel is equivalent to that of polynomial regression, but without the combinatorial blowup in the number of parameters to be learned. When the input features are binary-valued (Booleans), then the features correspond to logical conjunction of input features.

$$K(x, xi) = 1 + sum(x * x)^{d}$$

d = degree of the polynomial

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='poly', degree=8)
svclassifier.fit(X_train, y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=8, gamma='scale', kernel='poly',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

Figure 2.5.1.10.Polynomial Kernel SVM

(c)Sigmoid Kernel SVM

The sigmoid kernel was quite popular for support vector machines due to its origin from neural networks. Although it is known that the kernel matrix may not be positive semi-definite (PSD), other properties are not fully studied. In this paper, we discuss such non-PSD kernels through the viewpoint of separability. Results help to validate the possible use of non-PSD kernels. One example shows that the sigmoid kernel matrix is conditionally positive definite (CPD) in certain parameters and thus are valid kernels there.

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='sigmoid')
svclassifier.fit(X_train, y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='sigmoid',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

Figure 2.5.1.11.Sigmoid Kernel SVM

(d)Gaussian Kernel SVM

The Gaussian kernel is an example of radial basis function kernel.

$$k(x, y) = \exp\left(-\frac{||x - y||^2}{2\sigma^2}\right)$$

FIGURE 2.5.1.12

Alternatively, it could also be implemented using

$$k(x, y) = \exp\left(-\gamma||x - y||^2\right)$$

FIGURE 2.5.1.13

The adjustable parameter sigma plays a major role in the performance of the kernel, and should be carefully tuned to the problem at hand. If overestimated, the exponential will behave almost linearly and the higher-dimensional projection will start to lose its non-linear power. In the other hand, if underestimated, the function will lack regularization and the decision boundary will be highly sensitive to noise in training data.

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='rbf')
svclassifier.fit(X_train, y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

Figure 2.5.1.14.Gaussian Kernel SVM

Model validation

Model validation is the process where a trained model is evaluated with the testing dataset. The main objective of using the testing dataset for testing is to test the generalization ability of a trained model. Figure 2.5.1.15 shows the process of the model validation.

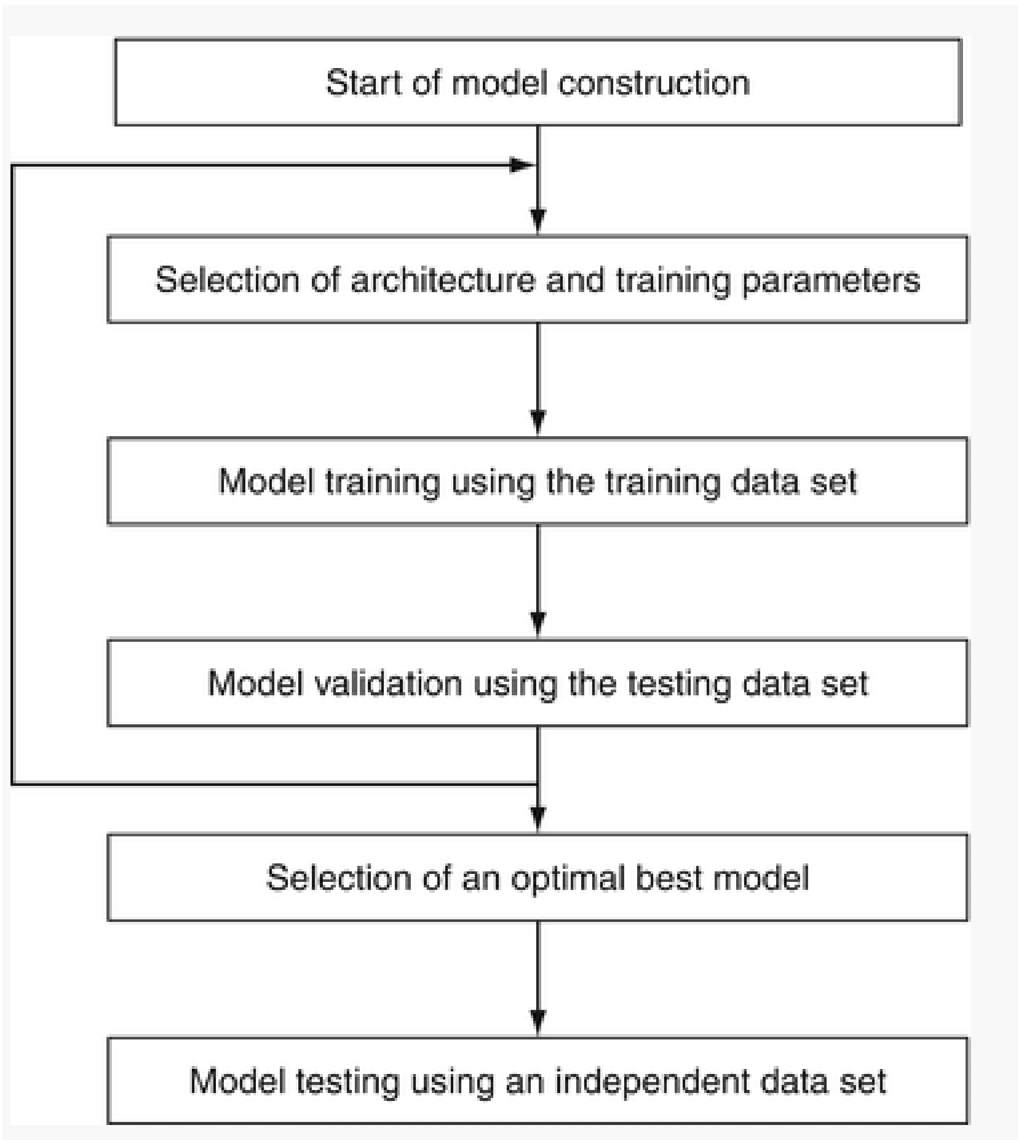


Figure 2.5.1.15.Model validation Process

As model validation methods following methods were used.

- Train and Test split
- K-fold cross-validation
- Nested cross-validation

2.1.1.1.Deploy the model and create Web API services to transfer data

Pickle is used for serializing and de-serializing Python object structures, also called marshalling or flattening. Serialization refers to the process of converting an object in memory to a byte stream that can be stored on disk or sent over a network. Later on, this character stream can then be retrieved and de-serialized back to a Python object. Pickle is used for loading the trained machine learning model.

The data transferring among the components mainly affects to the process of the solution. In order to retrieve the sensor data from the AWS dynamo DB instance RESTful web API is developed using Node JS as the programming language. To retrieve the predicted output of the model API service is created using the Flask web framework.

2.1.2. Testing

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

2.1.2.1.Front-end testing

Table 2.1.2.1.Test Case1

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 2.1.2.2.Test Case2

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

Table 2.1.2.3.Test case 3

Test case ID	003
Test case scenario	Show warning alert on water quality parameter values based on range

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

2.1.2.2.Back-end testing

Table 2.1.2.2.1Test case4

Test case ID	001
Test case scenario	Check retrieval of database sensor data values
Test steps	a. Start the Node JS backend server b. Open ‘Postman’ IDE c. Enter url and query parameters
Test data	URL = http://127.0.0.1:3000/parameters/getParameters

Expected result	Latest sensor data of the device should retrieve
Actual result	As expected
Pass/Fail	Pass

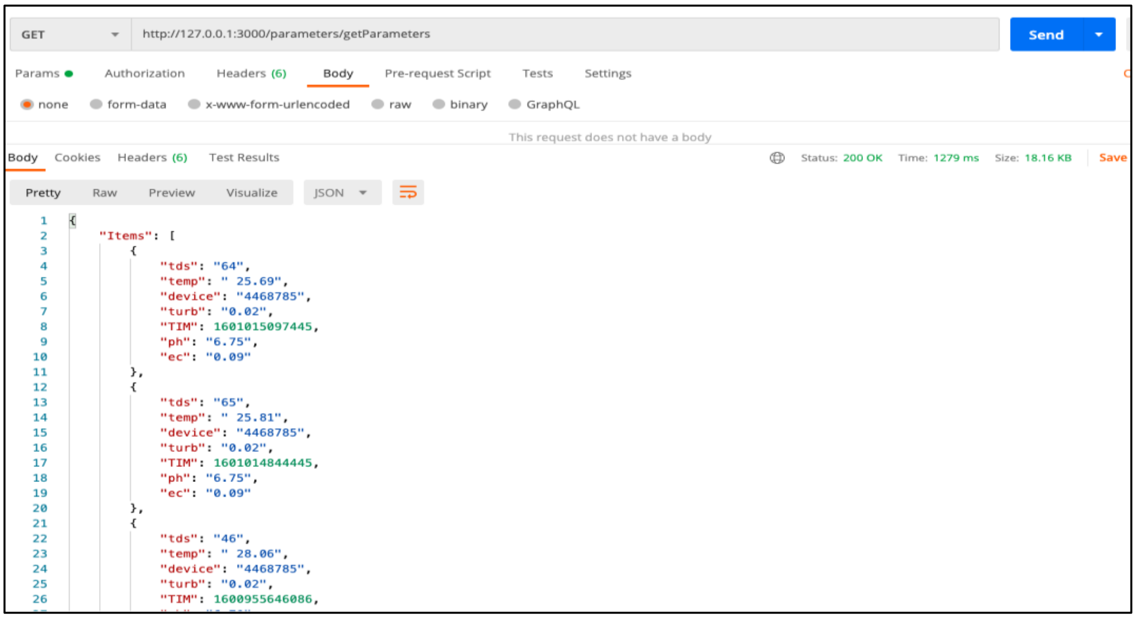


FIGURE2.1.2.2.BACKEND RESULT1

Table 2.1.2.2.2 Test case 5

Test case ID	002
Test case scenario	Check retrieval of predicted values.
Test steps	a. Start the python backend server b. Open ‘Postman’ IDE c. Enter url and query parameters

Test data	URL = http://127.0.0.1:3000/parameters/getParameters Temperature =27.36 Ph =6.94 Turbidity =14.15 TDS = 162.80 EC =0.2 Site = “Galnewa”
Expected result	Prediction for the given parameters should retrieve
Actual result	As expected
Pass/Fail	Pass

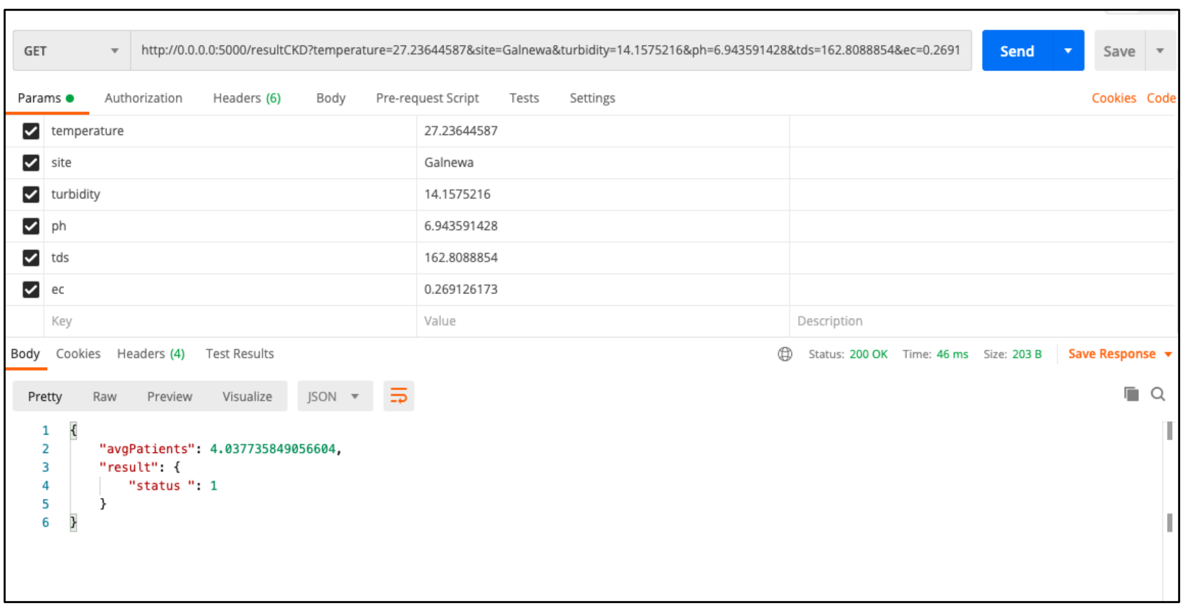


FIGURE2.1.2.3.BACKEND RESULTS2

3. RESULTS & DISCUSSION

3.1. Results

3.1.1. Results of each model

As it elaborated in the above section number of algorithms are trained for the risk prediction of CKDu. From the selected machine learning classification algorithms Polynomial Kernel Support Vector Machine algorithm was selected for the prediction. The selection process happened based on the accuracy of each model. The model with the highest accuracy and lowest average error has selected as the model for the prediction which is Polynomial SVM algorithm. Table 3.1.1.1 shows the evidence of the model selection process.

Table 3.1.1.1.Accuracy comparison

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

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='sigmoid',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

```
y_pred = svclassifier.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[  4 155]
 [  4 166]]
              precision    recall  f1-score   support

           0       0.50      0.03      0.05       159
           1       0.52      0.98      0.68       170

    accuracy                           0.52       329
   macro avg       0.51      0.50      0.36       329
weighted avg       0.51      0.52      0.37       329
```

```
from sklearn.metrics import accuracy_score
accuracy_rate = 100.0 * accuracy_score(y_test, y_pred)
print ("The test data accuracy : " + str(accuracy_rate))
```

```
The test data accuracy : 51.97568389057751
```

Figure 3.1.1.1.Accuracy evidence sigmoid

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='poly', degree=8)
svclassifier.fit(X_train, y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=8, gamma='scale', kernel='poly',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

```
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)
```

```
y_pred = svclassifier.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[ 35 124]
 [ 26 144]]
              precision    recall  f1-score   support

           0       0.57      0.22      0.32       159
           1       0.54      0.85      0.66       170

    accuracy                           0.76       329
   macro avg       0.76      0.75      0.23       329
weighted avg       0.76      0.74      0.23       329
```

```
from sklearn.metrics import accuracy_score
accuracy_rate = 100.0 * accuracy_score(y_test, y_pred)
print ("The test data accuracy : " + str(accuracy_rate))
```

```
The test data accuracy : 76.99
```

Figure 3.1.1.2,Accuracy evidence polynomial

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='rbf')
svclassifier.fit(X_train, y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

```
y_pred = svclassifier.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[ 33 119]
 [ 39 138]]
              precision    recall  f1-score   support

           0       0.46      0.22      0.29       152
           1       0.54      0.78      0.64       177

    accuracy                           0.52       329
   macro avg       0.50      0.50      0.47       329
weighted avg       0.50      0.52      0.48       329
```

```
from sklearn.metrics import accuracy_score
accuracy_rate = 100.0 * accuracy_score(y_test, y_pred)
print ("The test data accuracy : " + str(accuracy_rate))
```

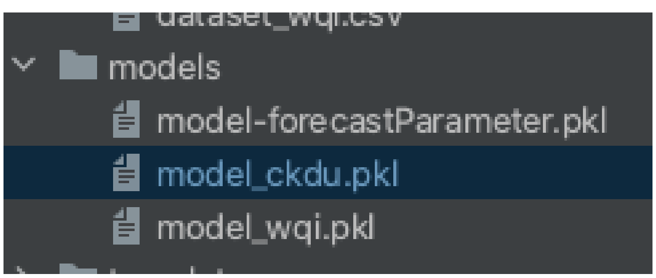
```
The test data accuracy : 51.97568389057751
```

Figure 3.1.1.3.Accuracy evidence Gaussian

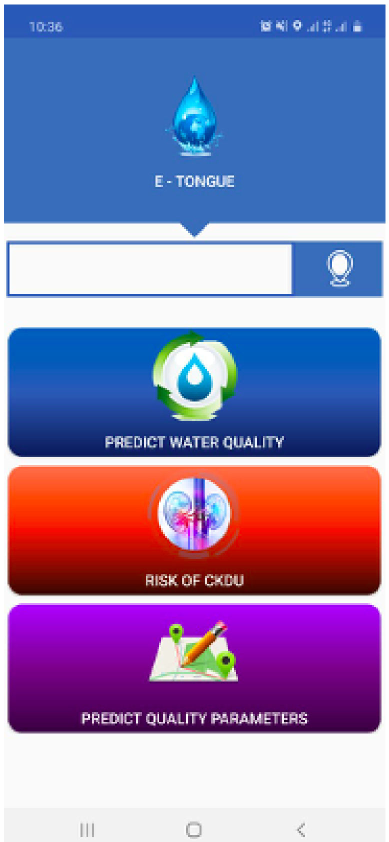
3.1.2. Model deployment

As it elaborated on the methodology, once the machine learning model is trained the model is loaded or export using pickle as “model_ckdu.pkl”.

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



3.1.3. User interface of mobile application



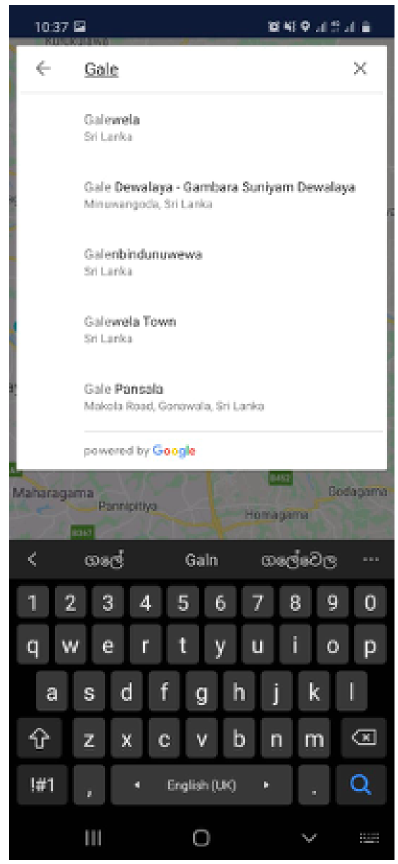


Figure 3.1.3.1.Mobile UI part1

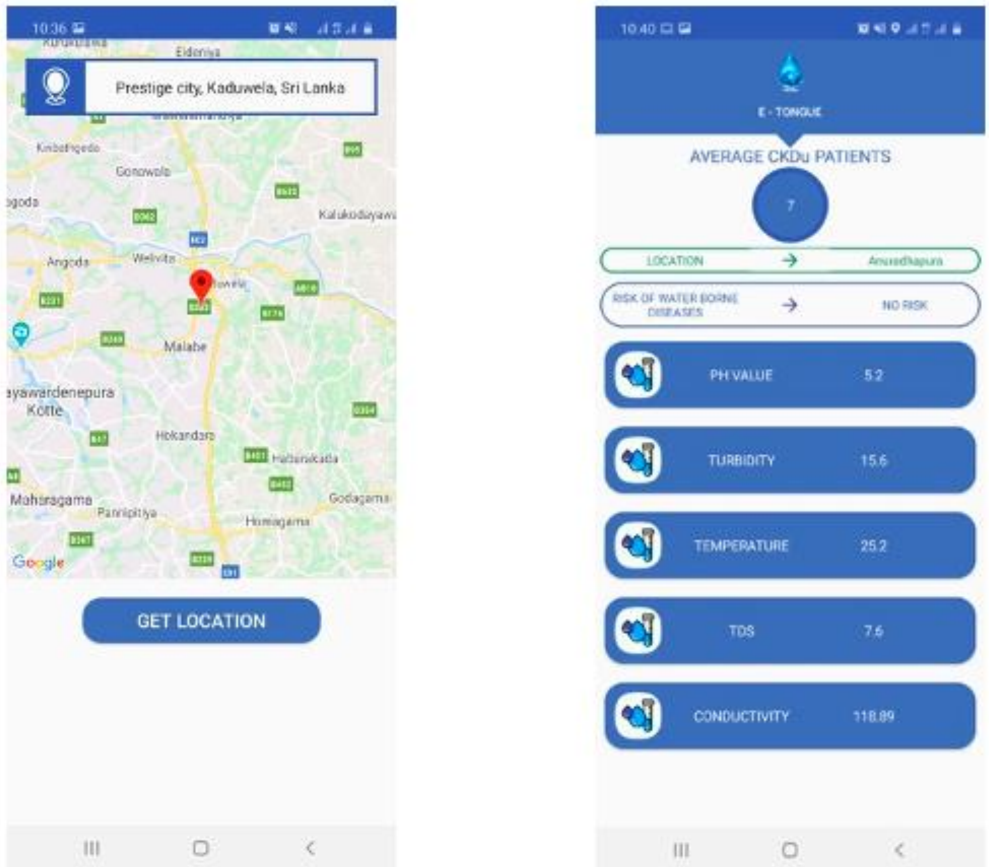


Figure 3.1.3.2.Mobile UI Part2

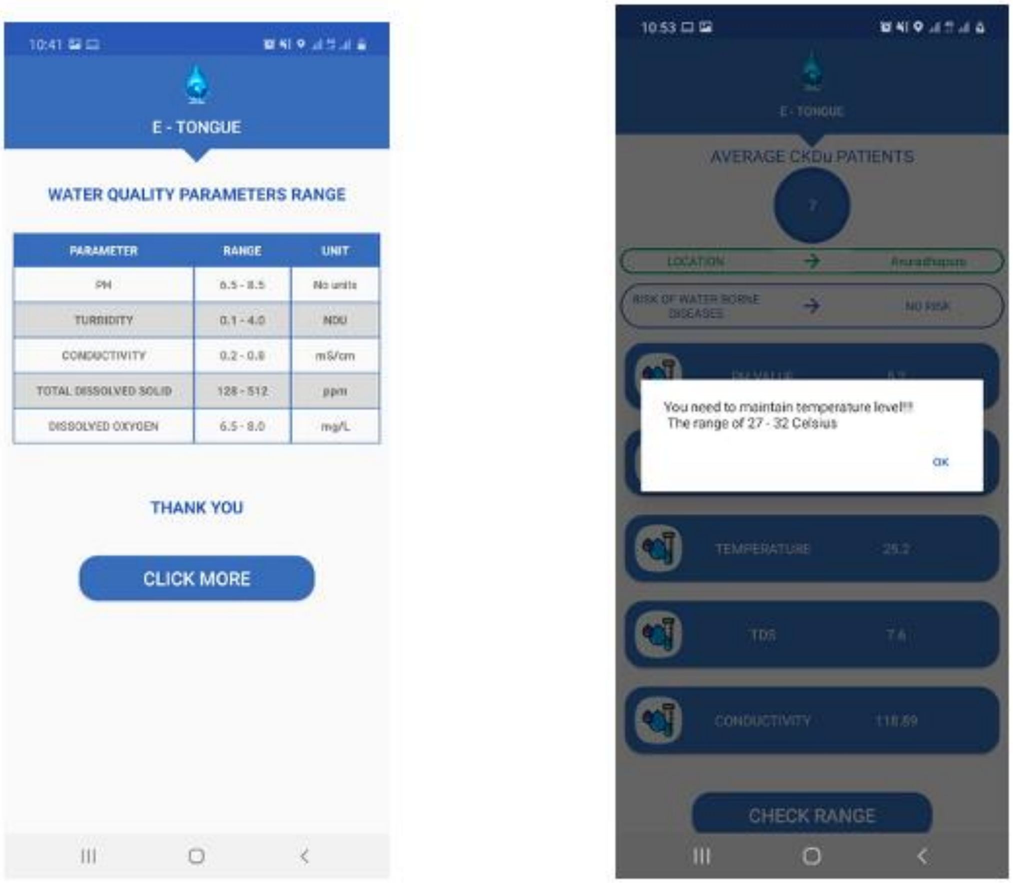


Figure 3.1.3.3.Mobile UI part3

3.2. Research Findings

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

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

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

3.3. Discussion

The main objective of this research study is to develop a smart tool that can predict the safe consumption of the ground water. It is further escalated into four main components being implement IoT based sensor device, predict the water quality based on WQI value, predict the possible risk of exposing to CKDu and forecast water quality parameters. The components apart from IoT based sensor device other three components have developed using machine learning techniques. This individual component focuses on the predicting the possible risk of exposure to CKDu.

CKDu is has become heavy burden to Sri Lanka. The causes for the CKD are not so obvious to identify. As it discussed in the literature survey multiple factor are assumed to be the cause of the CKDu and water quality being one of them. But there are no devices that could identify the risk of CKDu based on the water quality. Based on that research gap this individual component is implemented.

The desired outcome is gained by using the machine learning techniques. As the ML model training algorithm Polynomial Kernel SVM has used. And the outcome of the prediction is visualized to the end users using E-tongue mobile application. Following the difficulties that were followed during the implementation process.

- Due to COVID-19 pandemic situation gathering data was a difficult process.
- Difficulties in handling large datasets.
- For training process of the model, Polynomial Kernel SVM machine takes a more time comparing to the other algorithms.
- Integration difficulties with the mobile application and backend servers without hosting them.

4. CONCLUSION

This research study has presented the comprehensive approach to predict the possible risk of exposing to CKDu based on the water quality parameters on a given location. This cost-effective prototype developed to overcome the limitation of applications that can identify the risk of CKDu.

IoT based device gathers the sensor data from a given water sample. The gathered sensor data is transferred to the AWS Dynamo DB instance. Once the data gathering is over, from the mobile application it retrieves the sensor data from the database. The sensor data for one water sample reading contains multiple readings for 2 minutes with 12 seconds intervals. The average of each parameter is calculated from the front-end. Once sensor data average is calculated user can call the functionality predict CKDu. The mobile application fetches the user's current location using GPS and with the formatted sensor data that is retrieved from the device is sent to the Polynomial Kernel SVM algorithm that is deployed in the backend servers. The predicted results of the given water sample are generated and sent back to the mobile application. Prediction outcome is displayed in the mobile application with the values of the water quality parameter values. According to the standard ranges of each parameter user is notified if parameter in danger zone. The solution has obtained the accuracy of 76%.

Finally, as it elaborated in above sections this individual components gives the solution to predict the possible risk of exposing to CKDu using machine learning techniques.

REFERENCES

[1]S. Rajapakse, M. Shivanthan and M. Selvarajah, "Chronic kidney disease of unknown etiology in Sri Lanka", International Journal of Occupational and Environmental Health, vol. 22, no. 3, pp. 259-264, 2016.

[2]"Drinking Water Quality in Chronic Kidney Disease of Unknown Aetiology (CKDu) Prevalent and Non-prevalent Areas in Giradurukotte, Sri Lanka", International Journal of Advances in Agricultural and Environmental Engineering, vol. 3, no. 1, 2016.

[3]W. Wanasinghe, M. Gunarathna, H. Herath and G. Jayasinghe, "Drinking Water Quality on Chronic Kidney Disease of Unknown Aetiology (CKDu) in Ulagalla Cascade, Sri Lanka", Sabaragamuwa University Journal, vol. 16, no. 1, p. 17, 2018.

[4] UN-Water Global Analysis and Assessment of Sanitation and Drinking water - Sanitation, drinking-water and hygiene status overview*

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

[6] 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

[7]"Chronic Kidney Disease Prediction using Machine Learning Models", International Journal of Engineering and Advanced Technology, vol. 9, no. 1, pp. 6364-6367, 2019.

[8]Kleiman RS, LaRose ER, Badger JC, et al. Using Machine Learning Algorithms to Predict Risk for Development of Calciphylaxis in Patients with Chronic Kidney

Disease. AMIA Jt Summits Transl Sci Proc. 2018;2017:139-146. Published 2018 May 18.

[9]W. H. S. D. Gunarathne, K. D. M. Perera and K. A. D. C. P. Kahandawaarachchi, "Performance Evaluation on Machine Learning Classification Techniques for Disease Classification and Forecasting through Data Analytics for Chronic Kidney Disease (CKD)," 2017 IEEE 17th International Conference on Bioinformatics and Bioengineering (BIBE), Washington, DC, 2017, pp. 291-296, doi: 10.1109/BIBE.2017.00-39.

[10]Wanasinghe, W., Gunarathna, M., Herath, H. and Jayasinghe, G., 2018. Drinking Water Quality on Chronic Kidney Disease of Unknown Aetiology (CKDu) in Ulagalla Cascade, Sri Lanka. *Sabaragamuwa University Journal*, 16(1)

[11] X. Ji, X. Shang, R. A. Dahlgren and M. Zhang, "Prediction of dissolved oxygen concentration in hypoxic river systems using support vector machine: a case study of wen-rui tang river china", Environmental Science and Pollution Research, vol. 24, no. 19, pp. 16 062-16 076, 2017.

[13] E. Olyaie, H. Z. Abyaneh and A. D. Mehr, "A comparative analysis among computational intelligence techniques for dissolved oxygen prediction in delaware river", Geoscience Frontiers, vol. 8, no. 3, pp. 517-527, 2017.

APPENDICES

Ckdu_predict.py

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

# %matplotlib inline

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

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

encode = {'site_no': {}}

count = 0

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

data.replace(encode, inplace=True)

dict_site = encode['site_no']

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

```
test_size=0.20)

"""Polynomial SVM"""

print(data.head())

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

y_pred = svclassifier.predict(X_test)

from sklearn.metrics import accuracy_score

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

from sklearn import metrics

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

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

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


# site_no,temperature,pH,turbidity,tds,ec
def predictCkduRisk(site, temperature, ph, turbidity, tds, ec):
```

```
    if site in sites:
        site = dict_site[site]
        inputs = [[site, temperature, ph, turbidity, tds, ec]]
        output = model.predict(inputs)
        return output

    else:
        return 'ERROR'


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

app.py

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

        avgPatients = calclAvgPatientsForSite(site)

        if (temperature != None) and (site != None) and (turbidity !=
None) and (ph != None):
            output = predictCKD(site, temperature, ph, turbidity, tds,
conductivity)
```

```
            res = {
                "result": output[0],
                "avgPatients": avgPatients
            }
            print(res)
        else:
            res = {
                'success': 'True',
                'message': 'Incorrect input'
            }

    except:

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

    return jsonify(res)
```

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

ParameterController.js

```
var AWS = require("aws-sdk");

let awsConfig = {
```

```
    "region": "us-west-2",

    "endpoint": "http://dynamodb.us-west-2.amazonaws.com",

    "accessKeyId": "AKIAI5Q2SLBW2YDVNETQ", "secretAccessKey":

"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"

};


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));
        }
      });
    }
}

module.exports = new ParameterController();
```

Route.js

```
onst express = require('express')
const router = express.Router()

router.use('/parameters',require('./routes/ParamRoute'));

module.exports = router
```