# Water Quality Prediction for Drinking using Machine Learning

*Report submitted to the SASTRA Deemed to be University*

*in partial fulfilment of the requirements*

*for the award of the degree of*

Bachelor of Technology


*Submitted by*

**Hema Kishore K**

**Reg. No.: 224003037, B.Tech CSE**

**Sai Charan Velpuru**

**Reg. No.: 224003098, B.Tech CSE**

**Varshith Sai Naragam**

**Reg. No.: 224003157, B.Tech CSE**


**May 2024**



**Department of Computer Science and Engineering**

**Srinivasa Ramanujan Centre**

**Kumbakonam - 612001**

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act,1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

THANJAVUR | KUMBAKONAM | CHENNAI

**Department of Computer Science and Engineering**

**Srinivasa Ramanujan Centre**

**Kumbakonam - 612001**

**Bonafide Certificate**

This is to certify that the report titled "**Water Quality Prediction for Drinking using Machine Learning**" submitted in partial fulfilment of the requirements for the award of the degree of B. Tech Computer Science and Engineering to Department of Computer Science and Engineering, is a bonafide record of the work done by Shri/Mr. **Hema Kishore K** (**Reg. No**. **224003037, B.Tech CSE**), **Sai Charan Velpuru (Reg. No.: 224003098, B.Tech CSE), Varshith Sai Naragam (Reg. No.: 224003157, B.Tech CSE)** during the academic year 2023-24, in the Srinivasa Ramanujan Centre, under my supervision.

**Signature of Project Supervisor :**

**Name with Affiliation :** Dr. P.Venkateswari/AP-II/CSE/SRC/SASTRA

**Date :**

Project Based Work Viva voce held on _____

**Examiner 1** **Examiner 2**

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA
THANJAVUR | KUMBAKONAM | CHENNAI

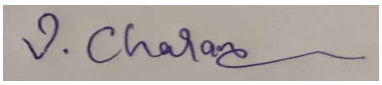
**Department of Computer Science and Engineering**

**Srinivasa Ramanujan Centre**

**Kumbakonam – 612001**

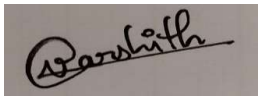
**Declaration**

We declare that the project report titled "**Water Quality Prediction for Drinking using Machine Learning**" submitted by us is an original work done by us under the guidance of **Dr.P.Venkateswari/AP-II/CSE/SRC/SASTRA** during the final semester of the academic year 2023-24, in the Department of Computer Science and Engineering. The work is original and wherever we have used materials from other sources, we have given due credit and cited them in the text of the report. This report has not formed the basis for the award of any degree, diploma, associate-ship, fellowship or other similar title to any candidate of any University.



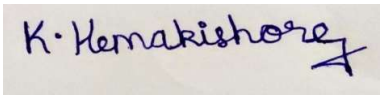
Sai Charan Velpuru

**Signature of Student** **Name of the Student**



Varshith Sai Naragam

**Signature of Student** **Name of the Student**



Hema Kishore K

**Signature of Student** **Name of the Student**

**Date :**

## ACKNOWLEDGEMENT

We pay our sincere obeisance to the God Almighty for his grace and infinite mercy and for showing on us his choicest blessings.

We would like to express our thanks to our Chancellor **Prof. R. Sethuraman,** Vice Chancellor **Dr. S. Vaidhyasubramaniam** and Registrar **Dr. R. Chandramouli** for having given us an opportunity to be a student of this esteemed institution.

We express our deepest thanks to **Dr. V. Ramaswamy**, Dean and **Dr. A. Alli Rani**, Associate Dean, Srinivasa Ramanujan Centre for their constant support and suggestions when required without any reservations.

We express our gratitude to HOD in charge **Dr. V.Kalaichelvi/ACP/CSE**, Srinivasa Ramanujan Centre for his constant support and valuable suggestion for the completion of the project.

We exhibit our pleasure in expressing our thanks **Dr. P.Venkateswari/AP-II/CSE**, our guide for her ever-encouraging spirit and meticulous guidance for the completion of the project.

We would like to place on record the benevolent approach and pain taking efforts of guidance and correction of **Dr.R.Bhavani APII/CSE,** the project coordinator and all department staffs to whom we owe our hearty thanks for ever.

Without the support of our parents and friends this project would never have become reality. We dedicate this work to our well-wishers, with love and affection.

# TABLE OF CONTENTS

# List of Figures

# ABBREVATIONS

**RFR** - Random Forest Regressor

**NumPy –** Numerical Python

**ML** - Machine Learning

**RF** - Random Forest

**SVM** - Support Vector Machine

**XG Boost** - Extreme Gradient Boosting

**ReLU** - Rectified Linear Unit

**SMOTE** - Synthetic Minority Oversampling Technique

**ILOC** - Integer-Location Based Indexing

**GUI -** Graphical User Interface

# Abstract

Water is necessary for the survival of humans, animals, and plants. Despite its importance, high-quality water is not always appropriate for residential, commercial or drinking purposes. A variety of factors, such as pollution, mining operations and industrial development impacts the quality of water by adding or changing its present parameters, thereby affecting the water's ability for human intake or general use. The World Health Organization's specifications specify the limits of various parameters that have to be present in water samples used for irrigation purposes or consumption. It would become quite difficult to collect water samples from multiple sources, determine the numerous characteristics present, and compare these metrics to pre-established standards while following various transportation and measurement procedures. Using Machine Learning (ML) algorithms, we can identify whether water samples are safe for irrigation and drinking. To classify the water, three machine learning models were tested: Random Forest (RF), Support Vector Machine (SV) and XG Boost. The three machine learning models were combined with recurrent feature reduction to identify which water parameter has the greatest impact on each model's classification accuracy.



**Key Words :** Machine Learning, Random Forest, Support vector machine, XG Boost, Accuracy .


## Specific Contribution :

Sai Charan V - Algorithms implementation, methodology and error solving.

Varshith Sai N - Data analysis and preprocessing.

Hema Kishore K - Gui and Algorithms comparison.

## Specific Learning :

Sai Charan Velpuru - Training , searching methodology of xgboost ,approach of lstm and implementation of algorithms.

Varshith Sai Naragam - Concepts of data analytics and preprocessing for our model.

Hema Kishore K - Developing gui using various methods for representing our output in a structured manner.

# CHAPTER – 1

# SUMMARY OF THE BASE PAPER

**Title: WaterNet: A Network for Monitoring and Assessing Water Quality for Drinking and Irrigation Purposes**

**Journal Name:** IEEE

**Publisher:** Olasupo O. Ajayi, Antoine B. Bagula, Hloniphani C. Maluleke, Zaheed Gaffoor Nebo Jovanovic and Kevin C. Pietersen
**Year:** 2022

The title of the base paper is "**WaterNet: A Network for Monitoring and Assessing Water Quality for Drinking and Irrigation Purposes**". This paper was published in the year 2022.

## 1.1 SUMMARY:

The paper discusses the critical problem of water quality assessment, emphasising its importance to human health, agriculture, and sustainability of the environment. It emphasises the difficulties caused by limited access to clean water and the absence of continuous monitoring systems. The proposed solution involves the establishment of a comprehensive monitoring system that combines real-time data collection with machine learning algorithms to automate water quality assessment.

**Key components of the proposed system include:**

- Utilization of machine learning algorithms, including Random Forest, Logistic Regression, LSTM, SVC, and XGBoost, to assess water quality for drinking purposes.

- Evaluation of regional standards and guidelines, especially those established by the World Health Organization (WHO) and the South African Bureau of Standards (SABS), to make sure the system's relevance and applicability in a variety of geographical locations.

- While acknowledging challenges such as data limitations and implementation complexity, the paper focuses on the proposed system's possible impact on public health, agriculture, and the sustainability of the environment. However, the system's effective implementation requires careful attention of validation, accuracy, expenses, infrastructure requirements, and moral considerations.

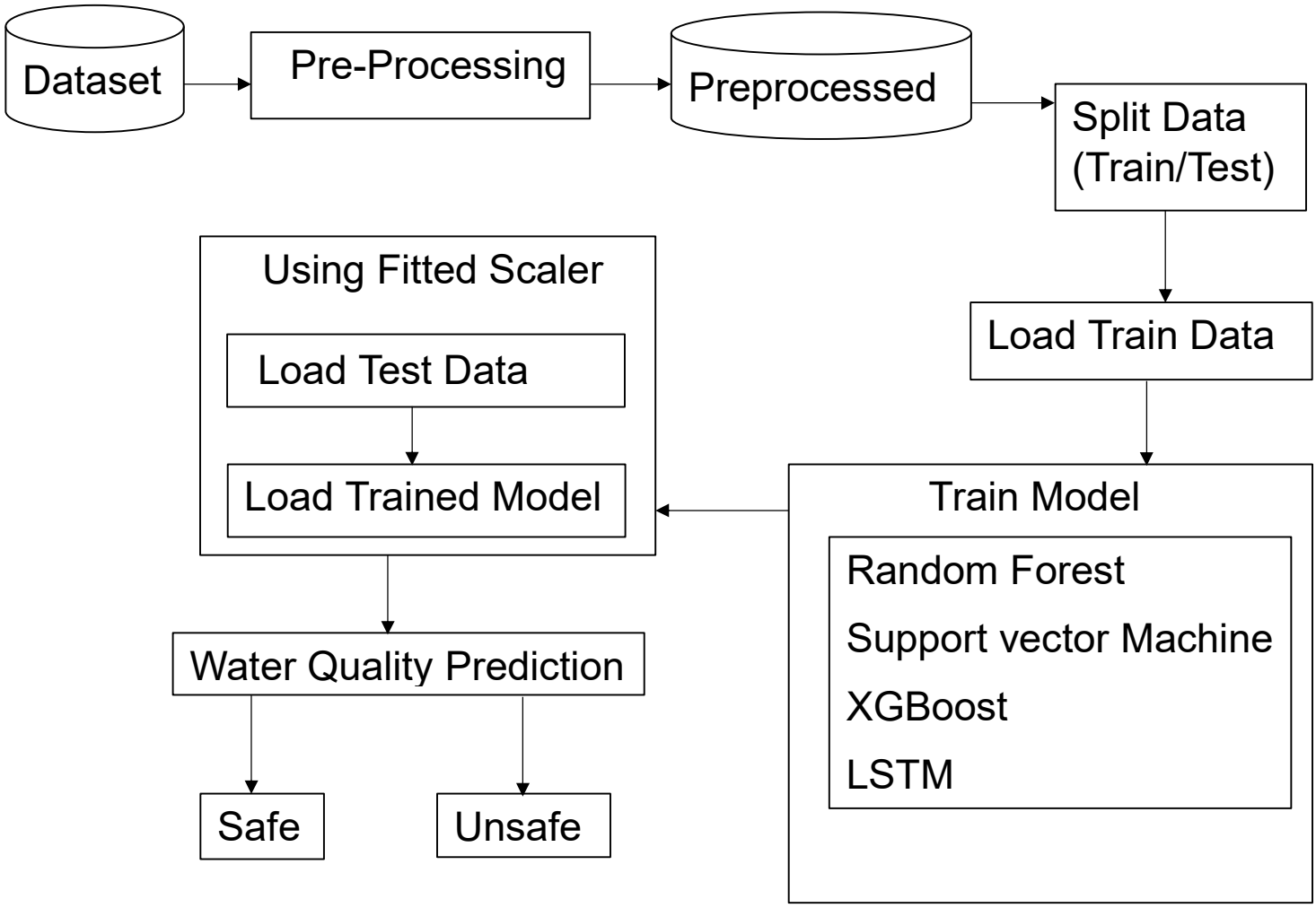
## 1.2 INTRODUCTION:

- Access to clean water is a fundamental human right and a necessary component for achieving sustainable development objectives.
- Contaminated water poses serious health risks, particularly in developing countries where access to safe water is limited.
- Water is crucial in agriculture, with irrigation required for crop growth and livestock consumption.
- Monitoring water quality is essential to ensure that it is safe for consumption, irrigation domestic and industrial use.
- Different regions have used different standards and guidelines for evaluating water quality, which are often based on global standards established by organizations such as the World Health Organization (WHO).

## 1.3 PROBLEM STATEMENT:

- Lack of access to clean water harms human health and threatens global agricultural sustainability.
- The lack of large, publicly available datasets impedes the development of effective machine learning models for assessing water quality.
- The system should include machine learning algorithms that automatically assess water quality for drinking and irrigation.
- Water quality management needs the integration of data from various sources, such as periodic sampling, laboratory analysis, and historical records.
- Adherence to regional standards and regulations, including those established by the WHO and SABS, is critical to maintaining the efficiency of the system across a wide range of geographical locations.

**1.4 ARCHITECTURE DIAGRAM:**




## 1.5 Data Set :

- This dataset contains measurements of water quality and potability assessments, which are used to determine whether water is safe for human consumption. The dataset's primary goal is to provide insights into water quality parameters and help determine whether the water is suitable or not.

- This data set contains 7999 rows and 21 columns (decimal) which include:

| | | |
|---|---|---|
| 1. Aluminium | 8. Copper | 15. Mercury |
| 2. Ammonia | 9. Flouride | 16. Perchlorate |
| 3. Arsenic | 10. Bacteria | 17. Radium |

| | | |
|---|---|---|
| 4. Barium | 11. Viruses | 18. Selenium |
| 5. Cadmium | 12. Lead | 19. Silver |
| 6. Chloramine | 13. Nitrates | 20. Uranium |
| 7. Chromium | 14. Nitrites | 21. is_safe - class attribute {0 - not safe, 1 - safe} |

## 1.6 Functions and techniques used in the source code:

1) **Data Loading:**
   **pd.read_csv():** Load data from a CSV file into a Pandas DataFrame.

2) **Data Exploration and Visualization:**
   **Pandas** and **NumPy** functions for exploring data (e.g., head(), info(), describe()).
   **Matplotlib**, **Seaborn**, and **Plotly** for data visualization (e.g., plt.plot(), sns.heatmap(), px.pie()).

3) **Data Preprocessing:**
   **Handling missing values:**
   isnull(), sum(): Check for missing values.
   fillna(): Fill missing values with a specified value or method.

4) **Feature scaling:**
   **StandardScaler():** Standardize features by removing the mean and scaling to unit variance.

5) **Balancing data:**
   **SMOTE():** Synthetic Minority Over-sampling Technique for balancing class distribution.

6) **Modeling:**
   **Machine learning models:**
   Logistic Regression: LogisticRegression()
   Random Forest: RandomForestClassifier()
   Support Vector Machine (SVM): SVC()
   XGBoost: XGBClassifier()

   **Hyperparameter tuning:**
   RandomizedSearchCV(): Randomized search cross-validation for hyperparameter tuning.

7) **Feature importance:**

**feature_importances_:** Attribute for extracting feature importance from tree-based models like Random Forest and XGBoost.

8) **Model Evaluation:**

**Evaluation metrics:**

**accuracy_score():** Calculate accuracy of classification models.

**classification_report():** Generate a detailed classification report including precision, recall, and F1-score.

**confusion_matrix():** Compute confusion matrix to evaluate the performance of classification models.

**Cross-validation:**

**cross_val_score():** Perform cross-validation to estimate the performance of machine learning models.

9) **Others:**

**KFold():** K-Folds cross-validator for splitting data into train and test sets in K consecutive folds.

**np.argsort(), np.concatenate():** Numpy functions for array manipulation.

Custom plotting function for feature importance analysis.

10) **Accuracy Comparison:**

Comparing the accuracy of different models using bar plots. This allows for a quick comparison of the performance of various algorithms on the same dataset.

# CHAPTER - 2

# MERITS AND DEMERITS OF BASE PAPER

## 2.1 MERITS:

- Addressing Critical Issues: The paper addresses major global water quality challenges, such as access to safe drinking water, health risks, and agricultural sustainability.

- Innovative Approach: The proposal for a monitoring system that combines real-time data collection with machine learning algorithms is novel and has the potential to improve the efficiency and effectiveness of water quality assessment.

- Utilization of Machine Learning: The paper uses advanced techniques to automate water quality assessment by employing various machine learning algorithms such as Random Forest, Logistic Regression, LSTM, SVC, and XGBoost.

- Potential for Impact: If efficiently implemented, the proposed system has the possibility to improve public health, agriculture, and environmental sustainability by providing access to clean and safe water.

## 2.2 DEMERITS:

- Data Limitations: The absence of huge open-source datasets for training machine learning models may limit the proposed system's effectiveness and generality.

- Complexity of Implementation: Integrating real-time data collection with various machine learning algorithms may present difficulties in system design, deployment, and maintenance.

- Validation and Accuracy: The paper fails to provide detailed data on the method for validation or the accuracy of the machine learning models, raising concerns about the proposed system's reliability.

## CHAPTER - 3

### SOURCE CODE

```
import numpy as np
import pandas as pd
import pandas as pd
import matplotlib.pyplot as plt
plt.style.use('ggplot')
plt.style.use('dark_background')
import seaborn as sns
import plotly.express as px
from imblearn.over_sampling import SMOTE
from collections import Counter
from sklearn.preprocessing import StandardScaler
import warnings
from sklearn.model_selection import (
    train_test_split,
    KFold,
    RandomizedSearchCV,
    cross_val_score
)
from scipy.stats import (
    randint,
    uniform
)
from xgboost import XGBClassifier
from sklearn.metrics import (
    accuracy_score,
    confusion_matrix,
    classification_report
)
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from xgboost.sklearn import XGBClassifier
file_path = r'C:\Users\charan\OneDrive\Desktop\waterQuality1.csv'


# Load data into a DataFrame
df = pd.read_csv(file_path)
# Display the first few rows of the DataFrame
print(df.head())
missing_value = ['#NUM!', np.nan]
df = pd.read_csv(r'C:\Users\charan\OneDrive\Desktop\waterQuality1.csv',na_values = missing_value)
df.head(5)
# Check for missing values
if df.isnull().sum().sum() == 0:
    print("No missing values found in the dataset.")
else:
    missing_values_per_feature = df.isnull().sum()
    plt.figure(figsize=(10, 6))
    sns.barplot(x=missing_values_per_feature.index, y=missing_values_per_feature.values, hue=missing_values_per_feature.index)
    plt.title('Missing Values Per Feature')
    plt.xlabel('Features')
    plt.ylabel('Number of Missing Values')
    plt.xticks(rotation=90)
    plt.legend([], frameon=False)  # Hide legend
    plt.show()
# Handling missing values for each attribute
for column in df.columns:
    if df[column].isna().sum() > 0:
```

```
        # Convert column to numeric (if it's not already numeric)
        df[column] = pd.to_numeric(df[column], errors='coerce')
        column_mean = df[column].mean(skipna=True)
        df[column].fillna(column_mean, inplace=True)
df.dropna(inplace = True)
df['is_safe'].value_counts()
# Split the df into features (X) and target variable (y)
X = df.drop('is_safe', axis=1)
y = df['is_safe']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42)
print("df split into training and testing sets.")
X_train, X_test, y_train, y_test = train_test_split(df.drop('is_safe', axis = 1),
                                                    df['is_safe'],
                                                    test_size = 0.2,
                                                    random_state = 0)
"""## Initial Analysis"""
# Checking the shape of the dataset
df.shape
# Getting information about the dataset
df.info()
# Converting the 'is_safe' feature into a categorical variable
df['is_safe'] = df['is_safe'].astype('category')
# Checking the number of unique values in each column
df.nunique()
# Descriptive statistics for the dataset
df.describe().T.style.background_gradient(subset=['mean', 'std', '50%', 'count'], cmap='PuBu')

# Descriptive statistics for is_safe = 1
```

```
df[df['is_safe'] == 1].describe().T.style.background_gradient(subset=['mean', 'std', '50%', 'count'], cmap='PuBu')


# Descriptive statistics for is_safe = 0
df[df['is_safe'] == 0].describe().T.style.background_gradient(subset=['mean', 'std', '50%', 'count'], cmap='RdBu')
# Exploratory Data Analysis
Corrmat = df.corr()
plt.subplots(figsize=(10, 10))
sns.heatmap(Corrmat, cmap="coolwarm", square=True, annot=True, fmt='.2f')
plt.title('Correlation Heatmap')
plt.show()


# Pie chart
fig = px.pie(df, names='is_safe', hole=0, template='plotly_dark')
fig.show()
# List of features you want to visualize
features_of_interest = ['aluminium', 'ammonia', 'arsenic', 'barium', 'cadmium', 'chloramine',
                        'chromium', 'copper', 'flouride', 'bacteria', 'viruses', 'lead',
                        'nitrates', 'nitrites', 'mercury', 'perchlorate', 'radium',
                        'selenium', 'silver', 'uranium']


# Creating box plots for multiple features
for feature in features_of_interest:
    plt.figure(figsize=(8, 6))
    sns.boxplot(x='is_safe', y=feature, data=df, hue='is_safe', palette='Set2', legend=False)
    plt.title(f'Box Plot of {feature} by is_safe')
    plt.xlabel("safety")
    plt.ylabel(feature)
    plt.show()
```

```
# Handle missing values
df.replace('#NUM!', np.nan, inplace=True)
# List of features for visualization
features_of_interest = ['aluminium', 'ammonia', 'arsenic', 'barium', 'cadmium', 'chloramine',
                        'chromium', 'copper', 'flouride', 'bacteria', 'viruses', 'lead',
                        'nitrates', 'nitrites', 'mercury', 'perchlorate', 'radium',
                        'selenium', 'silver', 'uranium']


# KDE plot of different features by is_safe
print('KDE plot of different features by is_safe\n')


fig, ax = plt.subplots(ncols=1, nrows=len(features_of_interest), figsize=(7, 6*len(features_of_interest)))


for i, col in enumerate(features_of_interest):
    try:
        sns.kdeplot(data=df, x=col, fill=True, alpha=0.4,
                    hue='is_safe', multiple='stack', ax=ax[i],warn_singular=False)
        ax[i].set_xlabel(' ')
        ax[i].set_ylabel(' ')
        ax[i].xaxis.set_tick_params(labelsize=14)
        ax[i].tick_params(left=False, labelleft=False)
        ax[i].set_title(f'KDE Plot of {col} by is_safe', fontsize=16)
    except np.linalg.LinAlgError:
        print(f"LinAlgError occurred for feature: {col}. Consider addressing the issue.")
        continue


plt.tight_layout()
plt.show()
# Define features (X) and target variable (y)
X = df.drop(columns=['is_safe'])  # Features
```

```
y = df['is_safe']  # Target variable

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Calculate percentage of data in training and testing sets
train_percent = len(X_train) / len(df) * 100
test_percent = len(X_test) / len(df) * 100

print(f"Percentage of data in training set: {train_percent:.2f}%")
print(f"Percentage of data in testing set: {test_percent:.2f}%")
# Balancing the data by SMOTE - Oversampling of Minority level
from imblearn.over_sampling import SMOTE
from collections import Counter
# Instantiate SMOTE
smt = SMOTE()

# Display count of target variable classes before SMOTE
counter_before = Counter(y_train)
print('Before SMOTE', counter_before)
# Convert continuous values into discrete classes
y_train = y_train.round().astype(int)

# Apply SMOTE to the training data
X_train_resampled, y_train_resampled = smt.fit_resample(X_train, y_train)
# Display count of target variable classes after SMOTE
counter_after = Counter(y_train_resampled)
print('\nAfter SMOTE', counter_after)
# Initialize the StandardScaler
ssc = StandardScaler()
```

```
# Scale the training and testing data
X_train_scaled = ssc.fit_transform(X_train)
X_test_scaled = ssc.transform(X_test)
# Print the scaled data
print("Scaled Training Data:")
print(X_train_scaled)
print("\nScaled Testing Data:")
print(X_test_scaled)
#Histogram
# Define feature names
feature_names = ['aluminium', 'ammonia', 'arsenic', 'barium', 'cadmium', 'chloramine',
                 'chromium', 'copper', 'flouride', 'bacteria', 'viruses', 'lead',
                 'nitrates', 'nitrites', 'mercury', 'perchlorate', 'radium',
                 'selenium', 'silver', 'uranium']

# Plot histograms for each feature in the training data
plt.figure(figsize=(15, 6))
for i in range(min(10, X_train_scaled.shape[1])):
    plt.subplot(2, 5, i + 1)
    plt.hist(X_train_scaled[:, i], bins=20, color='skyblue', edgecolor='black')
    plt.title(feature_names[i])  # Set custom feature name
plt.tight_layout()
plt.suptitle('Histograms of Scaled Training Data Features', y=1.05)
plt.show()

# Plot histograms for each feature in the testing data
plt.figure(figsize=(15, 6))
for i in range(min(10, X_test_scaled.shape[1])):
    plt.subplot(2, 5, i + 1)
    plt.hist(X_test_scaled[:, i], bins=20, color='salmon', edgecolor='black')
```

```
    plt.title(feature_names[i])  # Set custom feature name
plt.tight_layout()
plt.suptitle('Histograms of Scaled Testing Data Features', y=1.05)
plt.show()

#model comparision
models = [
    RandomForestClassifier(),
    SVC(),
    XGBClassifier()
]
# Initialize lists to store results
train_accuracy = []
test_accuracy = []
# Define KFold cross-validation
kfold = KFold(n_splits=10, random_state=7, shuffle=True)

# Perform cross-validation and evaluation for each model
for model in models:
    # Cross-validation on training data
    train_result = cross_val_score(model, X_train_scaled, y_train, scoring='accuracy', cv=kfold)
    train_accuracy.append(train_result.mean())
    # Fit the model on training data and make predictions on test data
    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)
    # Evaluate accuracy on test data
    test_accuracy.append(accuracy_score(y_test, y_pred))

# Create a DataFrame to compare model performances
model_comparison = pd.DataFrame({'Model': ['Random Forest', 'SVM', 'XGBoost'],
```

```
                    'Train_Accuracy': train_accuracy,
                    'Test_Accuracy': test_accuracy})
print('Model Comparison:')
print(model_comparison)
#Support Vector Machine
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
# Train the SVM model with adjusted hyperparameters for reduced accuracy
model_svm = SVC(C=2, kernel='rbf', gamma='auto', random_state=42)  # Adjust C parameter to reduce accuracy
model_svm.fit(X_train, y_train)

# Make predictions
pred_svm = model_svm.predict(X_test)
# Calculate accuracy score
svm_accuracy = accuracy_score(y_test, pred_svm)
# Print classification report
print("Classification Report:")
print(classification_report(y_test, pred_svm))

# Plot confusion matrix
cm = confusion_matrix(y_test, pred_svm)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Reds', cbar=False)
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
#Random Forest
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
# Train the Random Forest model with adjusted hyperparameters
model_rf = RandomForestClassifier(n_estimators=2, max_depth=5, min_samples_split=5, min_samples_leaf=1, random_state=42)
model_rf.fit(X_train, y_train)
# Make predictions
pred_rf = model_rf.predict(X_test)
# Calculate accuracy score
rf_accuracy = accuracy_score(y_test, pred_rf)
print('Accuracy:',rf_accuracy)
# Print classification report
print("Classification Report:")
print(classification_report(y_test, pred_rf))

# Plot confusion matrix
cm = confusion_matrix(y_test, pred_rf)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Reds', cbar=False)
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()

#XGBoost
model = XGBClassifier()
param_grid = {
    'n_estimators': randint(50, 251),
    'max_depth': randint(3, 15),
```

```
    'min_child_weight': randint(1, 11),
    'gamma': uniform(0.0, 1.0),
}
kf = KFold(n_splits = 5, shuffle = True, random_state = 0)
search = RandomizedSearchCV(model,
                            param_grid,
                            scoring = 'accuracy',
                            cv = kf,
                            n_iter = 100,
                            refit = True,
                            n_jobs = -1)

search.fit(X_train, y_train)
print(f'Train Score : {accuracy_score(y_train, search.predict(X_train))}')
print(f'Test Score : {accuracy_score(y_test, search.predict(X_test))}')
confusion_matrix(y_test, search.predict(X_test))
print(classification_report(y_test, search.predict(X_test)))

# Print confusion matrix
cm_xgb= confusion_matrix(y_test, search.predict(X_test))
plt.figure(figsize=(8, 6))
sns.heatmap(cm_xgb, annot=True, fmt='d', cmap='Reds', cbar=False)
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
print(search.predict(X_train))
#Feature importance
import xgboost as xgb
def plot_feature_importance(model, feature_names, title):
```

```
    try:
        if hasattr(model, 'feature_importances_'):  # Check if model has feature_importances_ attribute
            feature_importance = model.feature_importances_
            sorted_idx = np.argsort(feature_importance)
            y_ticks = np.arange(0, len(feature_names))
            fig, ax = plt.subplots(figsize=(8, 6))
            ax.barh(y_ticks, feature_importance[sorted_idx])
            ax.set_yticks(y_ticks)
            ax.set_yticklabels(np.array(feature_names)[sorted_idx])
            ax.set_title(title)
            plt.show()
        elif isinstance(model, xgb.XGBModel):  # Check if model is an XGBModel instance
            booster = model.get_booster()
            importance = booster.get_score(importance_type='weight')
            sorted_idx = np.argsort(list(importance.values()))
            feature_importance = [importance[list(importance.keys())[i]] for i in sorted_idx]
            y_ticks = np.arange(0, len(feature_names))
            fig, ax = plt.subplots(figsize=(8, 6))
            ax.barh(y_ticks, feature_importance)
            ax.set_yticks(y_ticks)
            ax.set_yticklabels(np.array(feature_names)[sorted_idx])
            ax.set_title(title)
            plt.show()

        else:
            print("Model does not support feature importance analysis.")
    except AttributeError as e:
        print("AttributeError:", e)
```

```
# Random Forest Classifier Feature Importance
print("Random Forest Classifier Feature Importance:")
plot_feature_importance(model_rf, X.columns, "Random Forest Classifier")

# XGB Classifier Feature Importance
print("XGB Classifier Feature Importance:")
# Fit the XGBoost model before calling plot_feature_importance
model.fit(X_train, y_train)
plot_feature_importance(model, X.columns, "XGB Classifier")

# Support Vector Classifier (SVC) Feature Importance
print("Support Vector Classifier (SVC) Feature Importance:")
plot_feature_importance(model_svm, X.columns, "Support Vector Classifier")

# Calculate accuracy scores for each model
accuracy_scores = [
    accuracy_score(y_test, pred_svm),
    accuracy_score(y_test, pred_rf),
    accuracy_score(y_test, search.predict(X_test)),
]

# Plotting the accuracy scores
plt.figure(figsize=(12, 8))
plt.bar(['SVC','Random Forest', 'XGBoost'], accuracy_scores, color=['grey','lightyellow','pink'])
plt.xlabel('Classifier')
plt.ylabel('Accuracy')
plt.title('Accuracy of Different Classifiers')
plt.ylim(0, 1)  # Set y-axis limit from 0 to 1

# Adding accuracy values on top of bars
```

```
for i, acc in enumerate(accuracy_scores):
    plt.text(i, acc + 0.02, f'{acc:.2f}', ha='center', va='bottom')
plt.show()
#Prediction using best algorithm
import numpy as np
# Take 5 random values from the dataset for prediction
random_indices = np.random.choice(len(X_train), 10)  # Adjust the number of samples as needed
X_random = X_train.iloc[random_indices]

# Predict probabilities using the trained XGBoost model
y_prob_random = xgb_model.predict_proba(X_random)[:, 1]  # Assuming class 1 is positive (safe for drinking)

# Define your threshold
threshold = 0.3 # Adjust the threshold as needed

# Convert probabilities to class labels based on the threshold
y_pred_random = (y_prob_random > threshold).astype(int)

# Interpret predictions
interpretation = ['Potable (Safe for Drinking)' if pred == 1 else 'Not Potable (Not Safe for Drinking)' for pred in y_pred_random]

# Display predictions along with the corresponding random values
predictions_df = pd.DataFrame({'Random Sample Index': random_indices, 'Predicted Potability': y_pred_random, 'Predicted Drinking Water': interpretation})
print("Predictions using XGBoost Model (Random Samples):")
print(predictions_df)
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout,Input
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
from sklearn.preprocessing import StandardScaler
file_path = r'C:\Users\charan\OneDrive\Desktop\waterQuality1.csv'

# Load data into a DataFrame
df = pd.read_csv(file_path)
# Display the first few rows of the DataFrame
print(df.head())
import pandas as pd
# Load data into a DataFrame
file_path = r'C:\Users\charan\OneDrive\Desktop\waterQuality1.csv'
df = pd.read_csv(file_path)

# Replace '#NUM!' with NaN in the target variable
df.loc[df['is_safe'] == '#NUM!', 'is_safe'] = np.nan

# Drop rows where the target variable has NaN values
df.dropna(subset=['is_safe'], inplace=True)

# Convert 'is_safe' column to numeric
df['is_safe'] = pd.to_numeric(df['is_safe'], errors='coerce')
```

```
# Fill NaN values resulting from non-numeric conversions with 0
df['is_safe'] = df['is_safe'].fillna(0)
threshold = 0.5
# Convert values based on the threshold
df['is_safe'] = (df['is_safe'] > threshold).astype(int)
# Display the updated value counts of the target variable
print(df['is_safe'].value_counts())

from sklearn.preprocessing import StandardScaler
df.replace('#NUM!', np.nan, inplace=True)
# Convert all columns to numeric
df = df.apply(pd.to_numeric, errors='coerce')
# Handle missing values by imputing them using mean
df.fillna(df.mean(), inplace=True)

# Apply StandardScaler to normalize the features
X = df.drop(columns=['is_safe'])  # Assuming 'is_safe' is the target column
y = df['is_safe']
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
# Display the first few rows of the scaled features
print(pd.DataFrame(X_scaled, columns=X.columns).head())
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder

# Define subsets of features
feature_subsets = ['aluminium', 'ammonia', 'arsenic', 'barium', 'cadmium', 'chloramine',
                   'chromium', 'copper', 'flouride', 'bacteria', 'viruses', 'lead',
```

```
            'nitrates', 'nitrites', 'mercury', 'perchlorate', 'radium',
            'selenium', 'silver', 'uranium', 'is_safe']

# Drop rows with missing values
cleaned_df = df.dropna(subset=feature_subsets).copy()
# Encode 'is_safe' column into numeric values
label_encoder = LabelEncoder()
cleaned_df['is_safe'] = label_encoder.fit_transform(cleaned_df['is_safe'])

# Calculate the correlation matrix
corr_matrix = cleaned_df[feature_subsets].corr()
# Plot the correlation matrix heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('Correlation Matrix Heatmap')
plt.show()

X_train, X_val, y_train, y_val = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
# Print the shapes of the training and validation sets
print("Shape of X_train:", X_train.shape)
print("Shape of X_val:", X_val.shape)
print("Shape of y_train:", y_train.shape)
print("Shape of y_val:", y_val.shape)

import tensorflow as tf
# Reshape the input data to have three dimensions
X_train_reshaped = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_val_reshaped = X_val.reshape(X_val.shape[0], X_val.shape[1], 1)

# Define the LSTM model architecture
```

```
model = tf.keras.Sequential([
    tf.keras.layers.LSTM(units=128, return_sequences=True, input_shape=(X_train_reshaped.shape[1], X_train_reshaped.shape[2])),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.LSTM(units=64),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(units=32, activation='relu'),
    tf.keras.layers.Dense(units=1, activation='sigmoid')
])

# Print the model summary
print(model.summary())
# Compile the model with appropriate optimizer, loss function, and metric
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
print("Model compiled successfully.")

# Define early stopping callback
early_stopping = EarlyStopping(monitor='val_loss', patience=20, restore_best_weights=True)
print("Early stopping callback defined.")
# Train the model with early stopping
history = model.fit(X_train.reshape(X_train.shape[0], X_train.shape[1], 1), y_train, epochs=30, batch_size=64,
                    validation_data=(X_val.reshape(X_val.shape[0], X_val.shape[1], 1), y_val),
                    callbacks=[early_stopping], verbose=1)
print("Model training completed.")

# Plot training history
plt.plot(history.history['loss'], label='train_loss')
plt.plot(history.history['val_loss'], label='val_loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
```

```
plt.legend()
plt.show()

# Calculate predicted probabilities
y_prob = model.predict(X_val_reshaped)
# Convert probabilities to class labels
y_pred = (y_prob > 0.5).astype(int)
# Calculate confusion matrix
conf_matrix = confusion_matrix(y_val, y_pred)
# Calculate accuracy
accuracy = accuracy_score(y_val, y_pred)
# Calculate classification report
class_report = classification_report(y_val, y_pred)

# Print confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Reds', cbar=False)
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()

# Print accuracy and classification report
print("\nAccuracy:", accuracy)
print("\nClassification Report:")
print(class_report)

import numpy as np
import pandas as pd
```

```
# Take random values from the dataset for prediction
random_indices = np.random.choice(len(X_val_reshaped), 10)  # Adjust the number of samples as needed
X_random = X_val_reshaped[random_indices]
serial_numbers = np.arange(len(X_val_reshaped))[random_indices]  # Generating serial numbers assuming the index as serial numbers


# Predict water quality using the trained LSTM model
y_random_prob = model.predict(X_random)


# Convert probabilities to class labels
y_random_pred = (y_random_prob > 0.3).astype(int)


# Interpret predictions
interpretation = ['Potable (Safe for Drinking)' if pred == 1 else 'Not Potable (Not Safe for Drinking)' for pred in y_random_pred]


# Display predictions
predictions_df = pd.DataFrame({'Serial Number': serial_numbers, 'Predicted Potability': y_random_pred.flatten(), 'Predicted Drinking Water': interpretation})
print("Predictions using LSTM Model:")
print(predictions_df)
```

# CHAPTER - 4

# SNAPSHOTS

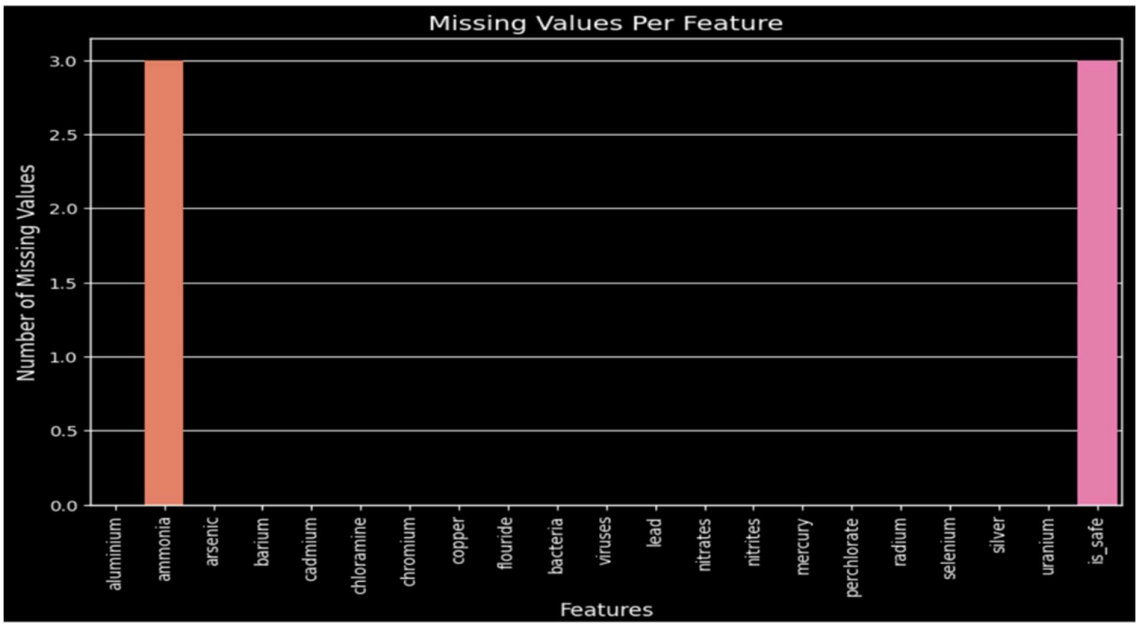



Fig 4.1 Missing Values

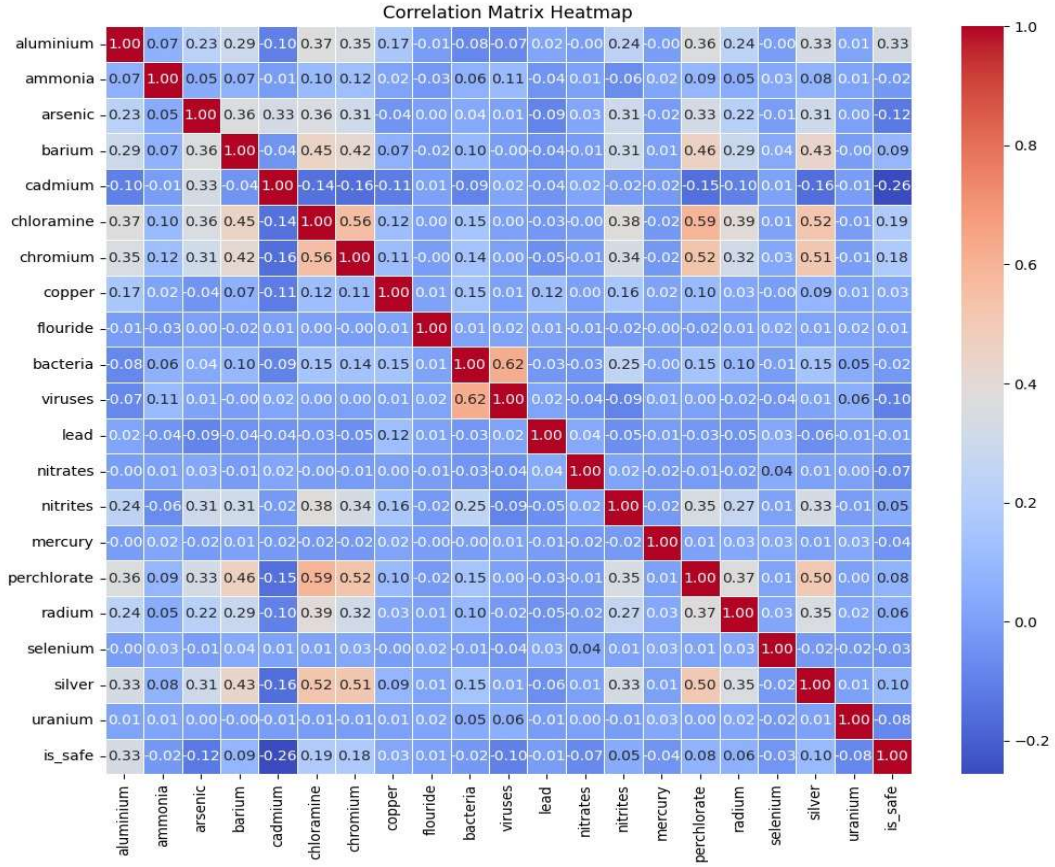



Fig 4.2 Heat Map

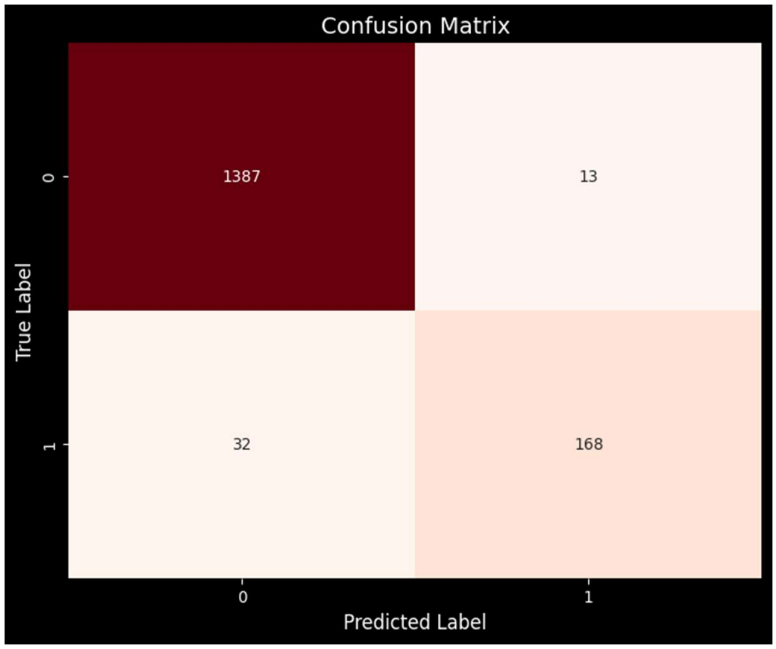


Fig 4.3 Confusion Matrix for SVM

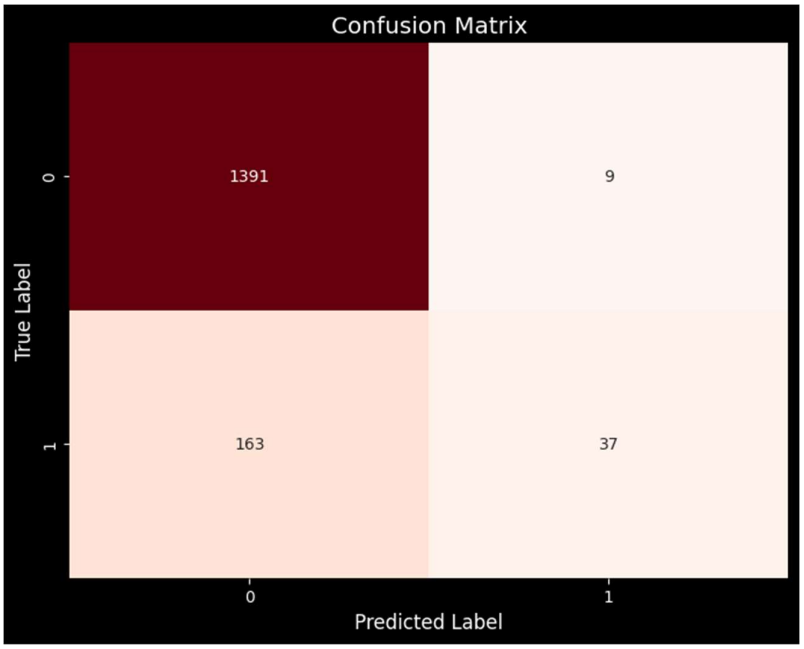


Fig 4.4 Confusion Matrix for Random Forest

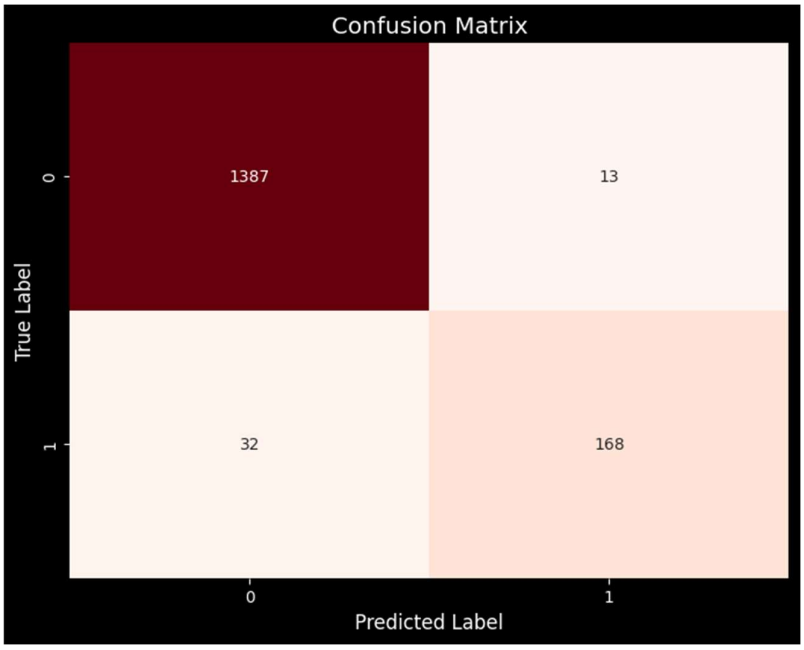


Fig 4.5 Confusion Matrix for XGBoost

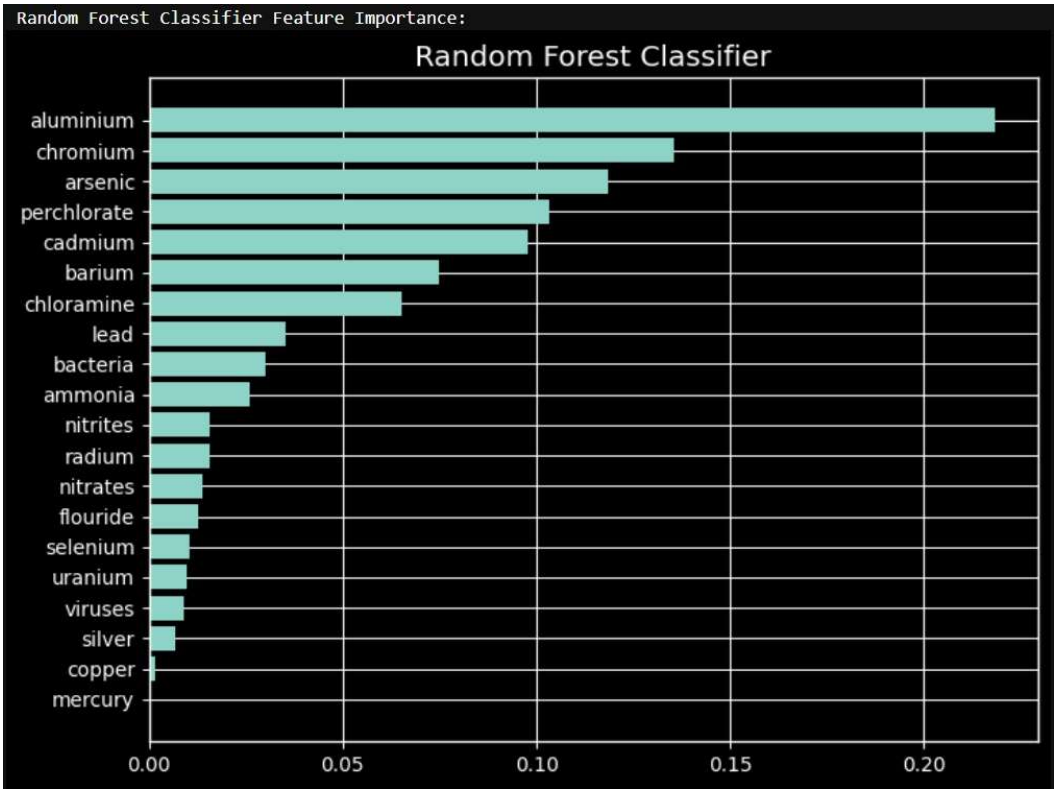


Fig 4.6 Feature importance of Random Forest

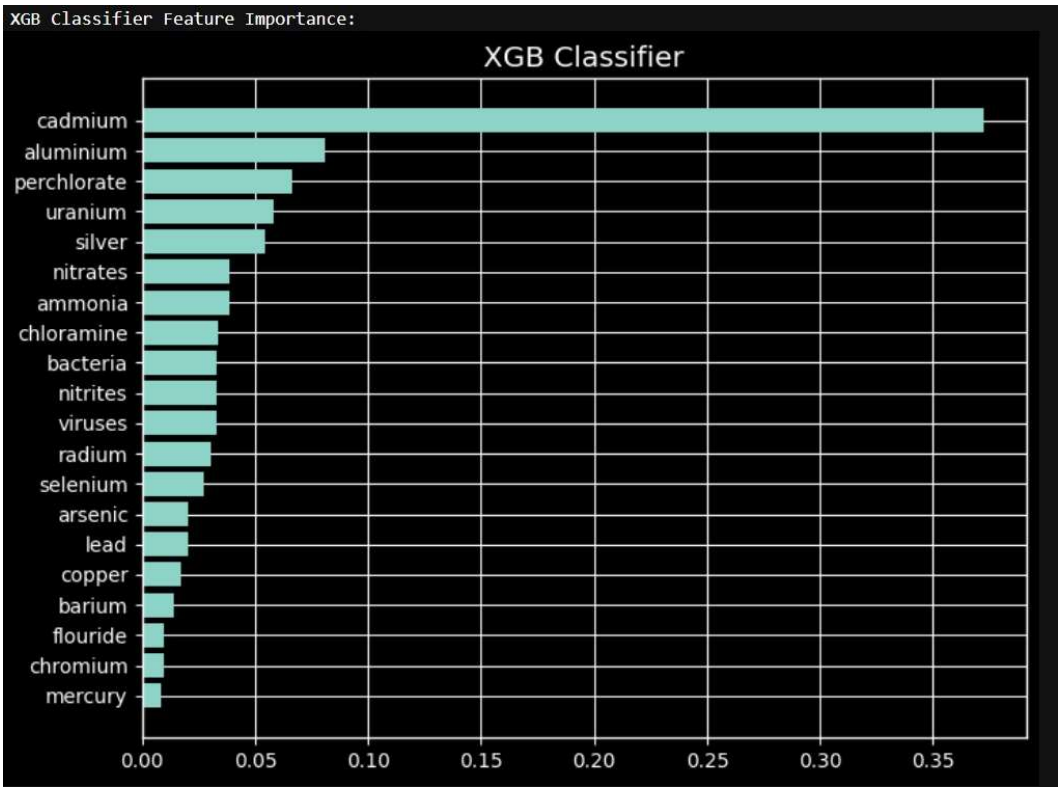



Fig 4.7 Feature Importance of XGB Classifier

Model: "sequential_1"

| Layer (type) | Output Shape | Param # |
|---|---|---|
| lstm_2 (LSTM) | (None, 20, 128) | 66,560 |
| dropout_2 (Dropout) | (None, 20, 128) | 0 |
| lstm_3 (LSTM) | (None, 64) | 49,408 |
| dropout_3 (Dropout) | (None, 64) | 0 |
| dense_2 (Dense) | (None, 32) | 2,080 |
| dense_3 (Dense) | (None, 1) | 33 |

**Total params:** 118,081 (461.25 KB)

**Trainable params:** 118,081 (461.25 KB)

**Non-trainable params:** 0 (0.00 B)

None

Fig 4.8 LSTM Model Summary

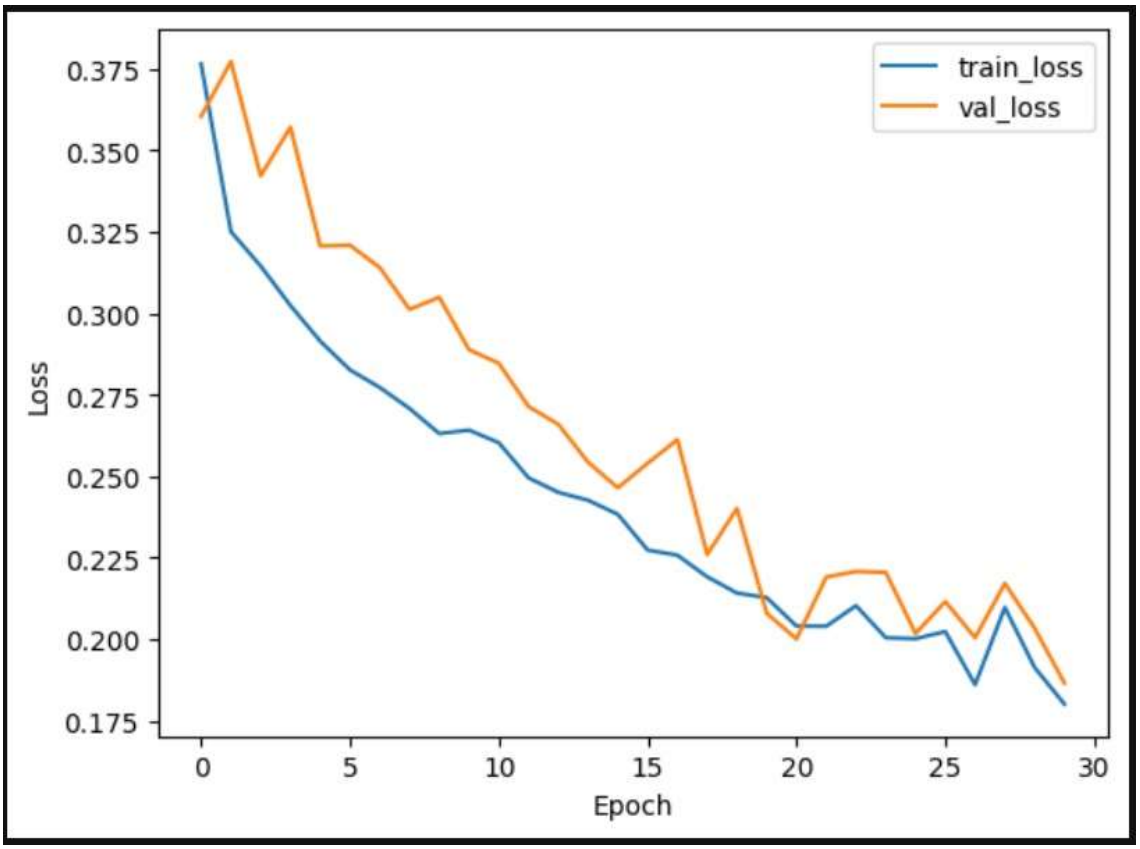



Fig 4.9 Loss Graph

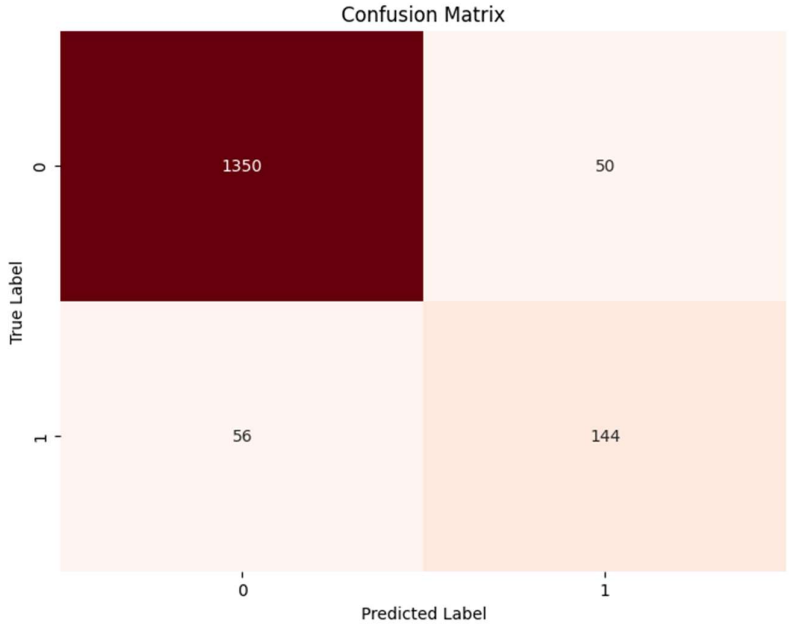



Fig 4.10 Confusion Matrix for LSTM

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 97ms/step
Predictions using LSTM Model:
   Serial Number  Predicted Potability            Predicted Drinking Water
0            383                     0  Not Potable (Not Safe for Drinking)
1            669                     1          Potable (Safe for Drinking)
2            584                     0  Not Potable (Not Safe for Drinking)
3           1422                     0  Not Potable (Not Safe for Drinking)
4            631                     1          Potable (Safe for Drinking)
5            953                     1          Potable (Safe for Drinking)
6           1127                     0  Not Potable (Not Safe for Drinking)
7            600                     0  Not Potable (Not Safe for Drinking)
8           1238                     0  Not Potable (Not Safe for Drinking)
9            134                     0  Not Potable (Not Safe for Drinking)
```

Fig 4.11 Predictions using LSTM Model

```
Predictions using XGBoost Model (Random Samples):
   Random Sample  Predicted potability            Predicted Drinking Water
0           1284                     0  Not Potable (Not Safe for Drinking)
1           6151                     0  Not Potable (Not Safe for Drinking)
2           1353                     1          Potable (Safe for Drinking)
3           5063                     0  Not Potable (Not Safe for Drinking)
4           4962                     0  Not Potable (Not Safe for Drinking)
5           4415                     0  Not Potable (Not Safe for Drinking)
6           3100                     0  Not Potable (Not Safe for Drinking)
7           6088                     0  Not Potable (Not Safe for Drinking)
8           2435                     0  Not Potable (Not Safe for Drinking)
9            990                     0  Not Potable (Not Safe for Drinking)
```

Fig 4.12 Predictions using XGBoost

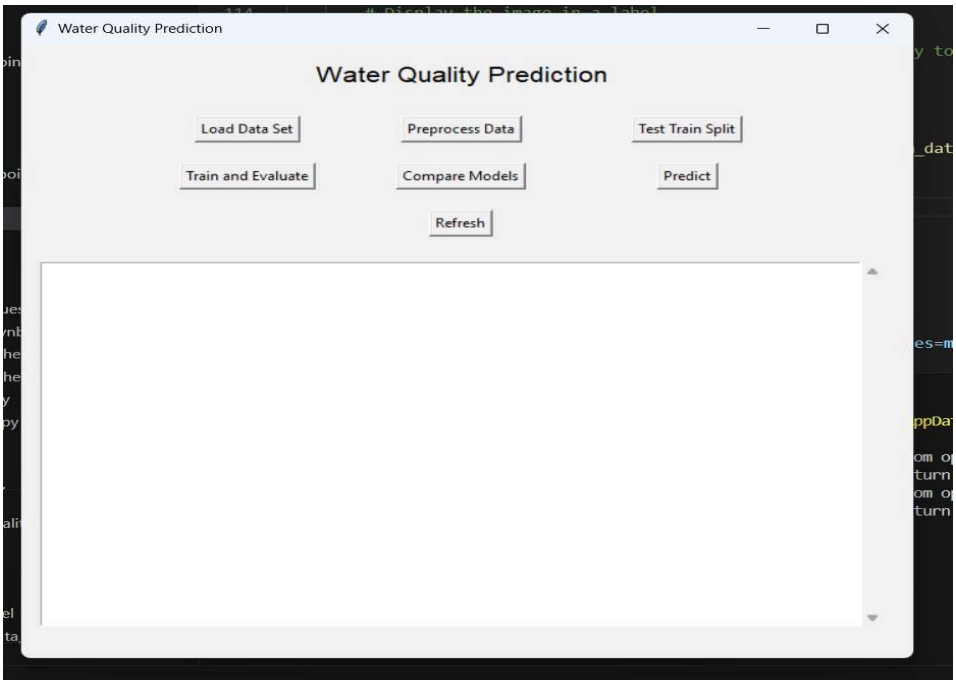



Fig 4.13 GUI Window

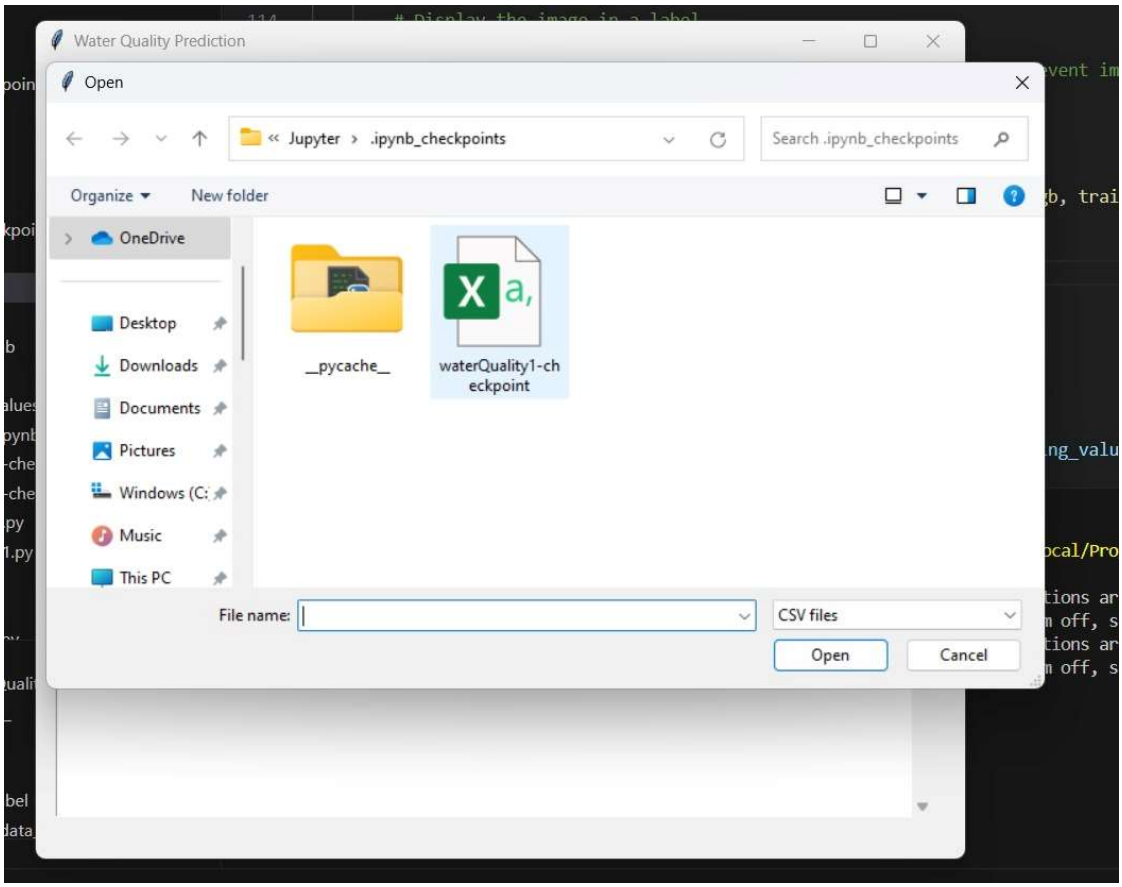



Fig 4.14 Loading Dataset

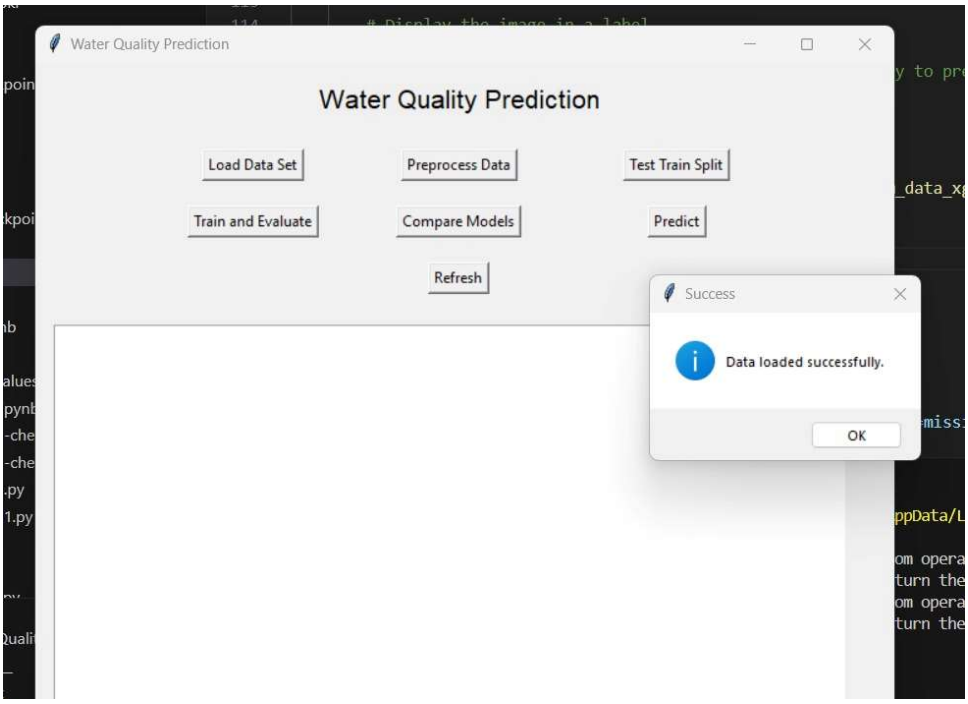



Fig 4.15 Data set Loaded Successfully

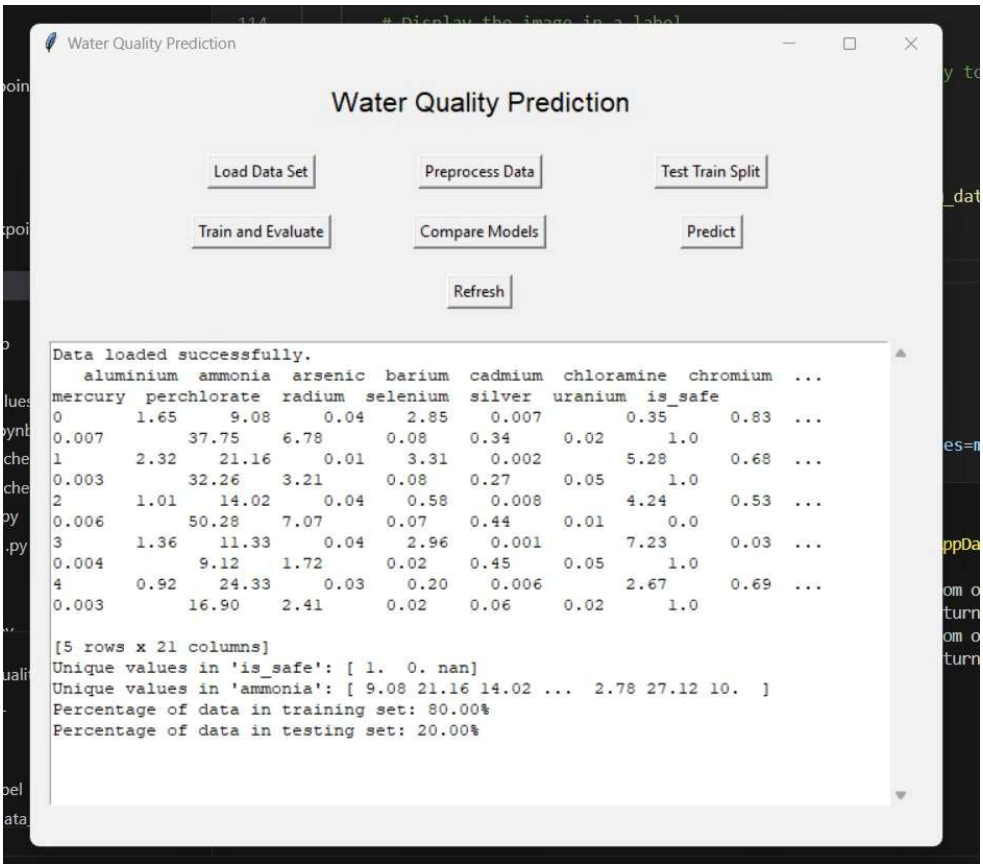



Fig 4.16 Results after performing Preprocessing data and Test train split steps

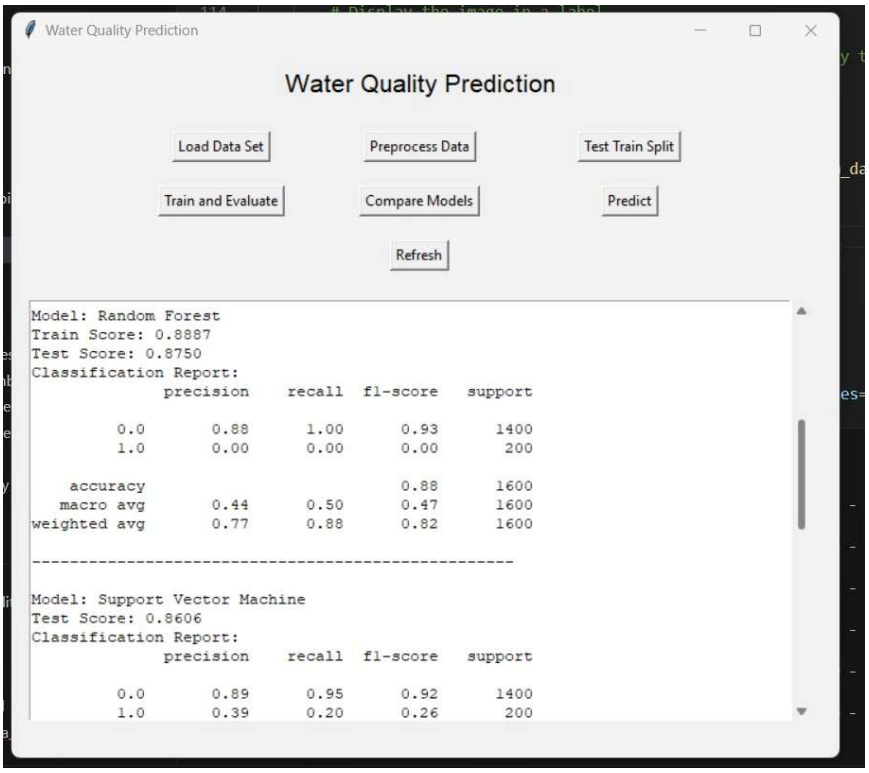



Fig 4.17 Train and Evaluate results for Random Forest

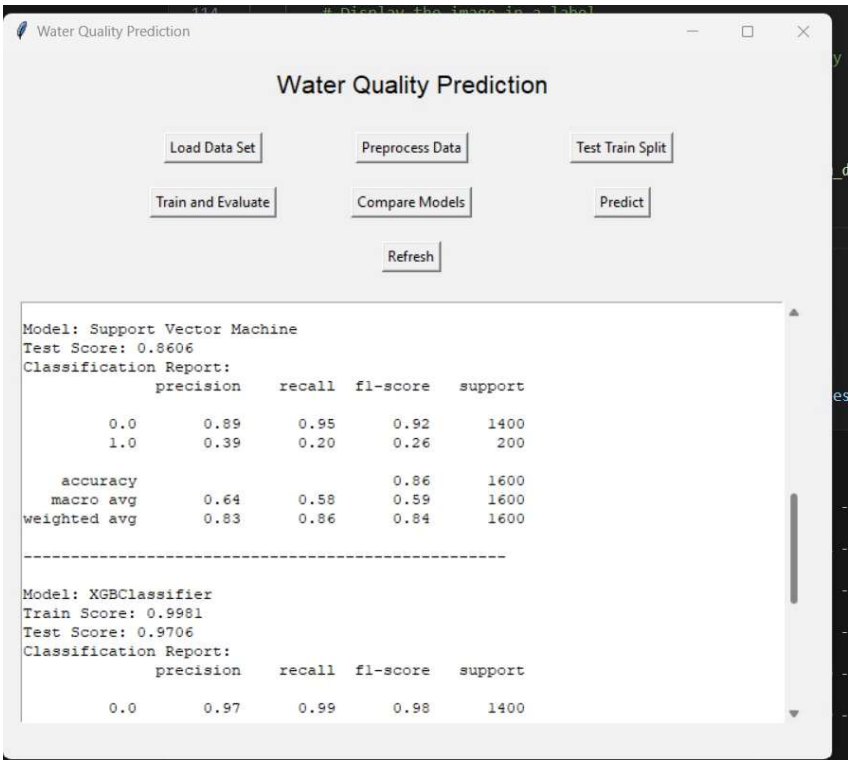



Fig 4.18 Train and Evaluate results for Support Vector Machine

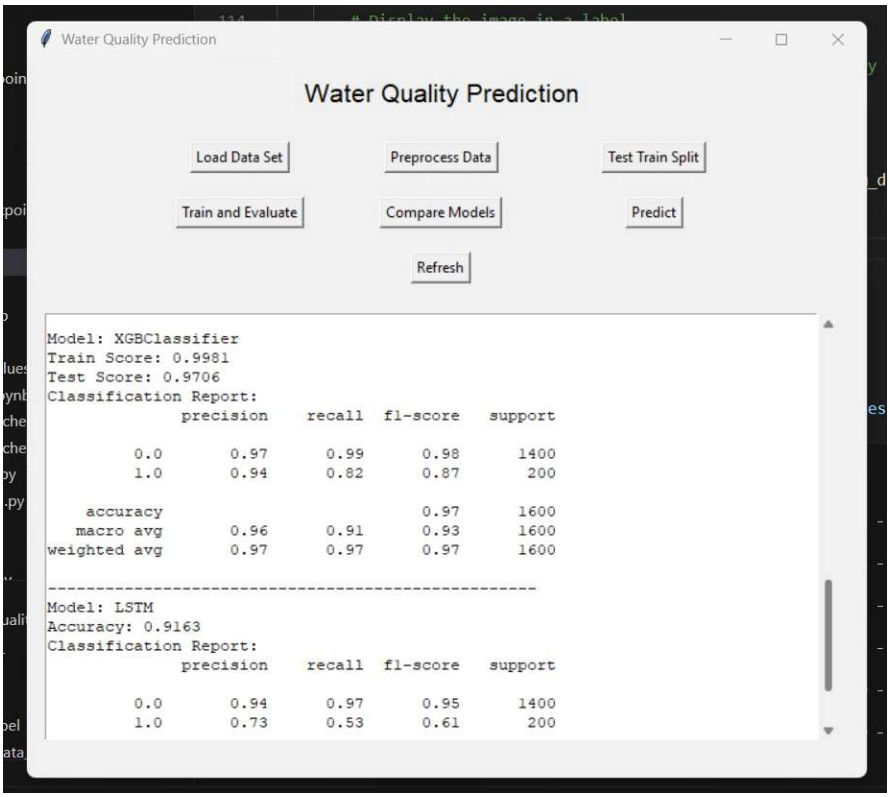



Fig 4.19 Train and Evaluate results for XGBClassifier

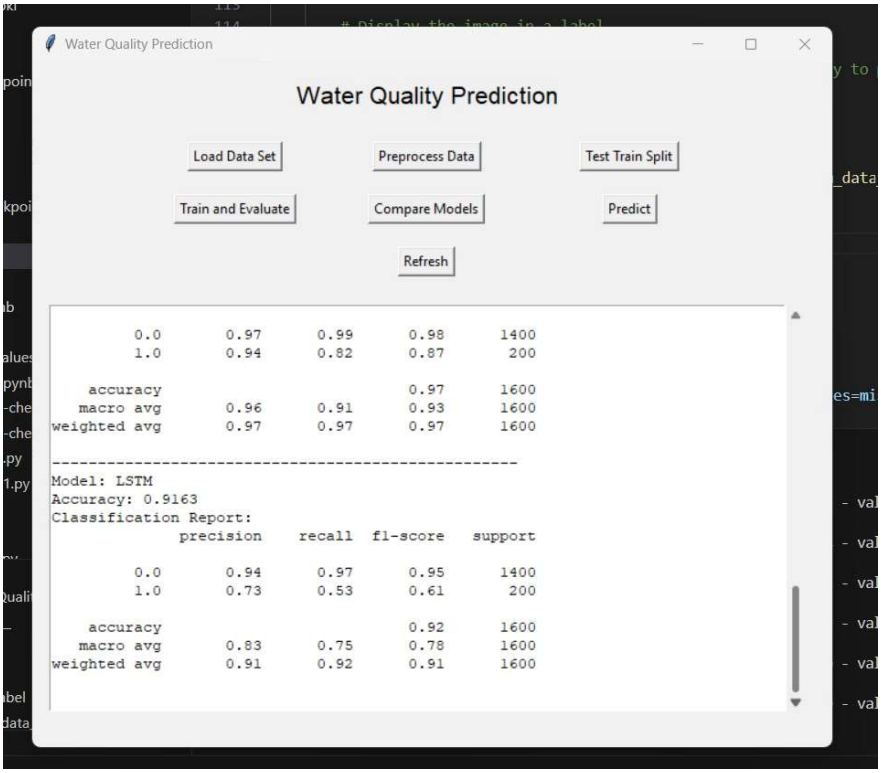



Fig 4.20 Train and Evaluate results for LSTM

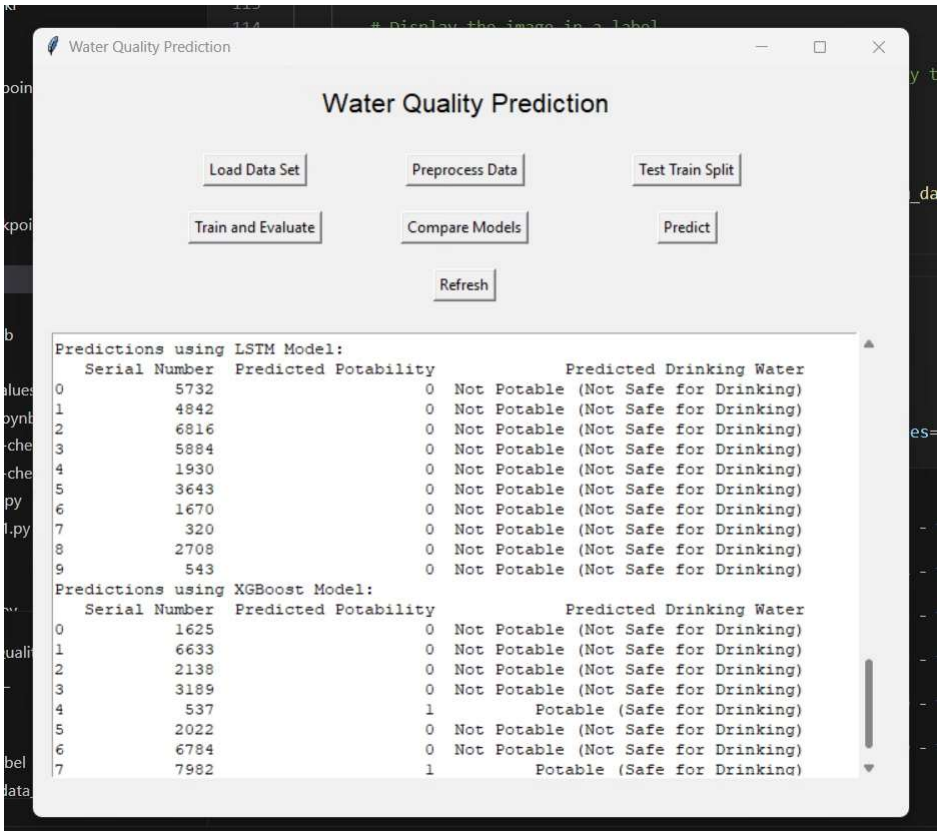



Fig 4.21 Predictions

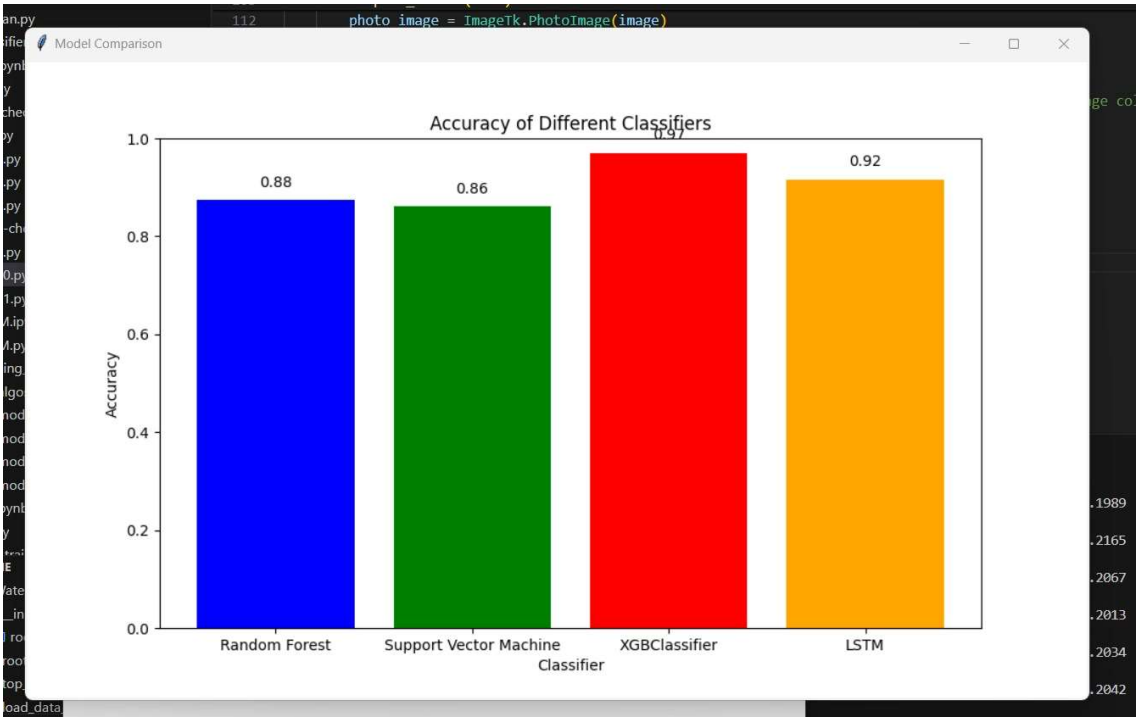



Fig 4.22 Accuracy comparison for all classifiers

# CHAPTER - 5
# RESULTS AND DISCUSSIONS

The results of the water quality prediction models show that machine learning algorithms can accurately predict water quality parameters. XGBoost emerged as the most accurate model, closely followed by LSTM, with Random Forest and SVM also performing well. XGBoost can be recommended due to its outstanding accuracy and efficiency, making it great for real-time applications requiring high precision. LSTM is especially effective for time series data and should be used when dealing with sequential water quality measurements over a period of time.Random Forest and SVM are dependable choices for water quality prediction, providing a good balance of accuracy and computational efficiency. They may be preferred in situations where accessibility and scalability are essential.

# CHAPTER – 6

# CONCLUSION

The project aimed to develop an accurate model for predicting water quality based on multiple factors, using machine learning techniques. To ensure the dataset's quality and consistency, significant data preprocessing was performed, including missing value handling and categorical variable encoded data. Several machine learning models, including XGBoost and LSTM neural networks, have been trained and evaluated to determine the best approach to water quality prediction. The models' performance was assessed using metrics such as accuracy, precision, recall, and F1-score, with LSTM showing ensuring results in terms of accuracy and predictability. Despite challenges such as data imbalances and design complexities, the project was able to create a reliable water quality prediction model which can assist in environmental monitoring and decision-making.

# CHAPTER – 7

# REFERENCES


1. World Health Organization. (1993). Guidelines for Drinking-Water Quality. World Health Organization. Accessed: Jan. 12, 2022. [Online]. Available: http://apps.who.int/iris/bitstream/handle/ 10665/44584/9789241548151-eng.pdf

2. Standard Methods for the Examination of Water and Wastewater, Federation WE, APH Association, American Public Health Association (APHA), Washington, DC, USA, 2005.

3. L. Wang, Support Vector Machines: Theory and Applications. Cham, Switzerland: Springer, Jun. 2005.

4. R. K. Horton, ‘‘An index number system for rating water quality,’’ J. Water Pollut. Control Fed., vol. 37, no. 3, pp. 300–306, Mar. 1965.

# CHAPTER - 8

# APPENDEX - A

# BASE PAPER

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 2, 2022, accepted April 27, 2022, date of publication May 3, 2022, date of current version May 10, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3172274*

# WaterNet: A Network for Monitoring and Assessing Water Quality for Drinking and Irrigation Purposes

**OLASUPO O. AJAYI[1], ANTOINE B. BAGULA[1], HLONIPHANI C. MALULEKE[1], ZAHEED GAFFOOR[2], NEBO JOVANOVIC[2], AND KEVIN C. PIETERSEN[3]**

[1]Department of Computer Science, University of the Western Cape, Bellville, Cape Town 7535, South Africa
[2]Department of Earth Science, University of the Western Cape, Bellville, Cape Town 7535, South Africa
[3]Institute for Water Studies, University of the Western Cape, Bellville, Cape Town 7535, South Africa

Corresponding author: Olasupo O. Ajayi (olasupoajayi@gmail.com; ooajayi@uwc.ac.za)

**ABSTRACT** Water is a fundamental requirement for human, animal, and plant survival. Despite its importance, quality water is not always fit for drinking, domestic and/or industrial use. Numerous factors such as industrialization, mining, pollution, and natural occurrences impact the quality of water, as they introduce or alter various parameters present therein, thus, affecting its suitability for human consumption or general use. The World Health Organization has guidelines which stipulate the threshold levels of various parameters present in water samples intended for consumption or irrigation. The Water Quality Index (WQI) and Irrigation WQI (IWQI) are metrics used to express the level of these parameters to determine the overall water quality. Collecting water samples from different sources, measuring the various parameters present, and bench-marking these measurements against pre-set standards, while adhering to various guidelines during transportation and measurement can be extremely daunting. To this end this study proposes a network architecture to collect data on water parameters in real-time and use Machine Learning (ML) tools to automatically determine suitability of water samples for drinking and irrigation purposes. The developed monitoring network is based on LoRa and takes the land topology into consideration. Results of simulations done in Radio Mobile revealed a partial mesh network topology as the most adequate. Due to the absence of large and open datasets on drinking and irrigation water, datasets usable for training ML models were developed. Three ML models - Random Forest (RF), Logistic Regression (LR) and Support Vector Machine (SVM) were considered for the water classification process and results obtained showed that LR performed best for drinking water, while SVM was better suited for irrigation water. Recursive feature elimination was then combined with the three ML models to reveal which of the water parameters had the greatest influence on the classification accuracies of the respective model.

**INDEX TERMS** Cyber physical system, LoRa, drinking water, irrigation water, machine learning, water quality index, water monitoring network.

## I. INTRODUCTION

Access to water is a critical component of human lives and is now considered a basic human right. Access to clean water is also one of the 17 Sustainable Development Goals (SDG) set up by the United Nations in 2015 to achieve a better future for all [1]. Specifically, the sixth goal, which is to ensure and sustain the availability of water and sanitation to all [2]. Potable water can also be linked to the third SDG goal – good health and well-being, as contaminated water can be a transmission medium for diseases such as cholera, typhoid, and diarrhoea, which are jointly the highest cause of mortality (especially children) in developing nations of Africa and Asia [3]. Water is also important in agriculture and food production. Recent statistics shows that about 10% of the world population is malnourished, with developing countries being hit the hardest, with starvation resulting in about 45% of infant mortality [5]. Ensuring global food security is thus

The associate editor coordinating the review of this manuscript and approving it for publication was Nadeem Iqbal.

of utmost importance. Food security has been recognized as a critical requirement, hence its inclusion as one of the SDG (goal 2), with specific focus on ending hunger, by promoting sustainable agriculture and improving food distribution. Food production and agriculture in general rely heavily on water, both for irrigation and for animal consumption. It is thus pertinent to ensure the availability and sustainable management of water fit for agricultural use.

There are several sources of water for both drinking and irrigation use, including rivers, streams, rain, and groundwater (accessed through wells and boreholes). The nature and characteristics of a source of water are often critical factors that influence the constituents of water samples obtained therein. Beyond natural factors, chemical wastes from human activities such as mining, crude oil extraction, and industrial wastes, most often end up in streams, rivers, and other sources of water, changing the nature and properties of these waters. These waters then end up in homes or farms, where they are used for domestic purposes, drank, fed to livestock, or used to water crops. Consuming this type of water can have dire health consequences or result in death. It is therefore paramount that a proper process be put in place to ensure end-to-end monitoring of the water right from the source to its last point of use. At each monitoring point, samples of water need to be collected to assess the quality or "fitness for use" for human (and animal) consumption, irrigation and domestic (or industrial) uses.

Several models have been developed to assess water quality, all of which consider various parameters, including chemical (such as hydrogen potential (pH), calcium, oxygen, sulphate levels etc.), microbial (such as E. coli, rotaviruses, Entamoeba etc.), and physical (temperature and clarity). These models produce a unit metric, known as the Water Quality Index (WQI), as output. Globally, different guidelines have been adapted for calculating WQI. For instance, in parts of Europe, the British Columbia Water Quality Index (BCWQI) and the Scottish Research Development Department (SRDD) are used, while in North America, the Canadian Council of Ministers of the Environment Water Quality Index (CCMEWQI) and National Sanitation Foundation Water Quality Index (NSFWQI) are predominant. In Asia, specifically India, the Bureau of Indian Standards (BIS) is prominent, while in Africa, notable standards include the South African National Standard for drinking water (SANS 241-1) and the Kenya Bureau of Standards (KEBS). A number of these models have been reviewed in [6]. It is important to note that many of these national standards are mostly local adaptations of the standards defined by the World Health Organization (WHO) [7]. This work is based on the South African and WHO standards.

Indeed, measuring water parameters for diverse water samples can be a laborious and daunting task, as it often involves adhering to a stringent set of rules in collecting the water samples, maintaining set conditions during transportation to the test laboratories, following standard methodologies in analysing the samples, and generally ensuring quality control. Some of these processes (and corresponding guidelines) are given in [8], [9]. The output of these processes indicates if the water sample is potable or non-potable. In this work, we propose a Cyber-physical network architecture for real-time monitoring of water parameters across a city and an alternative model based on machine learning to determine potability of water samples. Like [10]–[13] [14], our work also only focuses on the physical and chemical parameters of water, while ignoring the biological. This is because our model is meant to be sensor based (in the context of the Internet of Things), and to our knowledge, there are no physical sensors for measuring biological parameters, such as the presence of E. coli in water. We do not trivialize the importance of microbial water parameters, and our proposed model can indeed be adapted to consider these parameters by simply incorporating suitable physical sensors (if available) or virtual / soft sensors, such as the one proposed in [15] into our model.

Figure 1 gives a high-level depiction of our proposed architecture which is built upon 4 layers. The constituent components of this architecture are described as follows:

1) **Sensing Layer:** As depicted in the figure, the sensing layer interacts directly with the water samples in a river, stream, dam etc. to measure water parameters. It is built into a vertical pole tagged "sensor probe" and consists of numerous sensors bundled together. These sensors might include pH, conductivity, turbidity, temperature, residual chlorine etc., similar to those offered by Libelium [16]. All telemetry data measured by these sensors are sent to the Fog Nodes (FNs), wired or wirelessly, via the sending unit. In scenarios where installing sensors in water source(s) is extremely difficult or when the required sensors are not readily available, water parameter readings can be collected from the associated water treatment plants.
2) **Edge Layer:** This layer consists of low-end processing devices (edge modules), such as single board computers (e.g., Raspberry Pi or Nvidia Jetson), or microcontrollers (e.g. Arduino, ESP32). These devices act as i.) data pre-processing units, responsible for the collection, aggregation, filtration, and shaping of data received from the sensing layer; ii) network gateway to "ferry" telemetry data to the FNs, through 3G/4G/5G cellular or other low powered long-range network solutions.
3) **Fog / Cloud Layer:**
   - Fog Nodes (FNs): these are small sized distributed cloud computing nodes that bring computing and storage closer to the data source, thus reducing latency resulting from transmission delay to/from the remote Cloud [17]. The FN is responsible for classification of water samples using machine learning models such as the ones proposed in this work. Due to the limited computing power at the Fog (compared to the Cloud), only the most influential parameters need to be considered when

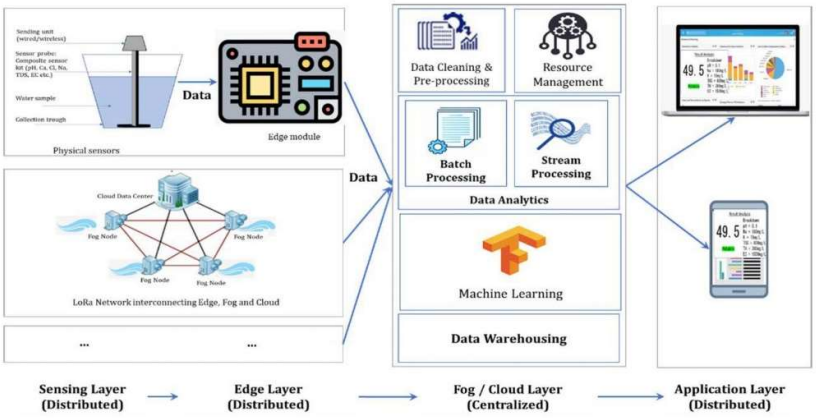



**FIGURE 1.** Conceptual framework for Water Quality Monitoring.

classifying water samples. This can be beneficial as less sensors would be required (since not all parameters are being measured) and by extension lower computing resources would be needed for the classification process. Furthermore, resource management, scheduling etc. can also be carried out on FNs. When long term storage and/or advanced computations are required, which are beyond the Fog's capacity, data are forwarded to the Cloud data centre.

- Cloud Data Centre: The Cloud is a remote high performance computing infrastructure, which provides computing on demand [18]. In our system, the Cloud serves as a data warehouse as well as a platform for performing advanced data analytics, dashboarding, and hosting for relevant services and software.

4) **Application Layer:** serves as an interface between users (water management authorities, end users / customers, other stakeholders) and software / services running in the Cloud. Relevant software for water parameter monitoring are hosted at this layer and made available to users through mobile and web platforms.

The water monitoring network proposed in this work is to be deployed in the City of Cape Town in Western Cape, South Africa, with the intention of monitoring water parameters in water storage dams and/or water treatment plants across the city. Data gathered by the monitoring network are then passed through Machine Learning (ML) models to determine their suitability for consumption or irrigation purposes. The specific contributions of this work can be summarized as follows:

1) Build a network for real-time collection and monitoring of water quality across water storage dams in the city of Cape Town. This network takes into consideration the unique geographical features of Cape Town, such as mountains and elevations that might obstruct radio frequency propagation.
2) Curate ample sized datasets on drinking and irrigation water that can be used to train (and test) machine learning models to automatically determine the "fitness for use" of a sample of water for drinking and/or irrigation purposes.
3) Build models that determine the most critical parameters that influence the accuracy of machine learning models in analysing water for drinking or irrigation.

Regarding the order of this paper, following this introductory section, is a review of related works in section II. Section III discusses our methodology for building the city-wide water monitoring network, while section IV presents the datasets curation process and machine learning models considered for determining quality of water samples. Implementation processes and obtained results from our experiments are discussed in sections V and VI respectively. Section VII discusses the economic viability of our proposed solution, while section VIII concludes the paper and gives insights into future directions.

## II. REVIEW OF LITERATURE

In this section, we review some existing works in literature on related subject matters. This section is divided into three main categories; first, the applications of wireless networks in monitoring water parameters. Second, yard-sticks for assessment drinkable water, and lastly, research works that focus on assessing suitability of water for irrigation purposes.

### 1) WIRELESS COMMUNICATION NETWORKS FOR WATER MONITORING

In [12], a network for measuring and monitoring water parameters in a metal producing city in Brazil was developed.

Twelve water monitoring stations were setup to measure several physico-chemical water parameters, including pH, dissolved solids, Zinc, Lead etc. Finally, obtained results were analysed using principal component analysis. In a similar manner, [13] developed a system to monitor water quality in Limpopo River Basin in Mozambique and set up 23 monitoring stations to measure physico-chemical and microbiological parameters, and ultimately assess the quality of water in the river basin. To address the challenges of optimal placement of gauges and sampling frequencies, which are often faced when developing water monitoring systems, the authors in [14] developed an economically viable model that combined genetic algorithm with 1-D water quality simulation. Though the work was only simulated by using genetic algorithm, the authors were able to solve the NP hard problem of optimally placing monitoring stations.

Monitoring water parameters often entails periodically sampling a body of water to capture relevant metrics. These metrics might include physico-chemical and microbiological measurements, such as potential of hydrogen (pH), temperature, sodium levels etc. In a water monitoring network, measured parameters need to be transferred to a base station where relevant decision(s) would be taken. Due to the sparse nature of transmitted data, light weight communication protocols capable of transmitting relatively small data over long distance are required for water monitoring networks. From literature, Low Power Wide Area Network (LPWAN) technologies have been favoured for such applications. An extensive discussion on LPWAN technologies was done in [19]. The work compared a few sub-GHz solutions including SigFox, LoRa, Ingenu and Telensa, with respect to their range, transmission rate, and channel count. Ingenu was reported to have the longest range in city settings at 15 km, followed by SigFox at 10 km (in cities) and 50 km (in rural areas); then LoRa at 5 km (in cities), and 15 km in rural settings.

Regarding the assessment of communication technologies, there has been a long-drawn debate over the efficacy of software simulations versus real-world testing. Though this debate still rages, several researchers have shown that simulation results are often at par with real-world tests. For instance, using LoRa, the authors in [20] compared simulation results with real world test for intervehicle communication. They used NS3 as a simulation platform and an Arduino UNO + Dragino LoRa module for the real-world tests, while Propagation loss, coverage Packet Inter-reception (PIR), Packet Delivery Ratio (PDR) and Received Signal Strength Indicator (RSSI) level were used as benchmark metrics. They concluded that the results of the simulator were consistent with those of the real-world tests. In a similar work, Hassan [21] also compared the efficacy of simulation results (from Radio Mobile simulator) with real-world tests (using micro controllers + LoRa modules) when using LoRa as a bridge for Wi-Fi. Unlike [20], [21] did not give a side-by-side comparison of simulated vs. real-world results for each metric considered but concluded that the simulator performed well. [22] set up seven pairs of XBee modules and compared communication performance using both the 800/900MHz and 2.4GHz frequencies. They concluded that simulation results from the Radio Mobile simulator corroborated with those of real-world tests.

#### 2) ASSESSING WATER POTABILITY

When assessing the quality of drinking water, the Water Quality Index (WQI) has been the de facto metric. It is a unitless numeric value that gauges the suitability of water for human consumption or general usage. As stated earlier, several models exist for calculating WQI depending on the location and environmental conditions in such locations. In a recent study by Uddin *et al.* [23], it was noted that there are about 35 WQI models in use globally; however, in their opinion, the major ones are the Horton Index, National Sanitation Foundation WQI, the Canadian Council of Ministers of the Environment (CCME) WQI, Scottish Research development Department (SRDD) index, Bascaron index (BWQI), Fuzzy Interface system (FIS), and the Malaysian water quality index (MWQI). The study compared these models in terms of structural composition, parameters considered, indexing and weighting criteria, application areas and inherent limitations. For most of these models, a WQI value of at least 50 was considered acceptable. In a related work [6] also reviewed several WQI models but with emphasis on parameter importance. The work selected the most common parameters used in literature and applied analytical hierarchical process (AHP) and measuring attractiveness by a categorically based evaluation technique (MACBETH) to assign weights to water parameters and select the most relevant ones.

In [10] the authors sought to assess the impact of mining activities on water quality in certain areas of Bangladesh. Twelve parameters were considered, including pH, electrical conductivity (EC), turbidity, hardness, salinity etc. These were then benchmarked against the WHO standards to determine WQI. In another work, [11] applied WQI to urban water resource management. The work follows up on an earlier study in [12], where a water monitoring network was set up to give information about water, by including information about the quality of water across the twelve monitoring points using WQI. Two models were used to calculate the WQI, namely CCME WQI and Cetesb WQI. CCME classified all samples as poor, while Cetesb resulted in a mix of Good, Fair and Poor.

A major shortcoming of WQI is its site specificity, which implies that WQI is often calculated for a specific body of water or region, using the parameters therein. It therefore cannot be automatically applied to a different water body except when the two share similar attributes and parameter ranges. Moreover, WQI are developed to target specific use case(s), hence, bounded by the constraints set for that use case(s). In a bid to tackle this and make WQI water sample agnostic, [24] proposed a universal WQI model that is applicable to all water bodies in South Africa. The authors applied 13 parameters selected from literature and experts. To obtain a universal WQI, the authors created a custom aggregation function, which treats the WQI inputs from different water

sources as a system of linear equations. Their unified model was able to classify water samples from the different sources effectively.

#### 3) ASSESSING WATER QUALITY FOR IRRIGATION

Irrigation water is a vital part of food production, especially crop farming. The quality of water can affect crop yield, hence concerted efforts need to be made to ensure proper water quality standards [25]. Like with drinking water, several classical techniques exist for ascertaining the quality of irrigation water (or irrigation water quality index – IWQI), however most are either tailored to drinking water alone or not economically viable for local farmers as they require many parameters [26]. Alternate techniques which rely on ML have been proposed by researchers, a few of which are discussed in this subsection.

The authors in [27] aimed to predict the levels of Exchangeable Sodium Percentage (ESP), Magnesium Adsorption Ratio (MAR), Potential Salinity (PS), Residual Sodium Carbonate (RSC), Sodium Adsorption Ratio (SAR), and Total Dissolved Solid (TDS) in irrigation water using ML models. Their work showed that Adaboost and Random Forest (RF) were good predictors, but Artificial Neural Network (ANN) and Support Vector Machine (SVM) were less sensitive to input variables. In [26] the authors also proposed a model for determining the quality of water for irrigation purposes using three parameters - sodium, chloride, and EC. The work started off with five water parameters – sodium, chloride, EC, bicarbonate, and SAR, which were then reduced to the final three using correlation models. They then compared the classification performance of various machine learning models on these three parameters and obtained results that showed that Random Forest performed the best, when compared to Decision Trees, Naïve Bayes, Gradient Boosting, SVM and ANN. In a similar work, IWQI was calculated in [28] using a model proposed in [29] with sodium, chloride, EC, bicarbonate, and SAR as parameters. Singh *et al.* [30] also considered the SAR, Sodium level, Kelly's Index (KI) and permeability index (PI) to determine IWQI using regression and ANN models. The regression models were used to determine the correlation between water parameters, while ANN models were used for the prediction.

A commonality among works on irrigation water is the term irrigation water quality index (IWQI), which is an index used to measure the quality of water for crop irrigation. It takes into consideration the individual contribution and relative weight of each water parameter when classifying water samples for irrigation [29]. There are various approaches to calculating IWQI, notable among which are the WQI approach proposed by WHO (as used in [10], [24], [31]) and that of Meireles *et al.* [29]. For this work, we stick to WHO's approach, which we also use for assessing the quality of drinking water. However, for completeness purposes, we summarize the steps of Meireles *et al.*'s approach in Algorithm 2.

It is important to note that though WQI and IWQI are widely used in literature, they have their limitations. An obvious disadvantage of combining water index for irrigation is that the specific effect of each water parameter is somewhat masked. For instance, sodium is known to affect soil dispersion as it reduces infiltration by increasing SAR. It is also toxic if sprayed on leaves through irrigation sprinkler. On the other hand, $NO_3$ and $PO_4$, may be beneficial for irrigation as they are nutrients required by plants. These causes and effects of individual constituents are masked off when WQI/IWQI are used.

From the reviewed literature two major inferences can be drawn, which are:

1) In many of the works that proposed a "network" for water monitoring, the actual network architecture was not shown. Most authors simply stated that a certain number of monitoring stations were setup to measure water parameters. This is probably because, the actual analyses were carried out in laboratories and not on site. Furthermore, details about the communication technologies, communication media and protocols used were not discussed. This work seeks to fill this research gap.
2) Most of these studies split water parameters into physical (temperature, cloudiness, etc.), chemical (pH, carbonate, nitrate levels, etc.), biological (presence of bacteria, virus, etc.), heavy metals (lead, cobalt, etc.) and others; and applied one or two WQI (or IWQI) models to determine water quality. The application of machine learning models, which are economically viable options for interpreting water sample analyses, is still in its infancy, especially in the context of developing nations. This is another interesting research gap which this study attempts to fill.

## III. THE WATER MONITORING NETWORK

As earlier stated, one of the objectives of this work is to develop a realistic network for monitoring water parameters in real-time. This network which we term "WaterNet" is based on a LPWAN technology and is intended to support a Cyber-Physical System for Water (CPS-W). CPS-W, like most CPS [32] combines an IoT-based sensing and actuation subsystem with Fog/Cloud computing [17]. This combination has been used in numerous applications, such as in health [33], transportation [34], [35], and environmental monitoring [36]. The nodes in WaterNet would be wirelessly interconnected by a two-layer LoRa network. LoRa (Long Range) is a type of LPWAN that emphasizes power conservation over data transmission rate [19]. It has been shown that by using LoRa, data can be transmitted up to a range of 300 km in ideal situations (clear line of sight, good antenna height, antenna gain, transmission power and transmission frequency) but at the cost of data bandwidth [37]. Due to the small size of telemetry data being exchanged across WaterNet nodes, only minimal bandwidth is required, hence, LoRa is suitable for our application.

### A. CAPE TOWN WATER SYSTEM

WaterNet is being proposed for monitoring water parameters in Cape Town, a city in the Western Cape province of South Africa. There are fifteen major water storage dams that supply water to the City of Cape Town (CCT) and its immediate environs. Eleven of these dams are owned by the CCT, while the other four are owned by the Department of Water and Sanitation [38]. Figure 2 shows a high-level depiction of the locations of the dams across the city. This work focuses on the 11 dams owned by CCT and develops a network model for monitoring water quality parameters for drinking and irrigation purposes. It is important to note that, beyond monitoring "fitness for use" for drinking and irrigation, this system can also be used to monitor water levels as well as usage and refill rates of the dams. Though a few of these dams have monitoring systems in place, most are either manually operated or are stand-alone systems. Our objective is thus, to develop a city-wide water quality monitoring network that interconnects all the dams and enables real-time online monitoring of water parameters across them.

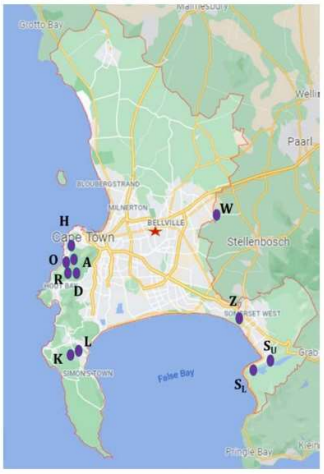



**FIGURE 2.** Locations of CCT-owned dams across the City of Cape Town, South Africa. Where A = Alexandra Resv. Dam; D = DeVilliers Dam; H = Hely-Hutchinson Resv. Dam; K = Kleinplaas Dam; L = Lewis Gay Dam; O = Woodhead Dam; SL = Steenbras Dam – Lower; SU = Steenbras Dam – Upper; R = Victoria Dam; W = Wemmershoek Dam; Z = Land-en-Zeezicht Dam.

### B. WATERNET

The 11 dams considered are connected to 7 Water Treatment Plants (WTP) across the city. Figure 3 gives a visual illustration of our proposed solution, with the dams labelled with alphabets (A, K, L,…), while the WTPs are labelled FN1, FN2…FN7. To monitor water parameters, sensors are installed in each dam to relay telemetry data through edge gateways (GW) to the WTP for processing. The WTPs are considered as Fog Nodes (FN) that can handle some degree of computationally intensive processing, including data aggregation, filtration, basic analysis, and storage. The WTPs (FNs) are in turn connected to a central location, in our case the ILLIFU Cloud computing research facility, located at the University of the Western Cape (UWC), where advanced computing activities and data warehousing take place [39].

## IV. ASSESSING WATER QUALITY

The purpose of WaterNet is to gather data on water parameters from dams across the city. These parameters are then used to assess the quality of water with regards "fitness for use" for drinking and irrigation purposes. In this work, rather than relying on instrumental and physico-chemical analysis carried out in laboratories to assess water parameters, we propose the use of machine learning (ML) models, which take the numerous water parameters into consideration and automatically determine if a sample of water is potable or fit for agricultural use. The motivation is to reduce the cost and complexities involved in collecting, testing, and analysing water samples to determine their status. By using ML and transfer learning, a pre-trained ML model from one site can be transferred to another location and results would be obtained in minutes.

Figure 4 is a flow chart depicting the methodology adopted, with each of the phases discussed as follows.

### A. DATA CURATION

Like most research on ML a dataset is required. However, due to the absence of large, dedicated, and open access datasets of drinking and irrigation water, especially in Africa, we created our own. To create our datasets, we aggregated several "small" datasets of water for drinking and irrigation (or agriculture) primarily from Elsevier's Data in Brief (DiB). DiB is an open access journal dedicated to publishing details on research data [40]. We used the following search phrases "irrigation water," "potable water," "groundwater," and "drinking water," then filtered out unrelated articles. We ended up with 11 publications (mostly from Asia), 7 of which also had data on irrigation water. The datasets were scraped, combined, and saved into two csv files, for drinking and irrigation respectively, using Microsoft Excel.

For our work, the primary requirement was to have data that could be used to train (and test) our ML models to classify water samples. Ideally, a water monitoring network would have been the source of these data on water parameter, however, since to our knowledge, no such data aggregation network exists, we had to improvise. At this stage we are less concerned about the source of the data, as this work is simply a proof of concept, instead we concerned ourselves with ensuring that the respective datasets contained relatively similar feature sets (water parameters). This is similar to what was done in [24]. Tables 1 and 2 show a comparison of features (water parameters) across the different publications considered.

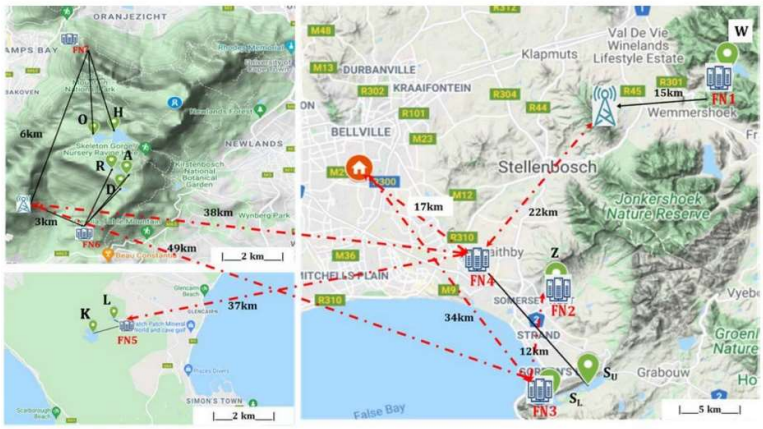



**FIGURE 3.** Proposed dam monitoring network for the City of Cape Town - WaterNet. Where FN1 = Wemmershoek WTP; FN2 = Helderberg WTP; FN3 = Steenbras WTP; FN4 = Faure WTP; FN5 = Brooklands WTP; FN6 = Constantia Nek WTP; FN7 = Kloof Nek WTP.

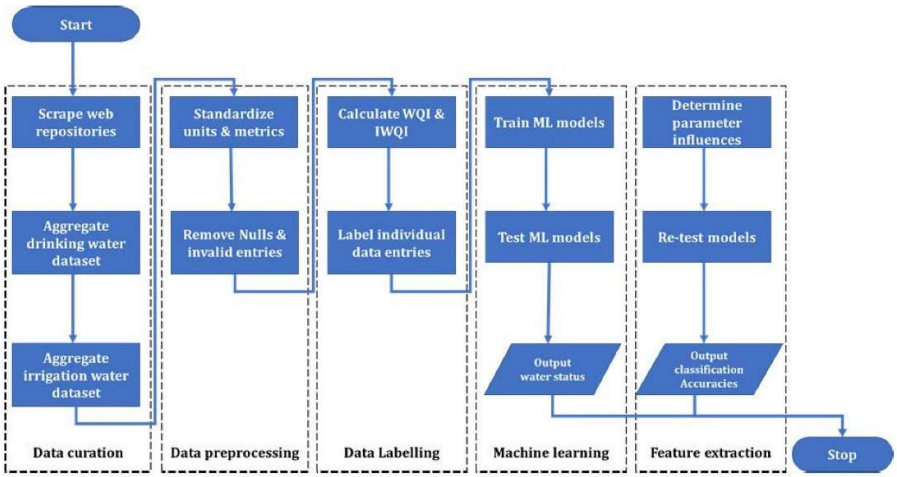



**FIGURE 4.** Process flow for Water Quality Assessment using ML.

Table 1 only shows the most common features across the datasets considered. Some of the datasets contained additional parameters such as nitrate, fluorine in [41]; nitrate, Florine, Salinity in [42]; Total Prec. Potential, Turbidity, Colour, Total coliform, E. coli, organic carbon, chlorophyll, nitrites, ammonium, phosphate, and iron in [43]; and Colour, nitrite, ammonia, zinc, barium, boron, copper, iron, lead, mercury etc. in [44]. These parameters were excluded from Table 1 because we only found at most 2 papers that had them in common.

In Table 2, the features marked "XX" were not included in the original dataset but were calculated by us using the constituent parameters and formula on Table 3. Table 3 also shows the definition of the acronyms used in Tables 1 and 2, and the unit of each feature. After aggregation we ended up with two datasets containing approximately

**TABLE 1.** Feature comparison for potable water.

| Ref. | pH | Na | Mg | Ca | Cl | K | CO3 | HCO3 | SO4 | F | Turbidity | TDS | EC | TH | Label |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [31] | X | | X | X | X | X | | | X | | | X | X | X | |
| [41] | X | X | X | X | X | X | | X | X | X | | X | X | X | X |
| [42] | | X | X | X | X | X | | X | X | X | | | | | |
| [43] | X | X | X | | X | X | X | | X | | X | | X | X | |
| [44] | X | X | | | X | | | | X | X | X | X | X | | |
| [45] | X | X | X | X | X | | | | X | | | X | X | X | |
| [46] | X | X | X | X | | | X | | | | X | X | | | |
| [47] | X | X | X | X | X | X | | X | X | | | X | X | X | X |
| [48] | X | X | X | X | X | | X | X | X | | | X | X | X | |
| [49] | X | X | X | X | X | X | X | X | X | | | X | X | X | |
| [50] | X | | | | | | | | | | X | | X | | |

**TABLE 2.** Feature comparison for irrigation water.

| Ref. | RSC | PI | KR | MH | Na% | SAR | SSP | Label |
|---|---|---|---|---|---|---|---|---|
| [31] | | X | X | X | X | X | X | |
| [41] | X | X | X | X | X | X | XX | |
| [42] | X | X | X | X | X | X | X | |
| [46] | | XX | XX | XX | X | X | XX | X |
| [47] | X | X | X | X | X | X | X | |
| [48] | X | X | X | X | X | X | X | |
| [49] | X | X | X | X | X | X | X | |

700 and 360 unique entries for drinking water and irrigation water respectively.

### *B. DATA PRE-PROCESSING & LABELLING*

Of the entire aggregated drinking water dataset, only about 16% were pre-labelled (i.e., the status of the water sample was included in the dataset). We then wrote a Python script to calculate the WQI (and IWQI) for the unlabelled data. From literature, researchers often assign different weights to the individual water parameters to counter the masking effect of WQI. However, because we are building a generic model, we assigned equal weights to all parameters in our Python script to avoid introducing any form of bias. For the drinking water dataset, we cross-referenced the calculated WQI value with those defined in [7] and [44]. If the calculated WQI $< 50$, the data entry was labelled 1 (i.e. potable) or 0 (non-potable) if otherwise. For the irrigation water dataset, we also set the threshold to 50, as such IWQI values $>= 50$ were considered permissible for irrigation. Hence, we labelled data entries with IWQI $< 50$ as not suitable for irrigation (0) and values $>= 50$ as usable (1).

Kindly note that the threshold value of 50 was only used as a general guide for assessing fitness of use. This value (i.e. 50) has also been used in several literature [42], [45]–[47] to indicate water of good (or excellent) quality. Indeed, the overall WQI may indicate that a sample of water is fit for use, but there may be some constituents beyond this threshold levels which are not captured, e.g., toxicity. Table 3 summarizes the acceptable value range for each parameter as used in our labelling script, while the process of calculating WQI and IWQI are discussed in the next subsections.

#### 1) CALCULATING WQI FOR DRINKING WATER

Water Quality Index (WQI) is a simple dimensionless index for assessing the quality of water based on various parameters [6]. There are numerous ways of determining the WQI based on different models as discussed in the introductory section. In this work we apply the method proposed by Horton [51] which is summarized in Algorithm 1

All the steps in Algorithm 1 are relatively straightforward except step 1. Feature selection has been a long-battled

**TABLE 3.** Acceptable range for various water parameters [7], [44].

| Parameter | Definition | Unit/Formula | Accepted Range |
|---|---|---|---|
| pH | Potential of Hydrogen | | 5 – 9.7 |
| Na | Sodium | mg/L | <= 200 |
| Mg | Magnesium | mg/L | < 50* |
| Ca | Calcium | mg/L | < 75* |
| Cl | Chloride | mg/L | <=300 |
| K | Potassium | mg/L | < 12* |
| SO4 | Sulphate | mg/L | <= 500 |
| HCO3 | Bicarbonate | mg/L | 120-200* |
| CO3 | Carbonate | mg/L | 1% of HCO3* |
| Turbidity | Cloudiness | NTU | <= 5 |
| TDS | Total Dissolved Solids | mg/L | <= 1200 |
| EC | Electrical Conductivity | mS/m | <= 170 |
| TH | Total Hardness | mg/L | 100 – 300* |
| RSC | Residual Sodium Carbonate | $(HCO_3 + CO_3) - (Ca + Mg)$ | <1.25* |
| PI | Permeability Index | $\frac{Na + \sqrt{(HCO_3)}}{(Ca + Mg + Na)} 100$ | >70* |
| KR | Kelly's Ratio | $\frac{Na}{(Ca + Mg)}$ | <1.5* |
| MH | Magnesium Hazard | $[Mg/(Ca + Mg)]$ × 100 | <50* |
| Na% | Sodium Percentage | $\frac{(Na + K)}{(Ca + Mg + Na + K)}$ ×100 | <40* |
| SAR | Sodium Adsorption Ratio | $\frac{Na}{\sqrt{(Ca + Mg)/2}}$ | 0-10 |
| SSP | Soluble Sodium Percentage | $\frac{Na}{(Ca + Mg + Na)} x100$ | <50* |

**Algorithm 1** Calculating WQI for Drinking Water.

1. Select relevant parameters

$$(P = [P_1, P_2, P_3 \ldots P_n]).$$

2. Assign weights to each parameter ($w_p$), $1 < p < n$
3. Calculate relative weight

$$W_p = \frac{w_p}{\sum_{p=1}^{n} w_p}$$

4. Calculate quality index

$$q_p = \frac{C_p}{S_p * 100}$$

5. Obtain

$$WQI = \sum_{p=1}^{n} W_p * q_p$$

where P = parameter selected, $w_p$ = weight of parameter p, n = number of parameters, $C_p$ = concentration of p, $S_p$ = standard value for parameter p as stipulated by WHO [7].

**Algorithm 2** Calculating WQI for Irrigation Water [29].

1. Identify prominent parameters in the sample, i.e. EC, sodium, chloride, bicarbonate, SAR.
2. Determine weights for each parameter.

2a. Calculate quality measurement value

$$q_i = q_m ax - \frac{(X_{ij} - X_{inf}) * q_{iamp}}{X_{amp}}$$

2b. Calculate aggregate weight

$$w_i = \frac{\sum_{j=1}^{k} F_j * A_{ij}}{\sum_{j=1}^{k} \sum_{i=1}^{n} F_j * A_{ij}}$$

3. Obtain

$$IWQI = \sum_{i=1}^{n} q_i * w_i$$

where $q_{max}$ = max value of $q_i$ in its class; $X_{ij}$ is the value of parameter $i$; $X_{inf}$ is the lowest value in the class to which $X_{ij}$ falls; $q_{iamp}$ = class amplitude; $X_{amp}$ = amplitude of $X_{ij}$'s class; $w_i$ = parameter weight; $F$ = autovalue of the first component; $A_{ij}$ = explainability of parameter $i$ by $j$; $j$ = factor count. Details about each variable can be found in [29].

challenge due to various viewpoints on which parameters are most important, especially across different geographical domains. For instance, a certain parameter that might be considered critical in one country, because it is naturally present in their water bodies, might not be relevant in another country, where such elemental parameter is absent. To this end, various countries have developed their own models for measuring WQI. Common among these include the Canadian CCMEWQI, India's BIS, the U.S. Environmental Protection Agency (EPA), the South African SANS 241-1, and the global model by the World Health Organization (WHO).

#### 2) CALCULATING IWQI FOR IRRIGATION WATER

As stated earlier, there are two common approaches for calculating IWQI. The first one, which is used in this study, is based on WQI (Algorithm 1), while the second, proposed in [29], is summarized in Algorithm 2.

### C. MACHINE LEARNING MODELS FOR DETERMINING QUALITY OF WATER

As earlier stated, the second objective of this work is to use ML models to automatically classify water samples. We selected the 11 most common water parameters from the dataset sources to run the ML on. These were pH, sodium, magnesium, calcium, chloride, potassium, sulphate, carbonate, TDS, EC, and TH. Three ML classification models were considered, namely Random Forest (RF), Logistic Regression (LR) and Support Vector Classifier (SVC).

RF is the amalgamation of a multitude of decision trees [52]. It can be used for both regression and classification problems. When used as a classifier it outputs the "majority vote" from all the individual trees. Unlike decision trees (DT), RF generally does not suffer from over-fit on training data. It also uses bagging and random feature selection to overcome the high variance problem of DT [53]. LR models the probability that an event would occur, called the dependent variable based on one or more independent variable(s). LR is well suited to finding binary output probabilities, i.e., True or False (1 or 0) and it does not require a linear relationship between the dependent and independent variables. In applying it to this work, the 11 features were considered the independent variables, while the potability (1 or 0) was the dependent variable. SVC is a form of Support Vector Machine, which is a nonlinear solver for classification and regression problems [54]. A unique advantage of SVM over models such as Neural Network is its ability to perform well with smaller datasets, hence our decision to use it. Given a set of data points, SVM seeks to draw a line (hyperplane) that separates the data point into unique classes. Typically, the hyperplane must maximise the distance from support vectors of each class with the smallest possible data separation error. For this work we used a linear kernel with our SVC model.

For this work, water assessment is considered a classification problem, with the primary objective of classifying water samples into "fit for use" or not. LR was used because it is well suited for binary classification problems using the sigmoid function, though it is susceptible to outliers. On the other hand, SVC was considered because, like LR, it is also well suited for two-class problems but less affected by outliers. It also works well with smaller datasets, which is the case with this work. RF was considered because it can work with datasets of different sizes and with mixed feature sets. It is also generally faster than SVC. Finally, these 3 ML models were chosen because they represent 3 different types

of ML models, viz. LR is based on statistic (regression analysis), SVC is based on data geometry, while RF is a type of ensemble learning ML model.

### D. DETERMINING PARAMETER INFLUENCE

Here the goal is to dig deeper into the classification problem to determine which features (water parameters) are the most influential in determining water potability (or irrigation suitability) when using ML models. We utilized the recursive feature elimination (RFE) method to achieve this, similar to the approach used in [55].

RFE is a backward feature selection method that searches for the best performing features by first utilizing the entire feature set to train a given model. It then scores each feature based on its contribution to the overall performance of the model, after which it iteratively removes poor performing features and retrains the model, until further removal of features does not improve the model's performance [56]. In this work we used accuracy as the scoring factor and criteria for eliminating features.

## V. IMPLEMENTATION

Our implementation process was split into two phases – A and B. Phase A focused on WaterNet – the water monitoring network, while Phase B focused on assessing the quality of water based on water parameters received from WaterNet.

### A. PHASE A - SIMULATING WATERNET

To simulate this city-wide water monitoring network (WaterNet) a combination of Google maps, Topographic-map.com and Radio Mobile software [57] was used. Google maps is a web-based mapping and real-time location sharing service by Google [58], while Topographic-map.com is a free online tool which provides details about the geographical landscape of an area including hills, mountains, and valleys [59]. Radio Mobile is a network planning tool for simulating radio frequency propagation [57]. It uses an irregular terrain propagation model to simulate coverage and point-to-point transmissions of radio signals. This terrain propagation feature makes Radio Mobile ideal for our application because Cape Town is a city with many undulating plains, with several lowlands sandwiched between mountains and hills. This uneven geography of the city makes direct line of sight radio propagation difficult, and creates an interesting networking challenge, as most communication frequencies do not propagate through rocks and/or mountains. Radio Mobile is therefore ideal for testing reachability, signal strengths and line of sights of radio propagations in WaterNet.

We began by creating a custom map in Google map with all relevant points of interest marked. This was then imported into Radio Mobile as a KML file, with all coordinates embedded. In Radio Mobile, a 2-layer hierarchical network model was created. At the lower level the dams were connected to their respective WTPs (FNs) using LoRa network configured with frequency range of 863-870 MHz, transmission power of 14 dBm, receiver threshold of −80 dBm, and 10m high antennas. At the higher level, the FNs were connected to ILLIFU Cloud data centre using a 2.4 GHz LoRa network [60] configured with frequency ranging between 2.41-2.46 GHz, transmission power of 22 dBm, receiver threshold of −75 dBm, antenna gain of 21 dBi and height of 30 m. Figure 5 shows the 2-layer Cyber Physical hierarchical network of WaterNet, where X-GW are the gateways (edge devices) at each dam and FN1…FN7 are the WTPs hosting the Fog Nodes. To get more range we used a high spreading factor of 12. Ideally, higher spreading factor results in lower data rates [19] but this is acceptable in our use case, as we would only be sending small sized telemetry data at pre-set intervals.

### B. PHASE B - ASSESSING WATER QUALITY

With the WaterNet established and telemetry data on water parameters being sent to the Fog and Cloud respectively, data analysis and ML can be used to gain useful insights or assess the quality of water at each dam. For this work, we curated data from different sources to simulate parameters received from WaterNet. Simulations presented in this section thus focus on the use of ML to assess quality of water for drinking or irrigation purposes.

Experimental simulations were carried out on Google Colab, with a Python 3 Google Compute module, configured with 12 GB of RAM, and 2.3 GHz 2 Core Intel Xeon CPU. Sci-Kit learn was used for the ML models, Pandas and NumPy for data manipulation, while matplotlib was used for data visualization. Finally, the dataset was split into 84% training and 16% test data.

Three 3 ML models were considered and contrasted against each other using five metrics, namely accuracy, true positive (TP), false positive (FP), false negative (FN) and true negative (TN). Of these five metrics, accuracy, FP, and FN were the most critical to us. Accuracy is a measure of a model's classification performance, i.e., the percentage of water samples that were correctly classified. False positive is the percentage of impure water samples that were misclassified as potable. This is important because misclassifying non-potable water as drinkable can be hazardous with severe consequences to the health if consumed. False negative on the other hand is a measure of the percentage of potable traffic that were wrongly classified as not safe for consumption.

## VI. RESULTS & DISCUSSIONS

In line with the previous section on implementation, our results are also presented in two phases, the first focuses on the results of the water monitoring network (WaterNet), while the second focuses on assessing water quality for drinking and irrigation purposes using ML.

### A. WATER MONITORING NETWORK

Table 4 summarizes the results and observations from simulating the network on Radio Mobile. It can be observed that not all FNs can directly (1 hop) reach ILLIFU, in fact, only 2 (FN3 and FN4) are able to. Thus, a point-to-point star network

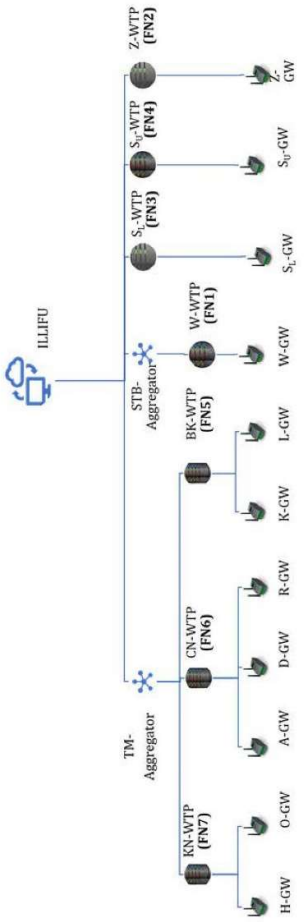



**FIGURE 5.** Hierarchical Cyber Physical Network for WaterNet.

topology cannot be used, instead a partial mesh with node rebroadcast is considered. FN1 is located close to the ground level and almost fully surrounded by high grounds, hence the need for a repeater (STA) mounted on a 1136 m high hilltop at Stellenbosch farms (-33.90989, 18.74262). Despite this repeater, FN1 (via the repeater) is still unable to directly reach ILLIFU but has FN4 in its line of sight. FN1 can therefore reach ILLIFU by hopping through STA and FN4.

Like FN1, FN2's LOS to ILLIFU is obstructed by a hill and has to be rebroadcasted via FN3. FN5 requires an antenna of about 40-50 m height to reach FN4, from where its signal is rebroadcasted to ILLIFU. FN6 and FN7 are somewhat isolated and unreachable by all FNs, because they are located behind Table Mountain. To allow reachability to both sites, a repeater (TMA) is placed on a hill around Hout Bay in Cape Town.

Figure 6 is a snapshot of the partial mesh network extracted from Radio Mobile. The figure reveals that most traffic traverse through FN3 and FN4, hence were the most critical nodes in the network. A reasonable explanation for this is that both FN3 and FN4 have clear line of sight to ILLIFU, as there are no high rise geographical structure on their paths.

### 1) LINK COSTS

Though IEEE 802.15.4 standard stipulates an "Rx sensitivity" value of -85dBm for both the 868/915 MHz and 2.4 GHz networks [61], commercially values of up to -100 dBm are acceptable. W.r.t signal voltage, the stipulation is 12.6 uV (calculated using Eq. 3), which implies that radio frequency signals that arrive at a receiver with a root mean square voltage of at least 12.6 uV can be detected with about 99% accuracy (less than 1% data error) [62]. From Table 5, almost all paths have a sensitivity value greater than -85dBm and a signal voltage greater than 12.6 uV. This implies reachability and less than 1% data error rate. The only exceptions are FN1-SMA, FN6-TMA and FN7-TMA with values sensitivity values lower than -85 dBm.

Table 5 summarizes the obtained link cost results from Radio Mobile. On the table, "Tx height" and "Rx height" respectively mean the antenna heights of the transmitter and receiver from ground level. Path loss (or path attenuation) means the reduction in power of the radio waves as they propagate through free space, resulting from reflection, refraction, absorption, etc. It is calculated using (1). Finally, "Rx Sensitivity" and "Signal Voltage" are indicators of the sensitivity of a receiver on a network path. Receiver sensitivity is the minimum input signal required to overcome noise and produce an acceptable output signal with less than 1% packet error rate at the receiver [62]. Receiver sensitivity and signal voltage are calculated using (2) and (3).

$$PathLoss = 20 * log(d) + 20 * log(f) + 20 * log(\frac{4 * \pi}{c}) - G_T - G_R \quad (1)$$

where d = distance between both transmitter and receiver antennas, f = signal frequency, c = speed of light, $G_T$ and $G_R$ are the gains of the transmitter and receiver antennas respectively.

$$ReceiverSensitivity(S_r) = \frac{S}{N}_{min} * K * T * B * \frac{NF}{G} \quad (2)$$

$$SignalVoltage(uV) = \sqrt{R * 10^{\frac{S_r - 30}{10}}} \quad (3)$$

where $S_r$ = receiver sensitivity, $S/N_{min}$ = Minimum signal-to-noise ratio required to detect a signal, $K$ = Boltzmann's Constant ($1.38 * 10^{-23} Joule^o K$), $T$ = absolute temperature of receiver in Kelvin ($290^o K$), $B$ = Receiver Bandwidth (Hz), $G$ = antenna gain of receiver, and $R$ = Resistance of the antenna.

**TABLE 4.** Observations from simulating waternet on radio mobile.

| Node | Location | Dam(s) /Node(s) | Nodes in Line of Sight | Direct link to ILLIFU | Route to IL-LIFU | Hops to ILLIFU | Comment |
|---|---|---|---|---|---|---|---|
| FN1 | -33.83471,19.07271 | W-GW | None | No | Via STA and FN4 | 3 | Obstructed by hills around stellenbosch |
| FN2 | -34.06278,18.87325 | Z-GW | FN3 | No | Via FN3 | 2 | Obstruction by hills around Somerset West |
| FN3 | -34.17483,18.84976 | SL-GW | FN2, FN4, ILLIFU, TM-Aggregator (TMA) | Yes | Yes | 1 | |
| FN4 | -34.03109,18.77176 | SU-GW | FN1, FN3, FN5, ILLIFU, STA, TMA | Yes | Yes | 1 | SU-GW could be sent to FN3 or via fibre optic cables to FN4. |
| FN5 | -34.16912,18.39958 | L-GW, K-GW | FN4 | No | Via FN4 | 2 | Using at least a 45m antenna |
| FN6 | -34.00554,18.39972 | A-GW, D-GW, R-GW | None | No | Via TMA FN4 | 3 | |
| FN7 | -33.94782,18.39513 | H-GW, O-GW | None | No | Via TMA FN4 | 3 | |
| STA | -33.88553,18.92877 | FN1, W-GN | FN1, FN4 | No | Via FN4 | 2 | Co-ordinates: -33.88553, 18.92877 |
| TMA | -33.99611,18.36361 | FN6, FN7 | FN3, FN4 FN6, FN7 | No | Via FN3 or FN4 | 2 | Co-ordinates: -33.99611, 18.36361 |
| ILLIFU | -33.93352,18.62795 | - | FN3, FN4 | - | - | 0 | |

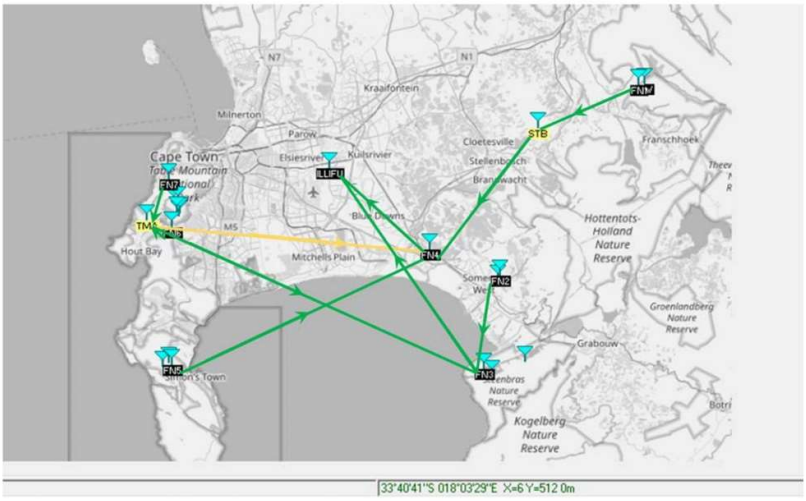



**FIGURE 6.** Snapshot of the partial mesh network topology from Radio Mobile simulator.

We considered two options to interconnect FN1 and SMA. The first is based on 863-870 MHz network, which was used for the lower-level connection as shown in Figure 5. The 863-870 MHz yielded sensitivity values lower than the IEEE stipulations at just 0.1 uV, hence not suitable. As an alternative, we used the 2.4 GHz network instead and obtained a much better signal voltage of 375.9 uV. The links between FN6, FN7 and TMA are sub-GHz and a sensitivity value of up to -100 dBm is acceptable for these kinds of frequency range. Though there is an increased probability of higher error rates, this is acceptable in our use case as telemetry messages being transmitted are very small and occasional re-transmissions would not overwhelm the network.

### 2) OTHER OBSERVATIONS

Figure 5 shows the two-layer network, wherein the dams are connected to their respective fog nodes via an 863-870 MHz LoRa network. It is important to note that this wireless connection might not always be feasible as many of the dams are either located below sea level and obstructed by mountains and hills or at higher altitudes than their respective WTPs. Figure 7 shows four instances of

**TABLE 5.** Inter-nodal link costs.

| | Tx Site | Rx Site | Distance (Km) | Tx Height (m) | Rx Height (m) | Path Loss (dB) | Rx Sensitivity (dBm) | Signal Voltage (uV) |
|---|---|---|---|---|---|---|---|---|
| 1a | FN1 | SMA | 15.3 | 281.7 | 1136.6 | 144.2 | -127.2 | 0.1 |
| 1b | FN1 | SMA | 15.3 | 281.7 | 1136.6 | 122.5 | -55.5 | 375.92 |
| 2 | FN2 | FN3 | 12.64 | 137.9 | 326.8 | 125.8 | -58.8 | 257.58 |
| 3 | FN3 | ILLIFU | 33.72 | 326.8 | 63.3 | 131.5 | -64.5 | 132.61 |
| 4 | FN4 | ILLIFU | 17.13 | 115.3 | 63.3 | 127.8 | -60.8 | 204.05 |
| 5 | FN5 | FN4 | 37.32 | 189.2 | 115.3 | 141.9 | -74.9 | 40.07 |
| 6 | FN6 | TMA | 3.49 | 267.6 | 722.9 | 108.9 | -91.9 | 5.66 |
| 7 | FN7 | TMA | 6.10 | 345.3 | 722.9 | 112.2 | -95.2 | 3.90 |
| 8 | TMA | FN3 | 49.03 | 722.9 | 326.8 | 136.9 | -69.9 | 72.01 |
| 9 | TMA | FN4 | 37.83 | 722.9 | 115.3 | 141.1 | -74.1 | 44.28 |
| 10 | SMA | FN4 | 21.71 | 1136.6 | 115.3 | 131.7 | -64.7 | 130.15 |

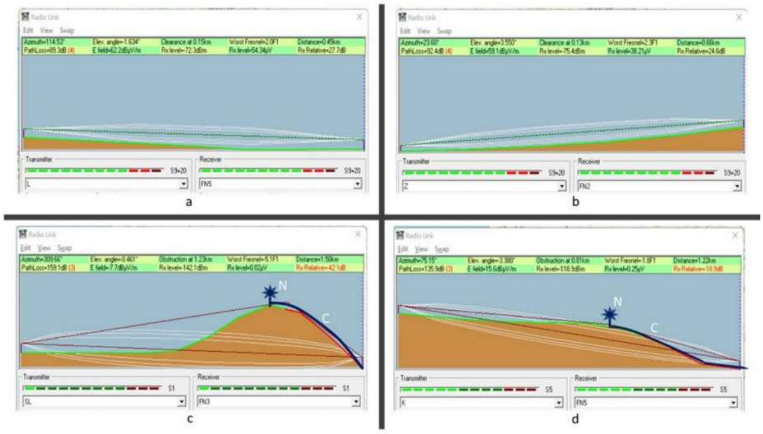



**FIGURE 7.** Snapshot of land structure between FNs and Dams from Radio Mobile simulator.

network connections between (water parameter sensors in) the dams and their corresponding WTP (FN). In the figures the brown coloured structure represents earth/ground, while the blue colour represents free space. The green and red lines represent reachability, while the white lines represent electromagnetic Fresnel zone.

Figures 7a and 7b show a clear line of sight between the dams and FNs. In such situations simple point-to-point network with antennas of height 10-30 m can be set up to connect both entities. Figures 7c and 7d on the other hand are more complicated scenarios, as there is no clear line of sight between the entities. For such cases, we proffer two solutions. In the first, the LoRa node can be placed on an elevated surface (where a line of sight to the FN can be established) and cable(s) used to connect the sensors in the dam to the LoRa node. This is as shown in Figures 7c and 7d, with the LoRa node and cable respectively labelled N and C. The second solution would be to take the parameter readings from the water treatment plant (FN) rather than the dam. Here it is assumed that pre-existing channels are in place to allow water flows directly from the dam to the treatment plant for processing.

### B. CREATING THE DATASET

It is expected that WaterNet would provide data about water parameters, however, for this work and as discussed earlier, we used hypothetical datasets curated data from existing datasets on the Internet. This subsection presents the result of the data curation process.

Using Algorithm 1, we were able to calculate water quality index (WQI) and irrigation water quality index (IWQI) for all data entries. WQI values range from 0 to 100 and were mapped to fitness of use for drinking or irrigation. Figures 8 and 9 show snippets of the final labelled datasets, with the penultimate column in both figures, representing the calculated WQI values, while the last column depicts the fitness of use, as described in section VI-B.

Figures 10 and 11 show the histogram depiction of each parameter in the drinking and irrigation water datasets. For the drinking water dataset, most of the parameters had relatively similar distribution shapes except pH, WQI and potability. The potability curve shows a slight imbalance between potable (300) and non-potable data (400), however, the numbers are close enough not to skew the results. Figures 12(a) and 12(b) show that pH and especially WQI

have a normal distribution curve (using Gaussian kernel), which is indicative of a relatively balanced drinking water dataset. Relatively balanced sample classes were also observed for parameters in the irrigation water dataset, with 175 samples considered suitable for irrigation and 185 considered as not suitable. Figure 12(c) also reveals a normal distribution curve for the calculated IWQI for the irrigation water dataset.

After labelling all data points that were originally unlabelled in the datasets, we used the pre-labelled data as test data, against which we benchmarked our labelled data using several machine learning models (ML). Furthermore, to determine the most influential water parameters, we ran experiments using RFE with cross validation to select the optimal number of features. These features (water parameters) were then ranked in descending order of their influence on the classification accuracy of each model. Finally, we reran the three ML models (RF, LR and SVC) with different combinations of excluded features to determine the change in their accuracies (if any). The ML models and corresponding results are discussed in the next subsection for both drinking and irrigation water.

### C. DRINKING WATER

#### 1) DETERMINING POTABILITY OF DRINKING WATER

Table 6 shows that all 3 models performed well w.r.t accuracy scores. RF had the least accuracy at 96.12% and, though impressive, had the highest False Negative (FP) rate at 5.17%. This implies that RF misclassified hazardous water samples as safe for drinking about 5% of the time. LR and SVC on the other hand resulted in FP values of 0% and are thus better alternatives for RF. However, SVC had a False Negative (FN) rate of 4.23%, implying that it misclassified some potable water samples as not drinkable. LR gave the best results of the 3 models with 99.22% classification accuracy and 1.41% FN. In essence, LR only misclassified safe drinking water as non-potable about 1.5% of the time.

**TABLE 6. Result of model comparison using all features on drinking water dataset.**

| | Model | Accuracy (%) | True Positive (%) | False Positive (%) | False Negative (%) | True Negative (%) |
|---|---|---|---|---|---|---|
| 1 | RF | 96.12 | 94.83 | 5.17 | 2.82 | 97.18 |
| 2 | LR | 99.22 | 100.00 | 0.00 | 1.41 | 98.59 |
| 3 | SVC | 97.67 | 100.00 | 0.00 | 4.23 | 95.77 |

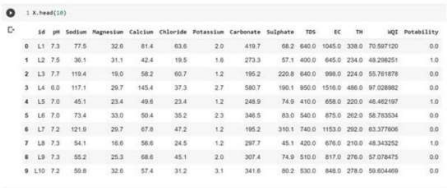
#### 2) DETERMINING PARAMETER INFLUENCE ON DRINKING WATER

Figure 13 shows a graphical depiction of the result of carrying out RFE on each of the models considered, that is, RFE on LR (RFE+LR), RFE on RF (RFE+RF), and RFE on SVC (RFE+SV). The result, though non-uniform, revealed that pH was the least influential parameter across board.

The graph shows that pH has zero influence on the overall classification accuracy, and this is expected as Figure 12(a) and Table 3 allude to this. From Table 3, pH values ranging between 5 and 9.7 are considered safe for consumption and since almost all pH values in our dataset fall within this range (as shown in Figure 12(a)), pH would therefore have minimal effect on the classification accuracy. A similar explanation holds true for sulphate ($SO_4$) with most of the values in our dataset being within the safe limit (<= 500mg/L). These are in sharp contrast to magnesium (Mg) for instance, which according to Table 3, should have a value of less than 50mg/L, but our drinking water dataset has numerous entries with values ranging between 100 – 750mg/L, hence the heavy influence of Mg.

To further examine the influence of different combinations of parameters on the classification accuracies of each model, we ran iterative experiments using all possible combinations of parameters. For each iteration we held one parameter constant and cycled through the other 10. Table 7 summarizes the results of the top 40 combinations for LR, RF and SVC respectively. For each model, the table shows the resulting classification accuracies when at least two water parameters are removed from the dataset.

Table 7 and Figure 14 further buttress our results in Figure 13, that pH had the least effect on the classification accuracies of all 3 models, while magnesium (and EC) were the most influential parameter for drinking water.



| | Id | pH | Sodium | Magnesium | Calcium | Chloride | Potassium | Carbonate | Sulphate | TDS | EC | TH | WQI | Potability |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | L1 | 7.3 | 77.5 | 32.6 | 81.4 | 63.6 | 2.0 | 419.7 | 68.2 | 640.0 | 1045.0 | 338.0 | 70.597120 | 0.0 |
| 1 | L2 | 7.5 | 36.1 | 31.1 | 42.4 | 19.5 | 1.6 | 273.3 | 57.1 | 400.0 | 645.0 | 234.0 | 49.208251 | 1.0 |
| 2 | L3 | 7.7 | 119.4 | 19.0 | 58.2 | 60.7 | 1.2 | 195.2 | 220.8 | 640.0 | 998.0 | 224.0 | 55.761878 | 0.0 |
| 3 | L4 | 6.0 | 117.1 | 29.7 | 145.4 | 37.3 | 2.7 | 580.7 | 190.1 | 950.0 | 1516.0 | 486.0 | 97.028982 | 0.0 |
| 4 | L5 | 7.0 | 45.1 | 23.4 | 49.6 | 23.4 | 1.2 | 248.9 | 74.9 | 410.0 | 658.0 | 220.0 | 46.462197 | 1.0 |
| 5 | L6 | 7.0 | 73.4 | 33.0 | 59.4 | 35.2 | 2.3 | 346.5 | 83.0 | 540.0 | 875.0 | 262.0 | 58.763534 | 0.0 |
| 6 | L7 | 7.2 | 121.9 | 29.7 | 67.8 | 47.2 | 1.2 | 195.2 | 310.1 | 740.0 | 1153.0 | 292.0 | 63.377606 | 0.0 |
| 7 | L8 | 7.3 | 54.1 | 16.6 | 56.6 | 24.5 | 1.2 | 297.7 | 45.1 | 420.0 | 676.0 | 210.0 | 48.343252 | 1.0 |
| 8 | L9 | 7.3 | 55.2 | 25.3 | 68.6 | 45.1 | 2.0 | 307.4 | 74.9 | 510.0 | 817.0 | 276.0 | 57.078475 | 0.0 |
| 9 | L10 | 7.2 | 50.8 | 32.6 | 57.4 | 31.2 | 3.1 | 341.6 | 80.2 | 530.0 | 846.0 | 278.0 | 59.604469 | 0.0 |

**FIGURE 8. Snippet of labelled drinking water dataset showing calculated WQI and potability values for the first 10 data points. Refer to units in Table 3.**

| | id | RSC | PI | KR | MH | Na | SAR | SSP | EC | IWQI | USABLE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0.12 | 59.2 | 0.5 | 39.8 | 33.6 | 1.8 | 33.3 | 1045 | 63.379762 | 1.0 |
| 1 | 2 | -0.21 | 58.9 | 0.3 | 54.8 | 25.6 | 1.0 | 25.1 | 645 | 50.337302 | 1.0 |
| 2 | 3 | -1.28 | 72.2 | 1.2 | 35.0 | 53.8 | 3.5 | 53.7 | 998 | 68.144246 | 1.0 |
| 3 | 4 | -0.20 | 55.2 | 0.5 | 25.2 | 34.7 | 2.3 | 34.4 | 1516 | 64.631448 | 1.0 |
| 4 | 5 | -0.33 | 62.5 | 0.4 | 43.8 | 31.1 | 1.3 | 30.8 | 658 | 51.432242 | 1.0 |
| 5 | 6 | 0.43 | 66.0 | 0.6 | 52.0 | 38.2 | 2.0 | 37.8 | 875 | 71.445437 | 1.0 |
| 6 | 7 | -2.64 | 63.6 | 0.9 | 42.0 | 47.7 | 3.1 | 47.6 | 1153 | 51.132837 | 1.0 |
| 7 | 8 | 0.68 | 69.6 | 0.6 | 32.6 | 36.2 | 1.6 | 35.9 | 676 | 65.043849 | 1.0 |
| 8 | 9 | -0.48 | 58.7 | 0.4 | 37.9 | 30.7 | 1.4 | 30.3 | 817 | 50.248115 | 1.0 |
| 9 | 10 | 0.04 | 60.9 | 0.5 | 48.4 | 32.5 | 1.6 | 31.9 | 848 | 60.917361 | 1.0 |

**FIGURE 9. Snippet of labelled irrigation water dataset showing calculated IWQI and usability values for the first 10 data points. Refer to units in Table 3.**

### D. IRRIGATION WATER

#### 1) DETERMINING USABILITY OF IRRIGATION WATER

Similar to the results on Table 6, Table 8 also shows that RF performed the worst of all three models w.r.t. to FP with a

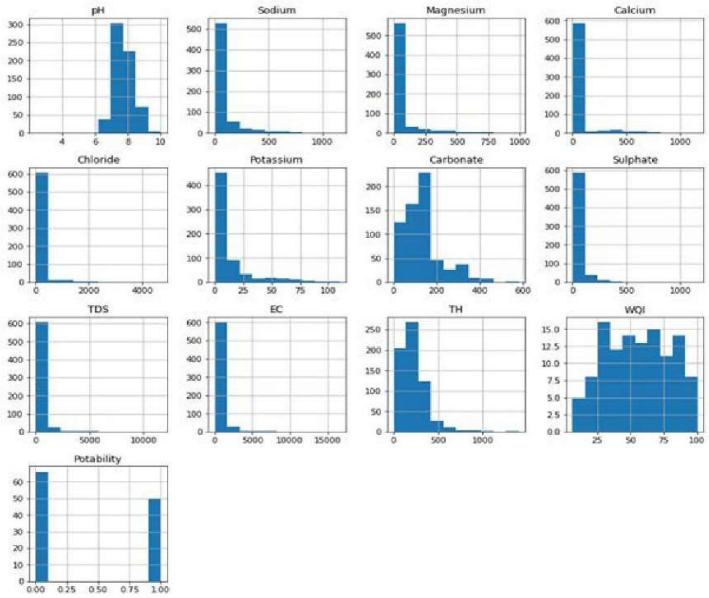



**FIGURE 10.** Histogram plot (count vs. value) of water parameters in the drinking water dataset.

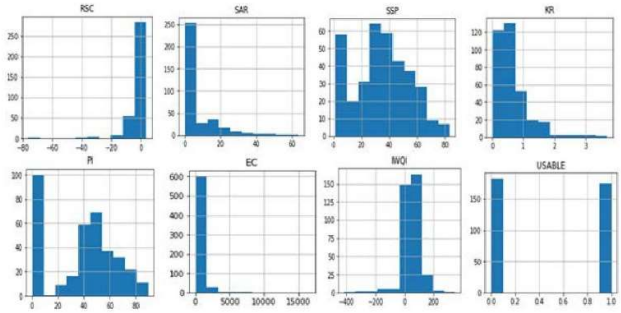
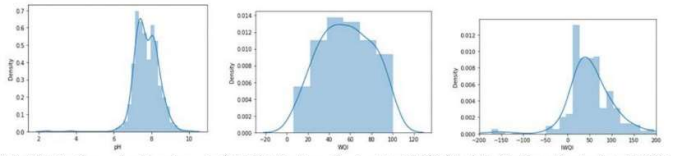



**FIGURE 11.** Histogram plot (count vs. value) of water parameters in the irrigation water dataset.

score of 8.33%. The same trend as in Table 6 is also observed for LR and SVC, with both having the lower FP rates of 5.56% and 5.50% respectively. However, in contrast to the results of the drinking water dataset, LR performed the worst w.r.t False Negative (FN) at 11.11%. The effect of FN are not as adverse on health as FP, hence, SVC would be considered the best option for irrigation water, as it gave acceptably high classification accuracy and the lowest False Positive value.

### 2) DETERMINING PARAMETER INFLUENCE IN IRRIGATION WATER

Figure 15 shows a graphical depiction of the results of recursive feature elimination (RFE+LR, RFE+RF, and RFE+SVC) on the irrigation water dataset. It reveals that SSP had the least influence on the classification accuracies of the models, while RSC was the most influential feature (water parameter). SAR and Na were also relatively influential across board. EC is seen to be very influential with



(a) Distribution of pH values in drinking water dataset (b) Distribution of calculated WQI values in drinking water dataset (c) Distribution of calculated IWQI values in irrigation water dataset

**FIGURE 12.** Density distribution curves for select water parameters.

**TABLE 7.** Parameter influence on classification accuracies of ML models for drinking water.

| Excluded Parameters | LR Accuracies (%) | RF Accuracies (%) | SVC Accuracies (%) |
|---|---|---|---|
| pH & TH | 99.22 | 98.45 | 97.67 |
| Carbonate & Na | 96.90 | 98.45 | 96.90 |
| Mg & Na | 96.12 | 98.45 | 96.90 |
| Chloride & Ca | 96.12 | 94.57 | 95.35 |
| Carbonate & Ca | 96.12 | 94.57 | 95.35 |
| Carbonate & Chloride | 96.12 | 94.57 | 94.57 |
| TDS & Na | 96.12 | 98.45 | 97.67 |
| TDS & Ca | 96.12 | 93.80 | 94.57 |
| pH & Mg | 95.35 | 96.12 | 94.57 |
| Chloride & SO4 | 95.35 | 95.35 | 95.35 |
| pH & Ca | 94.57 | 95.35 | 93.80 |
| pH & Chloride | 94.57 | 95.35 | 94.57 |
| pH & SO4 | 94.57 | 96.12 | 96.12 |
| Chloride & Na | 94.57 | 94.57 | 94.57 |
| Chloride & Mg | 94.57 | 96.12 | 94.57 |
| pH & Carbonate | 93.80 | 96.12 | 96.12 |
| Mg & K | 93.02 | 93.80 | 93.02 |
| Carbonate & K | 93.02 | 92.25 | 93.02 |
| TDS & K | 93.02 | 92.25 | 93.02 |
| TH & Na | 93.02 | 98.45 | 93.80 |
| TH & Mg | 93.02 | 96.12 | 93.02 |
| pH & K | 92.25 | 95.35 | 94.57 |
| TH & Ca | 92.25 | 93.80 | 93.02 |
| TH & Chloride | 92.25 | 93.02 | 93.02 |
| TH & K | 92.25 | 93.02 | 93.02 |
| Chloride & TDS | 91.47 | 94.57 | 92.25 |
| Chloride & EC | 90.70 | 93.80 | 90.70 |
| TH & Carbonate | 90.70 | 93.80 | 90.70 |
| pH & TDS | 89.92 | 94.57 | 91.47 |
| pH & EC | 89.92 | 93.80 | 90.70 |
| Mg & Carbonate | 89.15 | 93.80 | 89.15 |
| Mg & SO4 | 89.15 | 93.80 | 89.92 |
| TDS & SO4 | 89.15 | 89.92 | 89.92 |
| TH & SO4 | 89.15 | 92.25 | 88.37 |
| Carbonate & TDS | 88.37 | 92.25 | 89.15 |
| Mg & TDS | 87.60 | 90.70 | 89.15 |
| Carbonate & EC | 85.27 | 90.70 | 86.05 |
| TH & TDS | 85.27 | 89.15 | 85.27 |
| Mg & EC | 84.50 | 84.50 | 83.72 |
| TH & EC | 83.72 | 86.05 | 83.72 |

**TABLE 8.** Result of model comparison using all features on irrigation water dataset.

| | Model | Accuracy (%) | True Positive (%) | False Positive (%) | False Negative (%) | True Negative (%) |
|---|---|---|---|---|---|---|
| 1 | RF | 94.44 | 91.67 | 8.33 | 2.78 | 97.22 |
| 2 | LR | 91.67 | 94.44 | 5.56 | 11.11 | 88.89 |
| 3 | SVC | 93.06 | 94.44 | 5.50 | 8.33 | 91.67 |

RFE+LR and RFE+SVC but not with RFE+RF, yet the reverse is the case with Na. These contrasting influences are most likely responsible for the lower false positive values observed with LR and SVC compared to RF, and the lower false negative values of RF compared to LR and SVC on Table 8. Table 9 summarizes the results of the top 20 combinations of parameters influencing LR, RF and SVC when used on irrigation water.

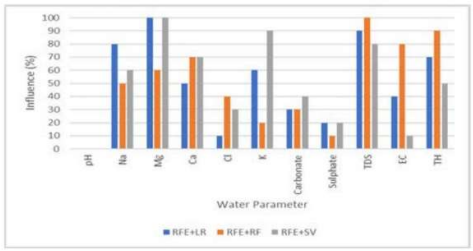



**FIGURE 13.** Parameters influencing the classification accuracy of drinking water.

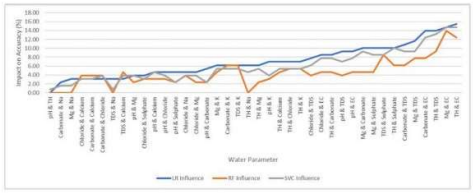


**FIGURE 14.** Impact of various parameters on classification accuracies of drinking water.

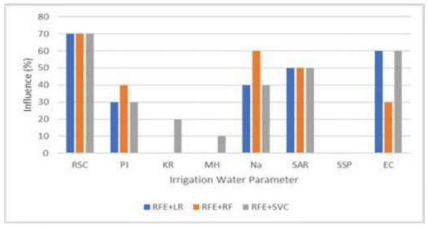



**FIGURE 15.** Parameter influencing the classification accuracy of irrigation water.

**TABLE 9.** Parameter influence on classification accuracies of ML models for irrigation water.

| Excluded Parameters | LR Accuracies (%) | RF Accuracies (%) | SVC Accuracies (%) |
|---|---|---|---|
| SSP & SAR | 93.06 | 93.06 | 95.83 |
| SSP & PI | 90.28 | 91.67 | 91.67 |
| SSP & KR | 90.28 | 93.06 | 93.06 |
| SSP & MH | 90.28 | 93.06 | 93.06 |
| SSP & Na | 88.89 | 93.06 | 93.06 |
| KR & RSC | 83.33 | 90.28 | 79.17 |
| KR & PI | 83.33 | 90.28 | 79.17 |
| RSC & KR | 81.94 | 90.28 | 86.11 |
| RSC & MH | 81.94 | 90.28 | 84.72 |
| Na & RSC | 81.94 | 90.28 | 86.11 |
| Na & PI | 81.94 | 88.89 | 80.56 |
| Na & KR | 81.94 | 88.89 | 80.56 |
| Na & MH | 81.94 | 87.50 | 81.94 |
| RSC & Na | 79.17 | 91.67 | 79.17 |
| KR & Na | 76.39 | 87.50 | 76.39 |
| Na & SSP | 76.39 | 95.83 | 93.06 |
| RSC & SAR | 73.61 | 87.50 | 66.67 |
| RSC & SSP | 68.06 | 90.28 | 84.72 |
| KR & SAR | 58.33 | 77.78 | 68.06 |
| KR & SSP | 58.33 | 91.67 | 90.28 |

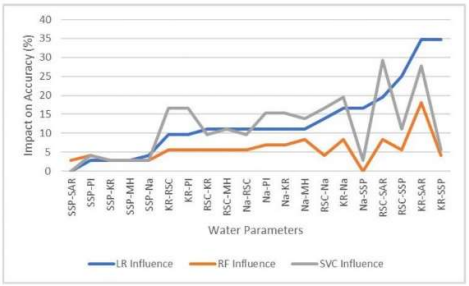



**FIGURE 16.** Impact of various parameters on classification accuracies of irrigation water.

The results in Figure 15 are further confirmed by those on Table 9 and Figure 16. Thus, it can be inferred that SSP has the least effect on the classification accuracy of the three ML models considered, while KR and RSC are the most influential parameters in irrigation water.

## VII. ECONOMIC VIABILITY

In this section, we discuss some basic financial considerations to highlight the advantage of our proposed LoRa-based WaterNet over pre-existing solutions such as cellular networks.

### A. INFRASTRUCTURE COST

Table 10 is a high-level hypothetical bill of materials (BOM), showing the main components required for WaterNet and their approximate costs in US Dollars (USD). The cost reported are based on prices obtained from various online retailers and were correct as at the time of writing. Though certain components such as cables, power adapters, connection jacks, software were not included, the BOM reveals that the solution is achievable with an estimated budget of about US$ 100,000. In essence, with this budget, a water monitoring network covering 11 widely dispersed (and sometimes remote) locations can be deployed in a matter of days. In comparison, setting up a single standard base transceiver station (cellular tower) in a remote location without cellular coverage, costs between US$ 100,000 – US$ 150,000. This cost is exclusive of foundation and concrete works, fencing and brick works, the air-conditioned control room, electrification and wiring, antennas, and backup power generator(s), all of which could raise the cost of the tower to about US$ 250,000. Beyond the cost, erecting cellular towers require extensive site surveys and environmental impact assessment prior to approvals from regulatory authorities, both of which can take several months to complete.

**TABLE 10.** High-level bill of material for waternet.

| Item | Description | Qty | Approx. Unit Price (USD) | Total Price (USD) | Comment |
|---|---|---|---|---|---|
| Sensors | Composite sensor module (such as Libelium Smart Water Ions [63]) | 11 | 7000 | 77000 | A cheaper option could be to purchase the individual sensors and connected them to the Edge nodes |
| Edge Nodes | Single board computer (such as Raspberry Pi) | 12 | 40 | 480 | |
| | Micro-controller with Analog to Digital Converter (such as Arduino Mega) | 12 | 30 | 360 | Interface between analog sensors and the Edge module |
| LoRa Modules | 800/900 MHz LoRa module (such as Semtech SX1272) | 14 | 50 | 700 | Interconnects the FNs and ILLIFU |
| | 2.4 GHz LoRa module (such as Semtech SX1280) | 18 | 75 | 1350 | Interconnects the GWs and FNs |
| LoRa Antennas | Outdoor antennas for the LoRa modules | 32 | 100 | 3200 | Boosts LoRa's range |
| Fog Nodes | High end computer e.g., Core i7 Gen 11, 16GB RAM, 1TB HDD, RTX3070 | 7 | 1500 | 10500 | |
| TOTAL | | | | 93,590 | |

To put this in context, setting up WaterNet to monitor water parameters using cellular networks would cost at least double the cost of using LoRa and would take significantly longer time. This is based on the assumption that only one cellular tower needs to be erected. In situations where all the locations to be monitored are in remote locations with no cellular coverage, the time and cost would grow astronomically. An argument can be made for situations where cellular coverage already exists. In such scenarios, WaterNet could piggyback on the existing infrastructure, thus, the cost

**TABLE 11. SWOT analysis of waternet.**

| Strengths | Weaknesses |
|---|---|
| • LoRa is significantly cheaper compared to other technologies.<br>• Operates on open / license-free radio frequencies.<br>• Coverage in remote areas.<br>• Easy to deploy.<br>• No recurrent subscription bills.<br>• Secure and dedicated network<br>• Can run autonomously with little human intervention | • Susceptible to obstructions from mountains, buildings, trees etc.<br>• 2.4GHz might be susceptible to interference from Wi-Fi, Microwave ovens and other 2.4GHz RF waves emitters.<br>• Requires erecting long antennas. |
| **Opportunities** | **Threats** |
| • Modular and highly scalable.<br>• Other functionalities such as usage prediction and microbial monitoring can be incorporated.<br>• Can be expanded to other locations and provinces.<br>• Output of data analysis at the Fog and Cloud nodes can give useful insights to help stakeholders make development plans. | • Competition from cellular carriers e.g., 3G and 4G LTE.<br>• Permits might be required to mount antennas.<br>• Can be hindered by Government policies. |

of the LoRa modules and antennas (US$ 5,250) would be excluded from the BOM. A close examination of Table 10 shows that the LoRa modules and accessories only account for about 5% of the total cost, hence their exclusion leaves about 95% of the original total cost (US$ 88,340). Moreover, by using cellular networks, other cost elements would be introduced, including cost of cellular gateways, SIM cards, recurrent data subscription fees, etc.; all of which would raise the price above the US$ 100,000 estimated budget. These show that our proposed LoRa-based WaterNet solution is a more economically viable option.

### B. SWOT ANALYSIS

The SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis is often used to identify potential opportunities, advantages, potential weaknesses, and threats to proposed solutions. Table 11 summaries the SWOT analysis for the water monitoring network.

## VIII. CONCLUSION

This work focused on two major concept, firstly, the proposal of a real-time water monitoring network for gathering data on water parameters from water bodies. Secondly, the application of machine learning (ML) models as means of assessing water quality. The developed water monitoring network is based on LoRa, a low power long range protocol for data transmission, and was developed using the City of Cape Town as case study. Results of the simulation done in Radio Mobile, revealed a partial mesh network topology as the most adequate network to cover the city. Data gathered from this monitoring network would ideally be aggregated on a Cloud server, where ML models can then be applied to assess the water's fitness of use for drinking or irrigation purposes. Due to the absence of relevant datasets, two suitable datasets were built in this work and used to training and testing three ML models considered, which are Random Forest (RF), Logistic Regression (LR) and Support Vector Machine (SVM). Results of the test showed that LR performed best for drinking water, as it gave the highest classification accuracy and lowest false positive and negative values, while SVM was better suited for irrigation water. Finally, a model for identifying the most influential water parameter(s) w.r.t classification accuracies of the ML models was then explored using recursive feature elimination (RFE). Obtained results showed that pH, and total hardness were the least influential parameters in drinking water, while SSP was the least for irrigation water.

Though the authors acknowledge the possible application of deep learning models, these were not used in this work. In future works, deep learning models such as the various variants of neural networks could be considered as expansion to this work. Furthermore, water quality indices were manually calculated and used to assess the "fitness for use" of water, future works could explore the application of unsupervised ML models as alternatives to manually calculated water quality indices. In the same vein, rather than using RFE, other approaches such as multi criteria decision making could also be considered to identify influential parameters. Finally, incorporating usage prediction models and microbial monitoring into the water network as well as tracking sources of water contaminates could also be avenues to further this work.

## REFERENCES

[1] B. X. Lee, F. Kjaerulf, S. Turner, L. Cohen, P. D. Donnelly, R. Muggah, R. Davis, A. Realini, B. Kieselbach, L. S. MacGregor, I. Waller, R. Gordon, M. Moloney-Kitts, G. Lee, and J. Gilligan, "Transforming our world: Implementing the 2030 agenda through sustainable development goal indicators," *J. Public Health Policy*, vol. 37, no. S1, pp. 13–31, Sep. 2016.

[2] *Integrated Approaches for Sustainable Development Goals Planning: The Case of Goal 6 on Water and Sanitation*, U. ESCAP, Bangkok, Thailand,2017.

[3] WHO. Water. *Protection of the Human Environment*. Accessed: Jan. 24, 2022. [Online]. Available: www.afro.who.int/health-topics/water

[4] L. Ho, A. Alonso, M. A. E. Forio, M. Vanclooster, and P. L. M. Goethals, "Water research in support of the sustainable development goal 6: A case study in Belgium," *J. Cleaner Prod.*, vol. 277, Dec. 2020, Art. no. 124082.

[5] *Global Nutrition Report 2016: From Promise to Impact: Ending Malnutrition by 2030*, International Food Policy Research Institute, Washington, DC, USA, 2016, doi: 10.2499/9780896295841.

[6] N. Akhtar, M. I. S. Ishak, M. I. Ahmad, K. Umar, M. S. Md Yusuff, M. T. Anees, A. Qadir, and Y. K. A. Almanasir, "Modification of the water quality index (WQI) process for simple calculation using the multi-criteria decision-making (MCDM) method: A review," *Water*, vol. 13, no. 7, p. 905, Mar. 2021.

[7] World Health Organization. (1993). *Guidelines for Drinking-Water Quality*. World Health Organization. Accessed: Jan. 12, 2022. [Online]. Available: http://apps.who.int/iris/bitstream/handle/10665/44584/9789241548151-eng.pdf

[8] *Standard Methods for the Examination of Water and Wastewater*, Federation WE, APH Association, American Public Health Association (APHA), Washington, DC, USA, 2005.

[9] L. S. Clesceri, A. E. Greenberg, and A. D. Eaton, "Standard methods for the examination of water and wastewater," Amer. Public Health Assoc. (APHA), Washington, DC, USA, Tech. Rep.21, 2005.

[10] M. F. Howladar, M. A. Al Numanbakth, and M. O. Faruque, "An application of water quality index (WQI) and multivariate statistics to evaluate the water quality around Maddhapara granite mining industrial area, Dinajpur, Bangladesh," *Environ. Syst. Res.*, vol. 6, no. 1, pp. 1–8, Jan. 2018.

[11] A. R. Finotti, R. Finkler, N. Susin, and V. E. Schneider, "Use of water quality index as a tool for urban water resources management," *Int. J. Sustain. Develop. Planning*, vol. 10, no. 6, pp. 781–794, Dec. 2015.
[12] A. R. Finotti, N. Susin, R. Finkler, M. D. Silva, and V. E. Schneider, "Development of a monitoring network of water resources in urban areas as a support for municipal environmental management," *WIT Trans. Ecol. Environ.*, vol. 182, pp. 133–143, May 2014.
[13] M. Chilundo, P. Kelderman, and J. H. O'keeffe, "Design of a water quality monitoring network for the limpopo river basin in Mozambique," *Phys. Chem. Earth, A/B/C*, vol. 33, nos. 8–13, pp. 655–665, Jan. 2008.
[14] M. Karamouz, M. Karimi, and R. Kerachian, "Design of water quality monitoring network for river systems," in *Critical Transitions in Water and Environmental Resources Management*. London, U.K.: IWA, 2004, pp. 1–9.
[15] J. Foschi, A. Turolla, and M. Antonelli, "Soft sensor predictor of E. Coli concentration based on conventional monitoring parameters for wastewater disinfection control," *Water Res.*, vol. 191, Mar. 2021, Art. no. 116806.
[16] Libelium.com. *IoT Solution for Water Management*. Accessed: Jan. 27, 2022. [Online]. Available: https://www.libelium.com/iot-solutions/smart-water/
[17] K. Ma, A. Bagula, C. Nyirenda, and O. Ajayi, "An IoT-based fog computing model," *Sensors*, vol. 19, no. 12, p. 2783, Jun. 2019.
[18] I. Odun-Ayo, O. Ajayi, and A. Falade, "Cloud computing and quality of service: Issues and developments," in *Proc. Int. Multi-Conf. Eng. Comput. Scientists (IMECS 2018)*, Hong kong, Mar. 2018, pp. 14–16.
[19] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 855–873, 2nd Quart., 2017.
[20] F. M. Ortiz, T. T. de Almeida, A. E. Ferreira, and L. H. M. K. Costa, "Experimental vs. simulation analysis of LoRa for vehicular communications," *Comput. Commun.*, vol. 160, pp. 299–310, Jul. 2020.
[21] H. A. Aden and K. R. Karlsson, "Evaluating LoRa physical as a radio link technology for use in a remote-controlled electric switch system for a network bridge radio-node," M.S. thesis, Dept. Elect. Eng., School Elect. Eng. Comput. Sci., KTH Royal Inst. Technol., Stockholm, Sweden, 2018.
[22] M. Zennaro, A. Bagula, D. Gascon, and A. B. Noveleta, "Long distance wireless sensor networks: Simulation vs reality," in *Proc. 4th ACM Workshop Netw. Syst. Developing Regions (NSDR)*, 2010, pp. 1–2.
[23] M. G. Uddin, S. Nash, and A. I. Olbert, "A review of water quality index models and their use for assessing surface water quality," *Ecolog. Indicators*, vol. 122, Mar. 2021, Art. no. 107218.
[24] T. Banda and M. Kumarasamy, "Development of a universal water quality index (UWQI) for South African river catchments," *Water*, vol. 12, no. 6, p. 1534, May 2020.
[25] P. Shrivastava and R. Kumar, "Soil salinity: A serious environmental issue and plant growth promoting bacteria as one of the tools for its alleviation," *Saudi J. Biol. Sci.*, vol. 22, no. 2, pp. 123–131, Mar. 2015.
[26] R. K. Yadav, A. Jha, and A. Choudhary, "IoT based prediction of water quality index for farm irrigation," in *Proc. Int. Conf. Artif. Intell. Smart Syst. (ICAIS)*, Mar. 2021, pp. 1443–1448.
[27] A. El Bilali, A. Taleb, and Y. Brouziyne, "Groundwater quality forecasting using machine learning algorithms for irrigation purposes," *Agricult. Water Manage.*, vol. 245, Feb. 2021, Art. no. 106625.
[28] R. M. Khalaf and W. H. Hassan, "Evaluation of irrigation water quality index IWQI for Al-Dammam confined aquifer in the west and southwest of Karbala city, Iraq," *Int. J. Civil Eng.*, vol. 23, pp. 21–34, Jun. 2013.
[29] A. C. M. Meireles, E. M. D. Andrade, L. C. G. Chaves, H. Frischkorn, and L. A. Crisostomo, "A new proposal of the classification of irrigation water," *Revista Ciência Agronômica*, vol. 41, no. 3, pp. 349–357, Sep. 2010.
[30] S. Yıldız and C. B. Karakuş, "Estimation of irrigation water quality index with development of an optimum model: A case study," *Environ., Develop. Sustainability*, vol. 22, no. 5, pp. 4771–4786, Jun. 2020.
[31] M. Y. Hasan, M. F. H. Khan, M. K. Islam, M. M. Hasan, M. A. Hossain, M. U. Monir, M. A. Samad, and M. T. Ahmed, "Dataset on the evaluation of hydrochemical properties and groundwater suitability for irrigation purposes: South-western part of Jashore, Bangladesh," *Data Brief*, vol. 32, Oct. 2020, Art. no. 106315.
[32] A. Bagula, O. Ajayi, and H. Maluleke, "Cyber physical systems dependability using CPS-IoT monitoring," *Sensors*, vol. 21, no. 8, p. 2761, Apr. 2021.
[33] B. Mf, H. Bagula, M. Mandava, L. Ck, and A. Bagula, "Cyber-healthcare kiosks for healthcare support in developing countries," in *Proc. Int. Conf. e-Infrastruct. e-Services Developing Countries*, Cham, Switzerland: Springer, Nov. 2018.
[34] K. Kyamakya, J. C. Chedjou, F. Al-Machot, A. H. Mosa, and A. Bagula, "Intelligent transportation related complex systems and sensors," *Sensors*, vol. 21, no. 6, p. 2235, Mar. 2021.
[35] D. Djenouri, E. Karbab, S. Boulkaboul, and A. Bagula, "Networked wireless sensors, active RFID, and handheld devices for modern car park management: WSN, RFID, and mob Devs for car park management," in *Securing the Internet of Things: Concepts, Methodologies, Tools, and Applications*. Hershey, PA, USA: IGI Global, 2020, pp. 1012–1024.
[36] N. Mbayo, A. Bagula, O. Ajayi, and H. Maluleke, "Environment 4.0: An IoT-based pollution monitoring model," in *Proc. Int. Conf. e-Infrastruct. e-Services Developing Countries*, Dec. 2021.
[37] T. Janssen, N. BniLam, M. Aernouts, R. Berkvens, and M. Weyn, "LoRa 2.4 GHz communication link and range," *Sensors*, vol. 20, no. 16, p. 4366, Jan. 2020.
[38] City of Cape Town. (2018). *Water Services and the Cape Town Urban Water Cycle*. Accessed: Jan. 28, 2022. [Online]. Available: https://resource.capetown.gov.za/documentcentre/Documents/Graphics
[39] Ilifu. *Cloud Computing for Data-Intensive Research*. Accessed: Feb. 7, 2022. [Online]. Available: https://www.ilifu.ac.za/
[40] *ScienceDirect Data Brief*. Accessed: Jan. 12, 2022. [Online]. Available: https://www.sciencedirect.com/journal/data-in-brief/about/aims-and-scope
[41] R. Divahar, P. S. A. Raj, S. P. Sangeetha, T. Mohanakavitha, and T. Meenambal, "Dataset on the assessment of water quality of ground water in kalingarayan canal, erode district, Tamil Nadu, India," *Data Brief*, vol. 32, Oct. 2020, Art. no. 106112.
[42] S. Acharya, S. K. Sharma, and V. Khandegar, "Assessment of groundwater quality by water quality indices for irrigation and drinking in south west Delhi, India," *Data Brief*, vol. 18, pp. 2019–2028, Jun. 2018.
[43] V. Masindi, "Dataset on physicochemical and microbial properties of raw water in four drinking water treatment plants based in south Africa," *Data Brief*, vol. 31, Aug. 2020, Art. no. 105822.
[44] SABS Standards Division. (2015). *South African National Standard (SANS) Drinking Water—Part 2: Application of SANS 241-1*. Accessed: Jan. 12, 2022. [Online]. Available: https://alabbott.co.za/wp-content/uploads/2020/02/abbott_sans_241_test_requirements.pdf
[45] P. Balamurugan, P. S. Kumar, and K. Shankar, "Dataset on the suitability of groundwater for drinking and irrigation purposes in the Sarabanga river region, Tamil Nadu, India," *Data Brief*, vol. 29, Apr. 2020, Art. no. 105255.
[46] A. Abbasnia, M. Radfard, A. H. Mahvi, R. Nabizadeh, M. Yousefi, H. Soleimani, and M. Alimohammadi, "Groundwater quality assessment for irrigation purposes based on irrigation water quality index and its zoning with GIS in the villages of chabahar, sistan and baluchistan, Iran," *Data Brief*, vol. 19, pp. 623–631, Aug. 2018.
[47] A. Verma, B. K. Yadav, and N. B. Singh, "Data on the assessment of groundwater quality in Gomti–Ganga alluvial plain of northern India," *Data Brief*, vol. 30, Jun. 2020, Art. no. 105660.
[48] H. Soleimani, A. Abbasnia, M. Yousefi, A. A. Mohammadi, and F. C. Khorasgani, "Data on assessment of groundwater quality for drinking and irrigation in rural area Sarpol-e Zahab city, Kermanshah province, Iran," *Data Brief*, vol. 17, pp. 148–156, Apr. 2018.
[49] K. Jafari, F. B. Asghari, E. Hoseinzadeh, Z. Heidari, M. Radfard, H. N. Saleh, and H. Faraji, "Groundwater quality assessment for drinking and agriculture purposes in Abhar city, Iran," *Data Brief*, vol. 19, pp. 1033–1039, Aug. 2018.
[50] M. P. P. Sithole, N. I. Nwulu, and E. M. Dogo, "Dataset for a wireless sensor network based drinking-water quality monitoring and notification system," *Data Brief*, vol. 27, Dec. 2019, Art. no. 104813.
[51] R. K. Horton, "An index number system for rating water quality," *J. Water Pollut. Control Fed.*, vol. 37, no. 3, pp. 300–306, Mar. 1965.
[52] E. V. A. Sylvester, P. Bentzen, I. R. Bradbury, M. Clément, J. Pearce, J. Horne, and R. G. Beiko, "Applications of random forest feature selection for fine-scale genetic population assignment," *Evol. Appl.*, vol. 11, no. 2, pp. 153–165, Feb. 2018.
[53] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, vol. 1, Aug. 1995, pp. 278–282.
[54] L. Wang, *Support Vector Machines: Theory and Applications*. Cham, Switzerland: Springer, Jun. 2005.
[55] S. May, O. Isafiade, and O. Ajayi, "An enhanced Naïve Bayes model for crime prediction using recursive feature elimination," in *Proc. 4th Int. Conf. Artif. Intell. Pattern Recognit. (AIPR)*, Sep. 2021, pp. 580–586.
[56] X.-W. Chen and J. C. Jeong, "Enhanced recursive feature elimination," in *Proc. 6th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2007, pp. 429–435.

[57] R. Coudé. *Radio Mobile*. Accessed: Jan. 29, 2022. [Online]. Available: www.ve2dbe.com
[58] Google. *Google Maps*. Accessed: Feb. 1, 2022. [Online]. Available: https://maps.google.com
[59] *Topographic Map. Free Opographic Maps Visualization and Sharing*. Accessed: Feb. 1, 2022. [Online]. Available: https://en-us.topographic-map.com/maps/6h3s/Cape-Town/
[60] Semtech. (2019). *Semtech SX128x Long Range Datasheet*. Accessed: Feb. 16, 2022. [Online]. Available: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R000000HVET/HfcgiChyabtiPTh6EjcDM6ZEwAOQV7IirEmRULgggMM
[61] *IEEE Standards Association. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, Standard 2006;802:4, 2006.
[62] S. Farahani, "Transceiver requirements," in *ZigBee Wireless Networks and Transceivers*. London, U.K.: Newnes, 2008, pp. 137–170, doi: 10.1016/B978-0-7506-8393-7.00004-2.
[63] Libelium. *Libelium Smart Water Sensor Platform Adds Ion Monitoring*. Accessed: Feb. 16, 2022. [Online]. Available: https://www.libelium.com/libeliumworld/smart-water-ions-sensors-calcium-fluoride-chloride-nitrate-iodide-lead-bromide-cupric-silver/

**OLASUPO O. AJAYI** received the Ph.D. degree in computer science from the University of Lagos, in 2017. He is currently a Senior Lecturer with the Department of Computer Science, University of the Western Cape, Cape Town, South Africa (UWC). He has published a number of articles around these research interests in journals, books, and proceedings of international conferences. His research interests include cloud/fog/edge computing, the Internet of Things, cyber-physical systems, and the enabling technologies of the 4IR.

**ANTOINE B. BAGULA** received the dual M.Sc. degree in computer engineering from the Universite catholique de Louvain (UCL), Belgium, and in computer science from the University of Stellenbosch (SUN), South Africa, and the Ph.D. degree in communication systems from the Royal Institute of Technology (KTH), Sweden. He is currently a Full Professor and the Head of the Department of Computer Science, UWC, where he also leads the Intelligent Systems and Advanced Telecommunication (ISAT) Laboratory. He is a well published scientist. His research interests include the Internet of Things, cloud/fog computing, and next generation networks including 4G/5G. For the last ten years, he has applied a combination of these research interests to the design and implementation of cyber-physical systems frameworks for health, energy, water, and pollution.

**HLONIPHANI C. MALULEKE** received the M.Sc. degree in computer science from UWC, in 2021, with a thesis focused on providing 5G communication in rural and under-served communities of South Africa, where he is currently pursuing the Ph.D. degree with research focused on the applications of cyber-physical systems in pollution monitoring, under the supervision of Prof. B. Bagula. His research interests include 5G communications, machine learning, CPS, and 4IR for rural and under-served areas of South Africa.

**ZAHEED GAFFOOR** is currently pursuing the Ph.D. degree with the Department of Earth Science, UWC. He is also a Research Scientist with IBM Research, where he works on climate and weather related research. His research interests include the application of data driven tools and techniques in the hydrological domain.

**NEBO JOVANOVIC** received the Ph.D. degree from the University of Pretoria. He is currently an Associate Professor in water resources management at UWC. He has over 20 years of experience in the water sector having conducted, led and published research in crop and hydrological modeling, dryland salinity, food security and drought impacts in small-holder irrigated farming systems, and applications of remote sensing in water management. His current and future research interests include the determination of water use and consumption to improve water allocations in semi-arid areas, in particularly to agricultural water users, and adaptation strategies and interventions for augmentation of water supply and water management in drought-prone areas.

**KEVIN C. PIETERSEN** received the Ph.D. degree in hydrogeology from UWC. He is currently a Research Fellow with the Department of Environmental and Water Studies, UWC. He has 30 years of experience in the water, environment, geosciences, and energy sectors. He has extensive experience in the exploration, development, and management of groundwater. He was the Team Leader for the recently completed Consultancy Services for Water Resources Management Research in the Eastern Kalahari Karoo Basin Transboundary Aquifer, a Groundwater Institutional Advisor for a World Bank Project entitled Groundwater Management in the Horn of Africa, and the Co-Chair of the International Association of Hydrogeologist (IAH) Transboundary Aquifer Commission. He also developed the Strategic Action Plan for the Stampriet Transboundary Aquifer System on behalf of UNESCO and currently provides technical support to the Great Limpopo Transfrontier Conservation Area as a Groundwater Expert to better understand water governance and water use in the Pafuri-Sengwe Node to inform drought preparation and mitigation measures at the community level. He is a fellow of the Geological Society and a Senior Fellow of the Water Institute of Southern Africa.

## APPENDIX – B

## PLAGARISM REPORT