WATER POLLUTION FORECASTING AND ALERT SYSTEM USING XGBOOST CLASSIFIER.

A PROJECT REPORT

Submitted by

THELMA PRINCY M [211420104289]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"WATER POLLUTION FORECASTING AND ALERT SYSTEM USING XGBOOST CLASSIFIER."** is the bonafide work of **"THELMA PRINCY M [211420104289]"** who carried out the project work under my supervision.

SIGNATURE

Dr.L. JABASHEELA, M.E., Ph.D .,
PROFESSOR
HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NAZARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

Mrs. A. KANCHANA, M.E., (Ph.D)
SUPERVISOR
ASSISTANT PROFESSOR

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NAZARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidate was examined in the End Semester Project Viva-Voce Examination held on.....

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION BY THE STUDENT

I Thelma Princy M [211420104289] hereby declare that this project report titled “Water Pollution Forecasting and Alert System using XGBoost Classifier”, under the guidance of Mrs. A. Kanchana, M.E., (Ph.D) is the original work done by me and I have not plagiarized or submitted to any other degree in any university.

THELMA PRINCY M

ACKNOWLEDGEMENT

My profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**., for his fervent encouragement. His inspirational support proved instrumental in galvanizing my efforts, ultimately contributing significantly to the successful completion of this project

I want to express my deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for graciously affording me the essential resources and facilities for undertaking of this project.

My gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

I express my heartfelt thanks to **Dr. L. JABASHEELA, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

I would like to express my sincere thanks to Project Coordinator **Dr. K. VALARMATHI, M.E., Ph.D.,** and Project Guide **Mrs. A. KANCHANA, M.E., (Ph.D)** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

THELMA PRINCY M [211420104289]

ABSTRACT

Water Pollution Forecasting is crucial for ensuring the safety of water resources and public health. This research project proposes a comprehensive approach that integrates data analytics and machine learning techniques, including KNN, XGBoost, etc., to predict water quality parameters with a high degree of accuracy. The study aims to leverage historical water quality data, meteorological information, and other relevant variables to develop predictive models. The methodology involves preprocessing and analyzing large datasets to identify patterns, correlations, and anomalies. The research specifically evaluates the performance of various machine learning algorithms, and the results indicate that XGBoost exhibits the highest accuracy among the models considered, achieving an impressive accuracy rate of 90%. Therefore, XGBoost will be deployed as the core classifier in the proposed Water Pollution Forecasting and Alert System. The predictive models will focus on key water quality indicators, such as chemical concentrations and environmental microbes. The deployment of XGBoost in a web application will enable real-time water quality monitoring, with an alert system triggered when contamination is detected. This integration of cutting-edge machine learning techniques, including XGBoost, into a practical application underscores the project's commitment to providing accurate and timely information for effective water quality management and pollution control.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

1.	ML	Machine Learning
2.	XGBOOST	Extreme Gradient Boosting
3.	KNN	K-Nearest Neighbors
4.	CNN	Convolutional Neural Networks

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Water quality is a pivotal aspect of public health, environmental sustainability, and resource utilization. Traditional methods of monitoring water quality often lack real-time capabilities, leading to delayed responses to contamination events or quality fluctuations. Water Quality Forecasting plays a pivotal role in safeguarding public health and preserving ecosystems. This project focuses on leveraging the power of data analytics and machine learning to enhance water quality prediction, providing a robust solution for pollution risk assessment. The initial phase involves a meticulous data analytics process, encompassing data cleaning and preprocessing, to ensure the accuracy and reliability of the dataset. Following the data preparation phase, a suite of machine learning models, including XGBoost, KNN etc., will be employed to build, train, and test the forecasting system. These models are chosen for their ability to discern complex patterns within the dataset, offering a comprehensive understanding of water quality dynamics. To make the water quality forecasting accessible to the public, the developed optimum model will be deployed in a user-friendly web application. This application will enable users to input relevant values, receiving accurate pollution risk levels and detailed information about harmful pollutants. In the event of elevated risk levels, the application will automatically alert the respective environmental authorities via email, fostering a proactive approach towards environmental conservation and public safety. This integrated system not only empowers individuals with real-time water quality insights but also establishes a direct link between citizens and environmental governance, contributing to a sustainable and informed community.

1.2 SCOPE OF THE PROJECT

The project on Water Quality Forecasting and Alert System serves as a crucial tool in environmental management and public health. By leveraging advanced analytics and machine learning algorithms, this initiative enables real-time prediction and monitoring of

water quality parameters. This proactive approach aids in the early detection of potential contamination or deterioration in water quality, allowing authorities to implement timely interventions and preventive measures. Furthermore, the project contributes to sustainable water resource management by providing valuable insights into the factors influencing water quality variations. Improved forecasting accuracy facilitates better decision-making for water treatment plants, ensuring the delivery of safe and clean water to communities. Overall, the project's significance lies in its ability to enhance environmental stewardship, protect public health, and support the sustainable utilization of one of our most vital resources.

1.3 PROBLEM DEFINITION

The Water Pollution Forecasting and Alert System addresses the serious challenge of water pollution by utilizing data analytics and machine learning. Traditional monitoring methods often fall short in providing real-time insights, leading to delayed responses to potential pollution incidents. This system aims to revolutionize water quality management by collecting comprehensive data from various sources, integrating advanced analytics to identify patterns, and employing machine learning models for early prediction of pollution events. The key components include the development of a user-friendly interface for stakeholders, the implementation of an alert system for timely notifications, and seamless integration with existing environmental policies. Overall, this innovative approach seeks to proactively safeguard water resources, ecosystems, and public health by providing actionable insights and timely alerts.

CHAPTER 2

LITERATURE REVIEW

1. **Title:** Water Quality Predictions for Urban Streams Using Machine Learning.

   Author: Lokesh Jalagam, Nathaniel Shepherd, Jingyi Qi, Nicole Barclay, Michael Smith

   Year: 2023

   Description: This study investigates the impact of land use and rainfall on water quality in urban streams in Mecklenburg County, North Carolina. It uses land use data from the Multi-Resolution Land Characteristics Consortium and monthly average precipitation data to predict total suspended solids (TSS) pollutant levels. The accuracy of the prediction is measured using statistical methods and compared among models. The study demonstrates the viability of the proposed approach for water quality prediction.

2. **Title:** An Ensemble Model for Water Temperature Prediction in Intensive Aquaculture.

   Author: Mingyan Wang, Qing Xu, Yingying Cao,shahbaz Gul Hassan, Wenjun Liu

   Year: 2023

   Description: A novel hybrid model for accurate water temperature prediction is proposed in intensive aquaculture systems. The model integrates advanced data processing and prediction techniques, including VMD method for data decomposition and CNN algorithm for feature extraction. The bi-directional LSTM and self-concerned combination are used for final prediction results. This study can be applied in fishery farming to reduce farming risks and promote modernization.

3. **Title:** A Novel Hybrid Model to Predict Dissolved Oxygen for Efficient Water

Quality in Intensive Aquaculture.

Author: Wenjun Liu, Shuangyin Liu,Shahbaz Gul Hassan, Yingying Cao, Longqin Xu, Dachun Feng

Year: 2023

Description: This study introduces a hybrid model using the Light Gradient Boosting Machine (LightGBM) and Bidirectional Simple Recurrent Unit (BiSRU) to predict dissolved oxygen content in aquaculture environments. The model uses linear interpolation and smoothing to identify significant parameters, and the attention method maps weighting and learning parameter matrices. The model can accurately anticipate dissolved oxygen trends in just 122 seconds, providing a reference for intensive aquaculture water quality regulation.

4. **Title:** Water Quality Monitoring Using IoT & Machine Learning

   Author: Andrew, Benard, Anthony

   Year: 2022

   Description: The authors propose a system using IoT and Machine Learning to monitor water quality, pilferage, and wastage. Safe water access is a fundamental human right, but contamination, leakages, and pilferage occur due to consumers using water from pipes and springs. The system uses machine learning algorithms for decision-making.

5. **Title:** A review of the application of machine learning in water quality evaluation.

   Author: Mengyuan Zhu, Jiawei Wang, Xiao Yang, Yu Zhang, Linyu Zhang, Hongqiang Ren, Bing Wu, Lin Ye

   Year: 2022

   Description: Machine learning is a crucial tool for data analysis, classification, and prediction in aquatic environments. It can efficiently solve complex nonlinear problems and is applied to water treatment, management systems, pollution

control, water quality improvement, and watershed ecosystem security. This review discusses applications in surface water, groundwater, drinking water, sewage, and seawater.

6. **Title:** Evaluation and Analysis of Goodness of Fit for Water Quality Parameters Using Linear Regression Through the Internet-of-Things-Based Water Quality Monitoring System.

   Author: Harish H. Kenchannavar, Prasad M. Pujar , Raviraj M. Kulkarni, and Umakant P. Kulkarni

   Year: 2022

   Description: The IoT-enabled water quality monitoring (WQM) system is being utilized in India to monitor freshwater resources, focusing on physicochemical parameters like temperature, pH, and dissolved oxygen. This is crucial due to the potential for water pollution in mineral-rich watersheds. The Ghataprabha river's water quality is assessed using linear regression analysis and one-way ANOVA, and the river data set is also used for training.

7. **Title:** Predictive Models for River Water Quality using Machine Learning and Big Data Techniques - A Survey

   Author: Jitha P Nair Vijaya M S

   Year: 2021

   Description: Water quality is being significantly impacted by pollution from industrial waste, human and agricultural runoff, and unwanted nutrients, posing a high risk to health and living organisms. Researchers are using machine learning and big data analytics to evaluate and predict water quality, analyzing prediction models and experimental results.

8. **Title:** A Complete Proposed Framework for Coastal Water Quality Monitoring System With Algae Predictive Model

Author: Nur Aqilah Paskhal Rostam, Nurul Hashimah Ahamed Hassain Malim, Rosni Abdullah, Abdul Latif Ahmad, Boon Seng Ooi, and Derek Juinn Chieh Chan

Year: 2021

Description: This paper presents a framework for predicting harmful algae blooms for water management, utilizing machine learning, deep learning, and deep time series forecasting algorithms. It emphasizes understanding algae growth factors and prediction problems, with the Long Short-term Memory (LSTM) algorithm being the best fit for accurate algal growth prediction.

9. **Title:** Comparison of Water Quality Classification Models using Machine Learning.

Author: Neha Radhakrishnan, Anju S Pillai

Year: 2020

Description: This paper compares water quality classification models using machine learning algorithms like SVM, Decision Tree, and Naïve Bayes, considering pH, DO, BOD, and electrical conductivity. The decision tree algorithm was found to be a better classification model after assessing results.

10. **Title:** Quality Risk Analysis for Sustainable Smart Water Supply Using Data Perception.

Author: Di Wu, Hao Wang, Hadi Mohammed, and Razak Seidu

Year: 2020

Description: This paper proposes a risk analysis framework for sustainable smart water supply systems in urban areas. It uses industrial process data for water quality changes and risk detection. The Adaptive Frequency Analysis (Adp-FA) method, tested on industrial quality data from a Norwegian project, is found to perform better in most aspects, potentially aiding early warnings and decision support.

CHAPTER 3

THEORETICAL BACKGROUND

3.1 DEVELOPMENT ENVIRONMENT

SOFTWARE REQUIREMENT

3.1.1 Python

3.1.2 Jupyter Notebook

3.1.3 HTML & CSS

HARDWARE REQUIREMENT

3.1.4 Processor: Intel

3.1.5 Memory (RAM): 8 Gb

3.1.6 Hard Drive: 32 GB

3.1.7 Internet Connection

3.2 SYSTEM ARCHITECTURE

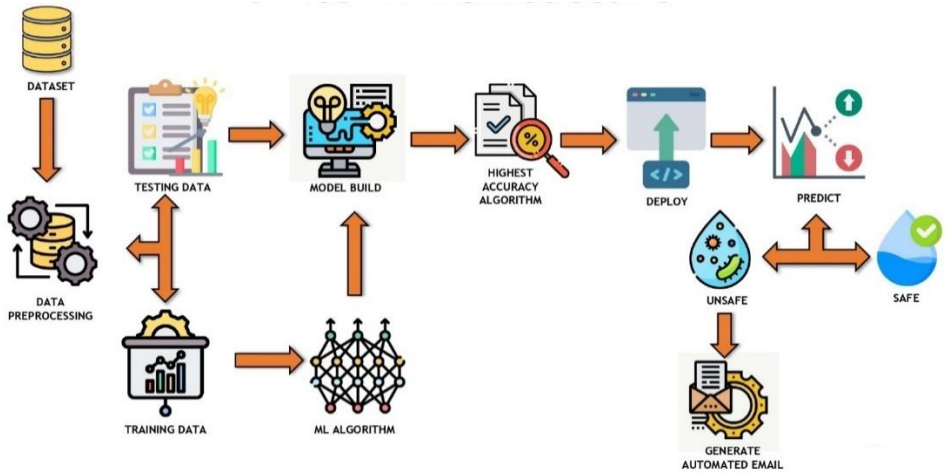


Fig 3.2.1 Architecture diagram

The "Water Pollution Forecasting and Alert System using XGBoost Classifier" uses a structured approach to acquire diverse datasets from various sources, ensuring data reliability through robust preprocessing. The cleaned dataset is then divided into training and testing sets for algorithm development and evaluation. Four powerful machine learning algorithms – Logistic Regression, K-Nearest Neighbors (KNN), Convolutional Neural Networks (CNN), and XGBoost – are compared for their performance metrics. XGBoost is chosen as the most accurate algorithm, showcasing its superiority in predicting water pollution levels. The algorithm is then deployed onto a user-friendly website, where users can input real-time water quality values. If an unsafe prediction is made, an email alert is generated, providing timely and proactive notification to users and stakeholders. This integrated system offers accurate and efficient water quality predictions, empowering users to make informed decisions regarding water usage. By combining machine learning, web deployment, and automated alerts, this Water Pollution Forecasting and Alert System contributes to proactive water resource management, environmental conservation, and public health.

3.3 PROPOSED SYSTEM

The Proposed Solution describes a process for gathering water quality data, cleaning & preprocessing it, extracting features, dividing the dataset into training and testing sets, using machine learning models like XG Boost and KNN, evaluating performance, creating a user-friendly web application, integrating the trained model for real-time prediction, and prompting users to notify authorities via email.

MERITS

- Machine learning models like XG Boost and KNN offer high predictive accuracy in water quality forecasting, ensuring reliable pollution risk assessments and pollutant details for users.
- The web application's intuitive design facilitates user-friendly input of water quality parameters and accurate pollution risk levels, promoting widespread system use and comprehension.
- The notification prompt allows users to notify authorities via email when pollution levels are high, promoting community engagement and environmental stewardship.

3.3.1 DATASET DESCRIPTION

The dataset, sourced from Kaggle, is a comprehensive compilation comprising 3,277 records and encompasses ten distinct attributes reflecting various water quality parameters. These attributes include pH, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic carbon, Trihalomethanes, Turbidity, and Potability. Each of these variables plays a pivotal role in characterizing the composition and safety of water. The pH level is indicative of the water's acidity or alkalinity, while Hardness measures the concentration of minerals. Solids represent the total dissolved solids in the water, Chloramines quantify the presence of disinfectants, and Sulfate gauges

the sulfate content. Conductivity is a measure of the water's ability to conduct an electric current. Organic carbon indicates the presence of carbon-based compounds, Trihalomethanes measure disinfection byproducts, and Turbidity reflects the water's cloudiness. Finally, the attribute Potability is a binary indicator denoting whether the water is suitable for consumption. This dataset, encompassing diverse attributes, forms the basis for the development and evaluation of machine learning models in the proposed Water Pollution Forecasting and Alert System.

3.3.2 INPUT DESIGN (UI)

The user interface (UI) of the system is meticulously crafted to ensure ease of use and accessibility for stakeholders involved in water quality management. It features intuitive input forms and interactive visualizations, allowing users to input relevant data such as location and specific water quality parameters. The UI is designed to be responsive and adaptable across various devices, enabling users to access the system seamlessly from desktop computers, tablets, and laptops.

3.3.3 MODULE DESIGN

3.3.3.1 UML DIAGRAMS

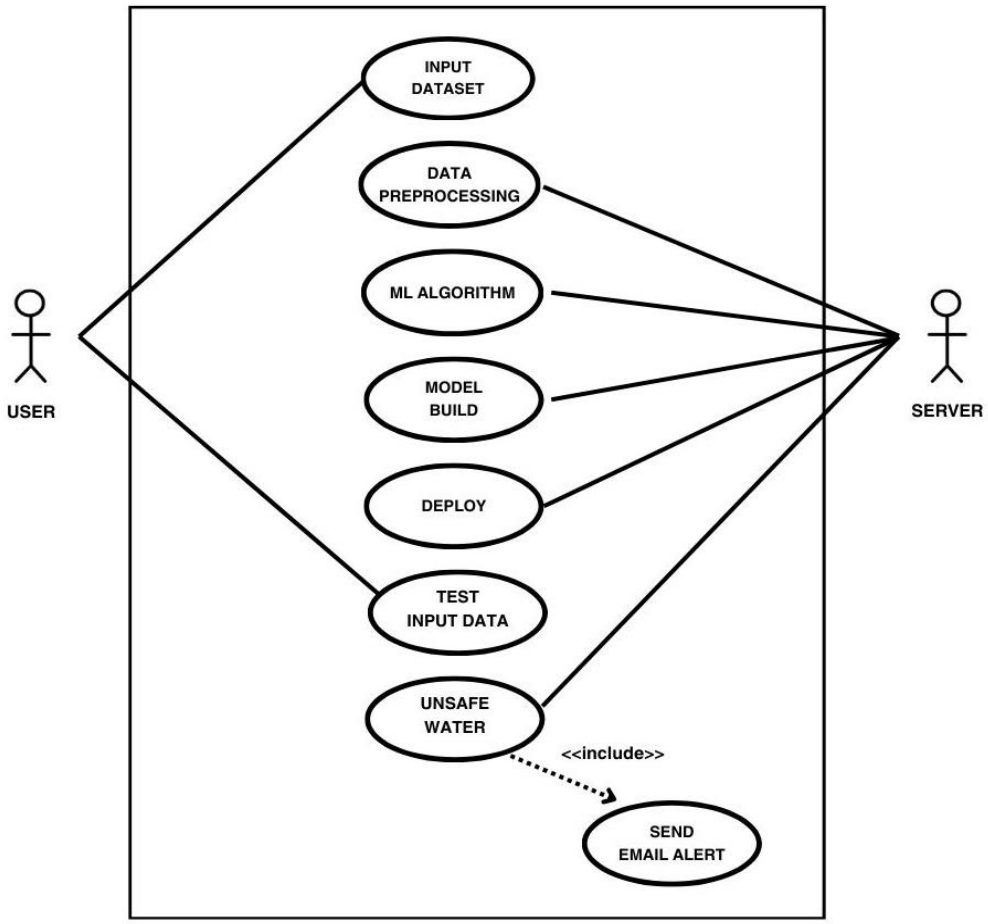


Fig 3.3.3.1 Use case diagram

The Water Pollution Forecasting and Alert System aims to provide accurate predictions of water quality using machine learning techniques. This use case diagram provides a visual representation of the primary functionalities and interactions within the Water Pollution Forecasting and Alert System. Actors include the User and Server, and the use cases cover data retrieval, preprocessing, algorithm building, website deployment, real-time prediction, and alert generation

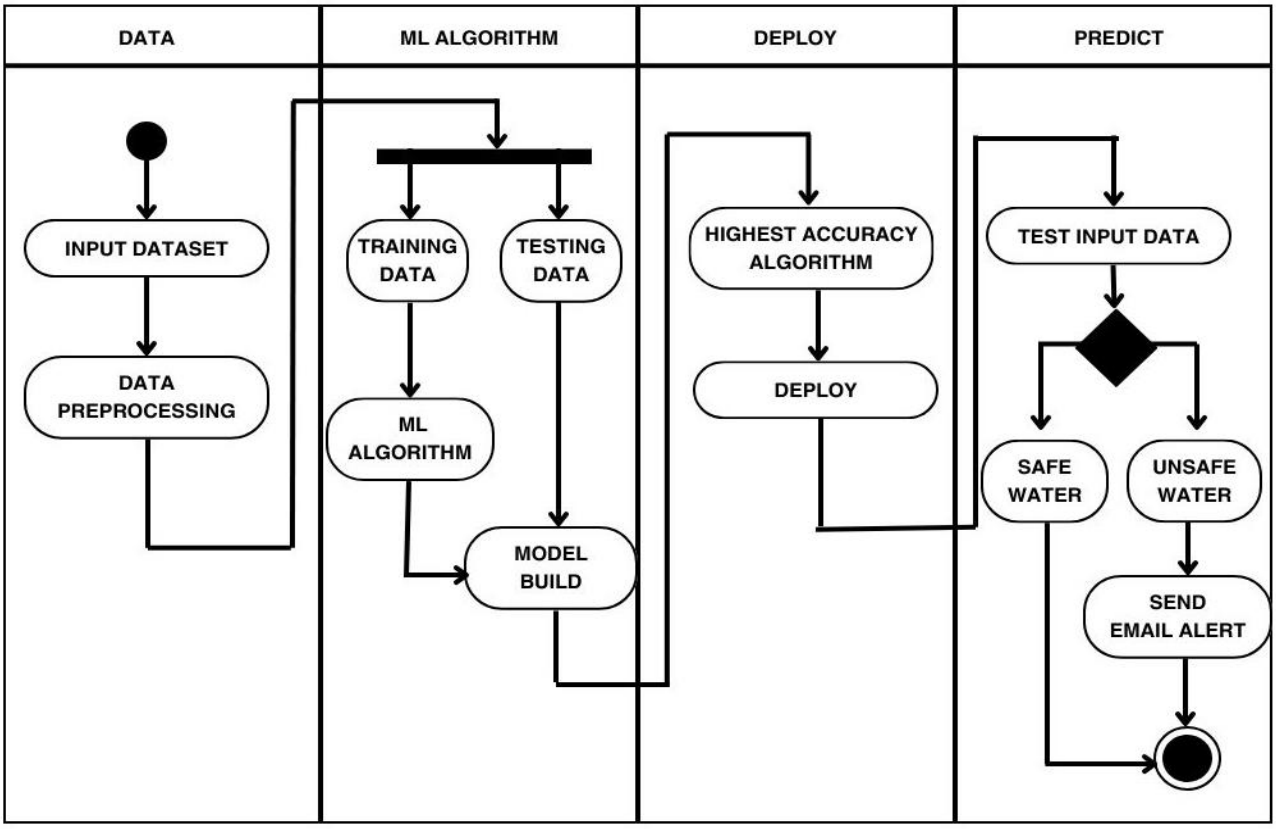


Fig 3.3.3.2 Activity diagram

The activity diagram represents the flow of activities within the Water Pollution Forecasting and Alert System. Initially, the system acquires data from various sources. The dataset undergoes preprocessing to handle missing values and outliers. Subsequently, the dataset is split into training and testing sets for algorithm development. The system identifies the highest accuracy algorithm, which is then deployed on the website. Users can input real-time water quality values through the website, triggering a prediction. If the water is predicted as unsafe, an automatic email alert is generated and sent to notify relevant parties.

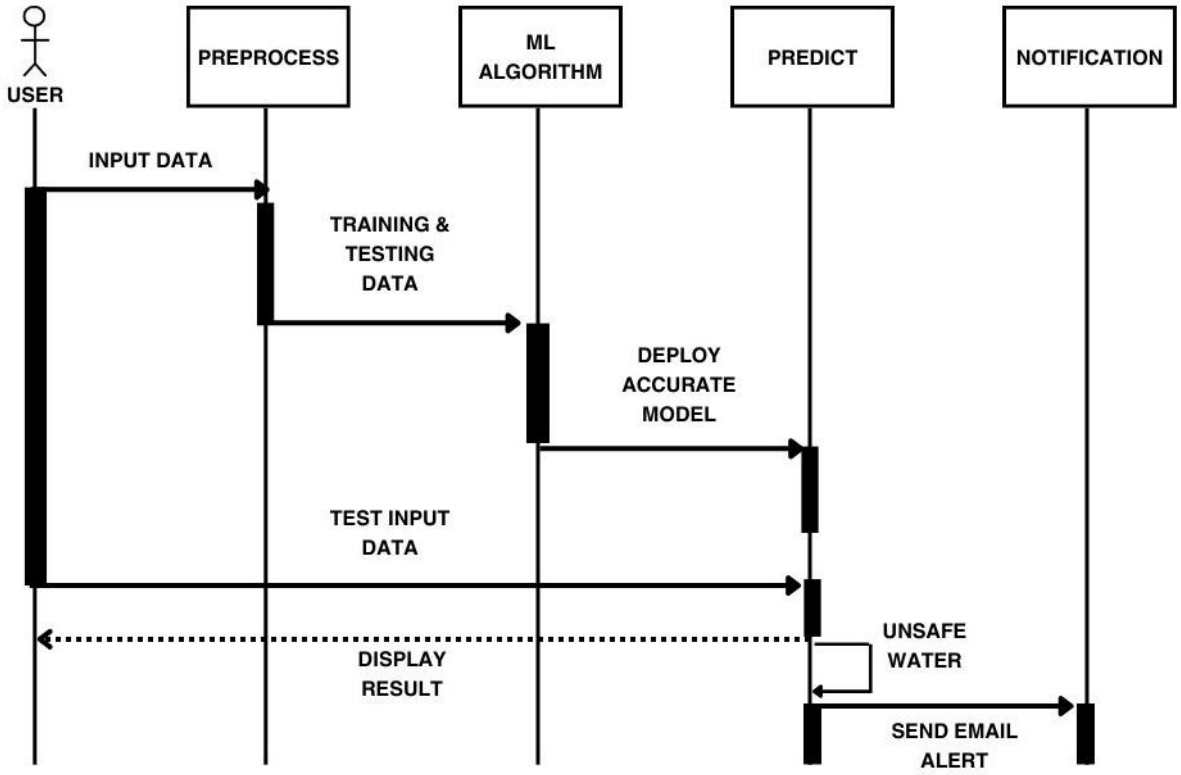


Fig 3.3.3.3 Sequence diagram

The sequence diagram for the "Water Pollution Forecasting and Alert System using XGBoost Classifier" uses a user interface to request water quality prediction. Real-time parameters are inputted, processed through an algorithm, and the highest accuracy algorithm is deployed. If unsafe water is detected, an email is sent to the user, highlighting user interaction and system response.

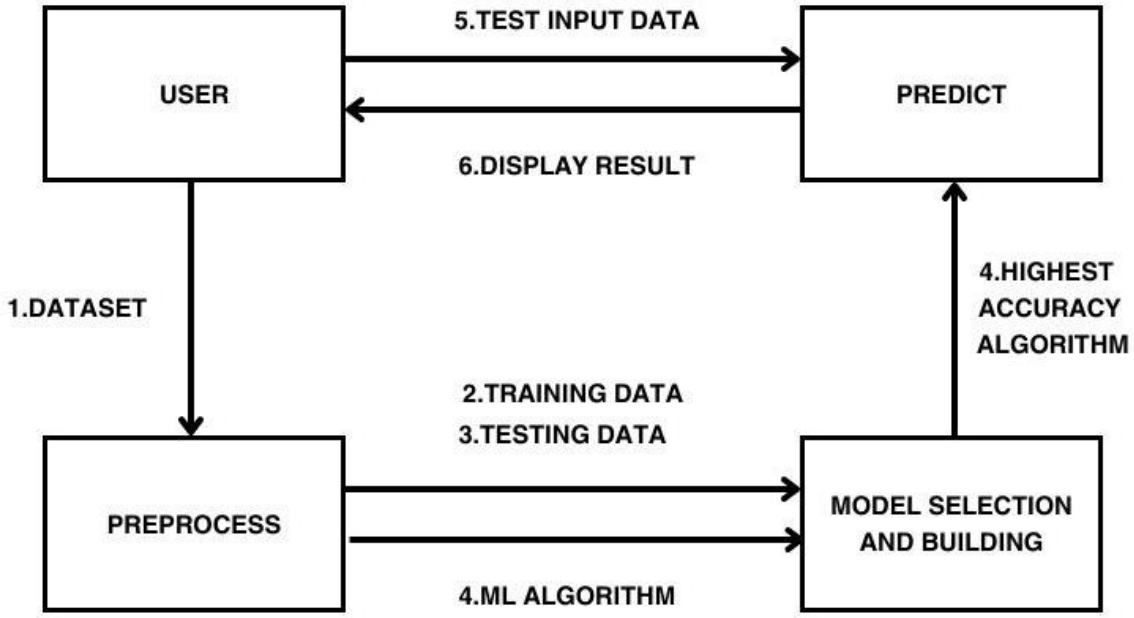


Fig 3.3.3.4 Collaboration diagram

The Water Pollution Forecasting and Alert System uses Machine Learning to provide accurate water quality predictions and timely user notifications. It uses diverse data sources, preprocesses, and divides into training and testing sets. Real-time water quality parameters are inputted, and unsafe conditions trigger automatic notifications.

3.3.3.5 DATAFLOW DIAGRAM

0 LEVEL DFD

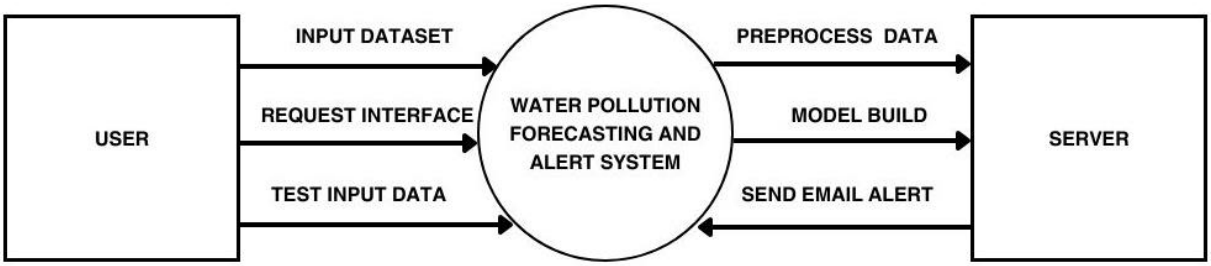


Fig 3.3.3.5 Level 0 Dataflow Diagram

Level 0 DFDs, also known as context diagrams, are the most basic data flow diagrams. They provide a broad view that is easily digestible but offers little detail. Level 0 data flow diagrams show a single process node and its connections to external entities.

FIRST LEVEL DFD

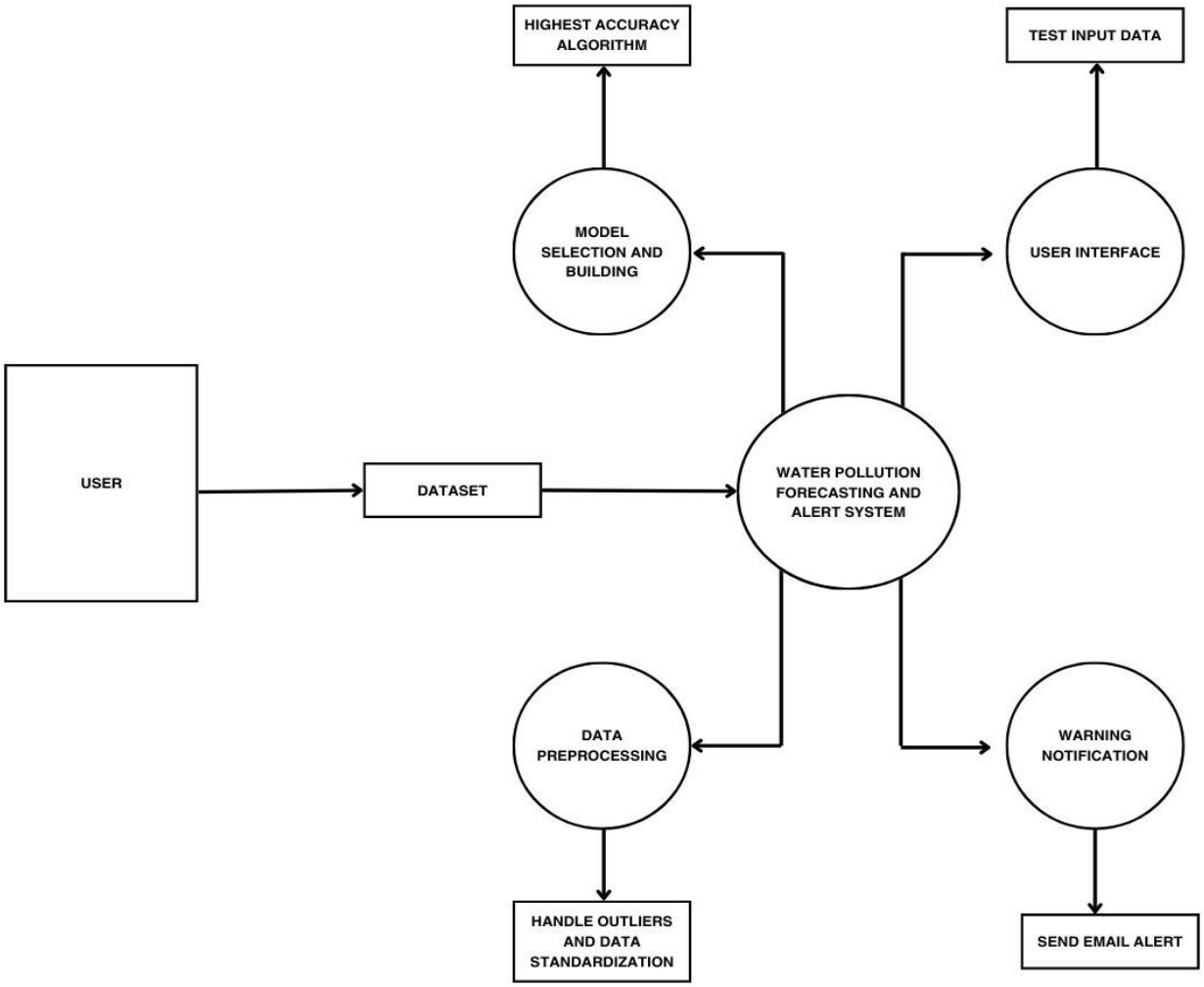


Fig 3.3.3.6 Level 1 Dataflow Diagram

Level 1 DFDs are still a general overview, but they go into more detail than a context diagram. In level 1 DFD, the single process node from the context diagram is broken down into sub-processes. As these processes are added, the diagram will need additional data flows and data stores to link them together.

SECOND LEVEL DFD

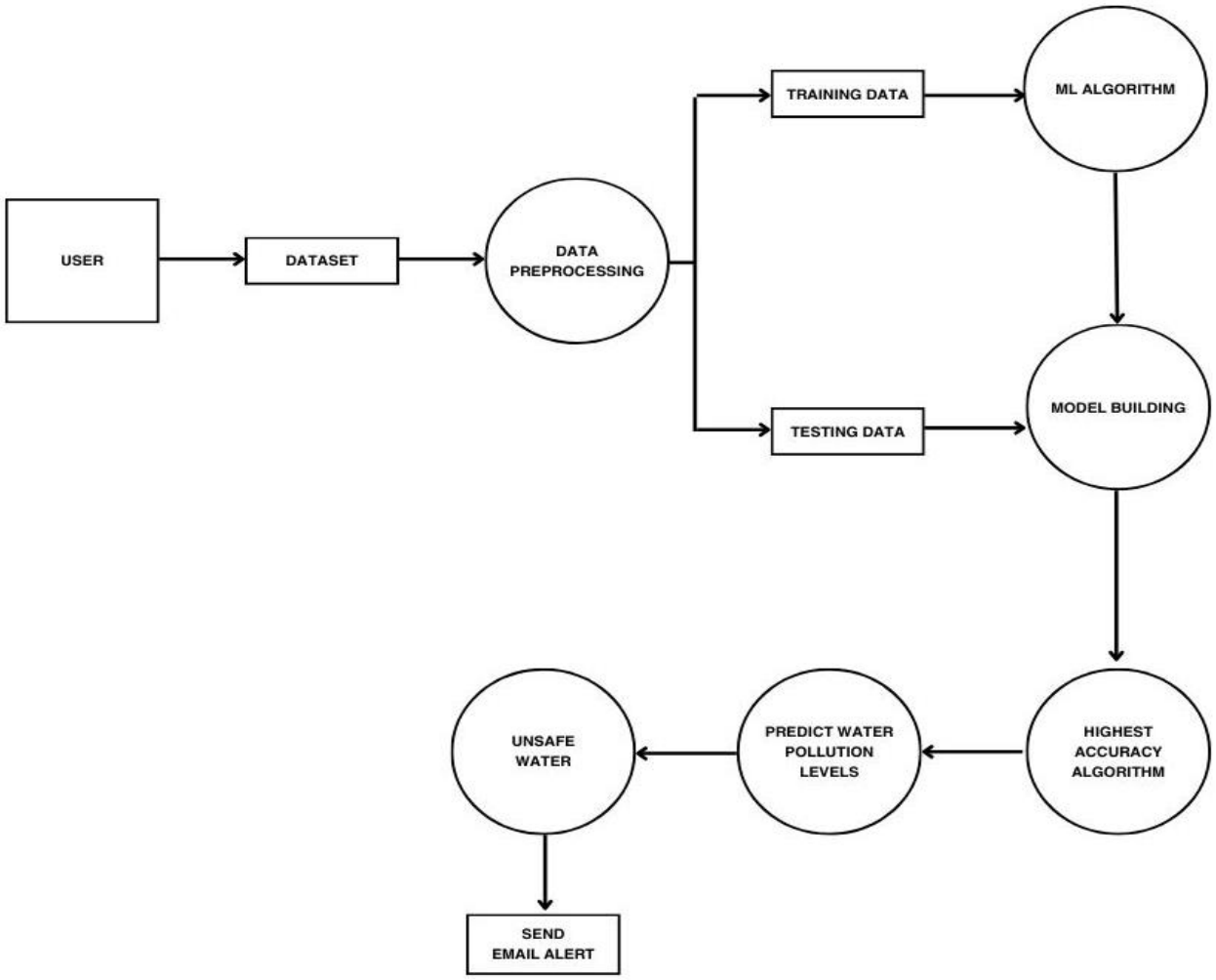


Fig 3.3.3.7 Level 2 Dataflow Diagram

Level 2 DFDs simply break processes down into more detailed sub-processes. In the context of the Water Pollution Forecasting and Alert System using XGBoost Classifier, a Level 2 Data Flow Diagram (DFD) delves deeper into the subprocesses within the key modules identified in the Level 1 DFD.

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 MODULE DESCRIPTION

Water Pollution Forecasting and Alert System consists of 4 main modules, They are:

- DATA PREPROCESSING
- ML ALGORITHMS
- FEATURE EXTRACTION
- MODEL PREDICTION

4.1.1 DATA PREPROCESSING

Water quality forecasting using data analytics and machine learning involves predicting future water quality parameters based on historical data, which can be obtained from platforms like Kaggle. These datasets typically include information on pH levels, chemical concentrations, and biological indicators. To implement this forecasting, data preprocessing is crucial. In this phase, raw datasets are cleaned to handle missing values and outliers, and feature engineering is performed to extract relevant information and create meaningful input variables. Temporal patterns may be captured through data aggregation into time intervals. Scaling and normalization techniques are then applied to ensure uniformity in variable ranges. The preprocessed data is subsequently divided into training and testing sets to train and evaluate machine learning models, enabling accurate water quality predictions. This proactive approach aids in better water resource system management and decision-making.

4.1.2 ML ALGORITHM

Water quality forecasting involves applying data analytics and machine learning algorithms to predict future water quality parameters. Regression algorithms, such as Logistic Regression, can model relationships between various features and water quality

metrics. Time series models, like CNN, are effective for capturing temporal patterns in water quality data. Decision tree-based algorithms, such as XGBoost, handle complex relationships and interactions within the dataset.

4.1.2.1 LOGISTIC REGRESSION

Logistic Regression is a statistical method used for binary classification, where the outcome variable consists of two possible categorical classes. It uses a sigmoid function to transform input features into probabilities, with the hypothesis function separating the input space into different classes. The cost function measures the dissimilarity between predicted and actual class labels. Gradient descent is used to update model parameters. Despite its linear assumption, logistic regression is computationally efficient, making it applicable to medical diagnosis, credit scoring, and marketing. However, it may struggle with complex, non-linear relationships in data.

4.1.2.2 K-NEAREST NEIGHBORS (KNN)

K-Nearest Neighbors (KNN) is a machine learning algorithm used for classification and regression tasks. It predicts query instances by identifying the nearest data points in the feature space. The algorithm assumes similar input instances have similar output values. In classification, the majority class among the 'k' neighbors determines the query instance's class. For regression, the algorithm computes the average of the nearest neighbors' output values. KNN's simplicity makes it easy to implement, but its computational cost can be significant, especially with large datasets. The choice of hyperparameter 'k' significantly influences performance. Despite its sensitivity, KNN is robust and effective in nonlinear decision boundaries or local patterns.

4.1.2.3 CONVOLUTIONAL NEURAL NETWORKS (CNN)

Convolutional Neural Networks (CNNs) are a deep learning technique that is particularly effective in processing structured grid data like images. These networks consist of multiple

layers, including convolutional, pooling, and fully connected layers, which capture hierarchical patterns and spatial relationships within images. They are known for their ability to learn hierarchical representations, making them particularly useful in image recognition tasks. Recent architectures like AlexNet, VGGNet, and ResNet have further improved CNN performance, achieving state-of-the-art results in computer vision applications. CNNs have applications in video analysis, natural language processing, and medical image diagnosis.

4.1.2.4 OPTIMAL PRECISION MODEL

XG BOOST

XGBoost is a powerful machine learning algorithm known for its exceptional performance in predictive modeling tasks. It belongs to the ensemble learning method family and is effective for classification and regression problems. XGBoost uses regularization techniques, parallel computing, and tree pruning to build an ensemble of weak learners, focusing on minimizing errors and minimizing overfitting. It introduces a learning rate to control each tree's contribution to the final prediction. XGBoost's speed, scalability, handling of missing data, and diverse features make it popular in data science competitions and real-world applications. It has proven effective in finance, healthcare, and natural language processing.

4.1.3 FEATURE ENGINEERING

In Water Pollution Forecasting, feature extraction involves identifying and selecting relevant information from raw data. This process entails transforming raw variables, such as chemical concentrations or environmental parameters, into meaningful features that capture key characteristics influencing water quality. Techniques like principal component analysis or wavelet analysis may be applied to extract essential patterns and reduce dimensionality. Feature extraction aims to enhance the input data for machine learning models, emphasizing crucial factors for accurate predictions. The selected features

contribute to a more effective representation of the underlying patterns in the water quality dataset.

4.1.4 MODEL PREDICTION

In water quality forecasting using data analytics and machine learning, output prediction involves forecasting future water quality conditions based on the trained models. When the predicted water quality is deemed abnormal, an automated system can trigger an email notification. This alerting mechanism enhances real-time monitoring and decision-making, enabling timely responses to potential water quality issues. Integrating email notifications into the system provides a proactive means of communicating deviations from expected water quality, facilitating prompt intervention and management actions to maintain water safety and quality.

CHAPTER 5

RESULTS & DISCUSSION

5.1 PERFORMANCE ANALYSIS

Performance analysis is a critical aspect of evaluating the effectiveness and efficiency of a system or process. In the realm of the water pollution forecasting and alert system using machine learning techniques, performance analysis serves as a comprehensive examination of the system's ability to accurately predict water quality, respond to real-time user inputs, and generate timely alerts. This analysis involves assessing the predictive accuracy of the deployed machine learning algorithm in the event of water quality concerns. Performance analysis is pivotal in identifying potential bottlenecks, optimizing algorithmic efficiency, and ensuring that the system meets or exceeds the specified requirements, ultimately contributing to the reliability and success of the water quality forecasting and alert system.

5.1.1 CONFUSION MATRIX:

A confusion matrix is a performance evaluation tool used in machine learning and statistical analysis. It is a table that summarizes the performance of a classification algorithm by comparing predicted and actual classes across different levels of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). In a binary classification problem, the confusion matrix has two classes, positive and negative. A positive outcome is the outcome that the model is trying to predict, and a negative outcome is the opposite of the positive outcome.

The confusion matrix is organized into four cells, as follows:

True Positive (TP): The model correctly predicted a positive outcome.

False Positive (FP): The model incorrectly predicted a positive outcome when the actual outcome was negative.

True Negative (TN): The model correctly predicted a negative outcome.

False Negative (FN): The model incorrectly predicted a negative outcome when the actual outcome was positive.

The following metrics are provided by the confusion matrix to help assess the classification model:

1. **ACCURACY:**

Accuracy is a fundamental metric that measures the overall correctness of a classification model. It represents the ratio of correctly predicted instances to the total instances.

Accuracy = (True Positive + True Negative) / Total Predictions

2. **SENSITIVITY (RECALL):**

Sensitivity, also known as Recall or True Positive Rate, measures how well a model can correctly identify positive instances out of the total actual positive instances.

Recall = True Positive / (True Positive + False Negative)

3. **F-SCORE:**

F-Score, also known as F1-Score, is the harmonic mean of precision and sensitivity. It provides a balanced measure, considering both false positives and false negatives.

F-score = 2 * ((Precision * Sensitivity) / (Precision + Sensitivity))

4. **PRECISION:**

Precision is the ratio of correctly predicted positive instances to the total predicted positive instances. It measures the accuracy of positive predictions.

Precision = True Positive/ True Positive + False Positive

5.1.2 COMPARISON OF ACCURACIES

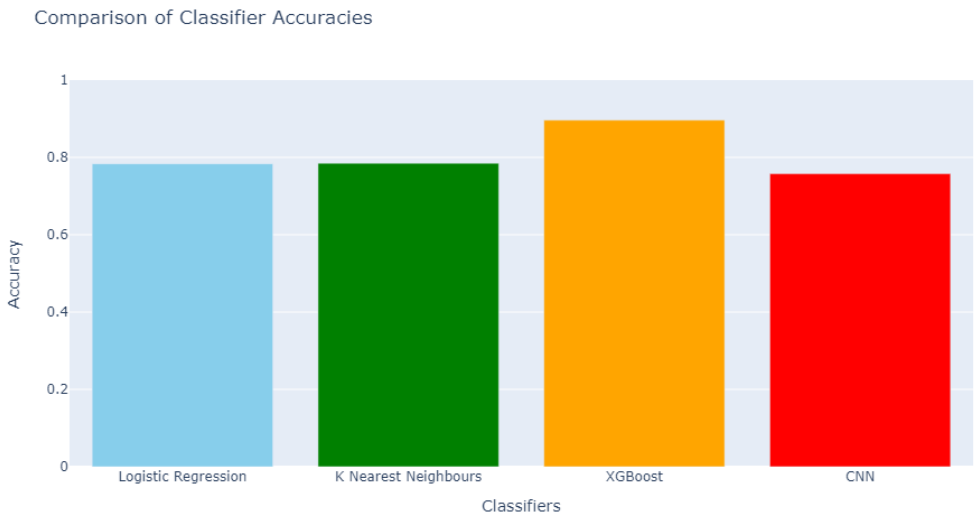


Fig 5.1.2 Comparison of Classifier Accuracies

A study examining four machine learning algorithms - Logistic Regression, K-Nearest Neighbors (KNN), Convolutional Neural Networks (CNN), and XGBoost - was conducted to create a Water Pollution Forecasting and Alert System. The algorithms were tested for accuracy in predicting water pollution levels. Logistic Regression showed 76% accuracy, while KNN and CNN had 77% accuracy. XGBoost, the standout performer, achieved 90% accuracy, demonstrating the strength of ensemble learning and advanced optimization techniques. The study highlights the importance of algorithm selection in machine learning applications. The findings suggest XGBoost holds significant promise for developing a robust system that can enhance water resource management and contribute to environmental conservation efforts by providing accurate and timely alerts.

5.2 RESULTS AND DISCUSSION

The models were trained and evaluated using metrics such as accuracy, precision, recall, and F1-score. After thorough analysis, it was found that XGBoost outperformed CNN and the other algorithms in terms of accuracy. The superior performance of XGBoost can be attributed to its ability to handle complex tabular data, feature importance analysis, and

optimization techniques such as regularization. XGBoost proved more effective in capturing the patterns inherent in water quality data. The interpretability of XGBoost allows for a better understanding of the features influencing water quality predictions. This information is valuable for stakeholders in making informed decisions regarding water resource management and environmental conservation. The algorithm's strength lies in its capacity to capture intricate relationships between various features, providing a nuanced understanding of the underlying patterns influencing water quality. Furthermore, XGBoost’s interpretability proves to be a valuable asset, allowing for a transparent analysis of feature importance and contributing factors in the prediction process. The optimization techniques integrated into XGBoost, such as regularization, enhance its generalization capabilities and make it particularly well-suited for the challenges presented by water quality prediction. Overall, the high accuracy achieved by XGBoost positions it as a robust and reliable choice for water quality modeling, with potential applications in environmental monitoring and resource management.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

- In conclusion, the proposed water quality forecasting system combines the robust predictive capabilities of machine learning models with a user-friendly web interface.
- The emphasis on accuracy ensures reliable pollution risk assessments, fostering informed decision-making.
- Additionally, the proactive notification feature empowers users to contribute actively to environmental well-being by promptly alerting authorities when necessary.
- This holistic approach not only enhances the accessibility of water quality information but also encourages a collective effort towards sustainable water management.

6.2 FUTURE ENHANCEMENTS

For future enhancements, the project envisions an integration of IoT devices and the incorporation of marine robots equipped with sensors. This expansion aims to enhance the monitoring capabilities of the water quality prediction system by deploying intelligent devices that can navigate through water resources. By leveraging IoT technology and marine robots, the system will achieve real-time data collection from diverse points, providing a more comprehensive understanding of water quality parameters. This integration promises to elevate the accuracy and efficiency of the system, enabling it to adapt to dynamic environmental conditions and ensuring a more proactive approach to water quality forecasting and alerting.

APPENDICES

A.1 SDG GOALS

The Sustainable Development Goals (SDGs) are a set of global goals adopted by the United Nations to address various social, economic, and environmental challenges. The Water Pollution Forecasting and Alert System described may align with several SDGs.

1. **SDG 6: Clean Water and Sanitation:**

   Target 6.3: Improve water quality by reducing pollution, eliminating dumping, and minimizing the release of hazardous chemicals and materials.

2. **SDG 9: Industry, Innovation, and Infrastructure:**

   Target 9.5: Enhance scientific research, upgrade technological capabilities, and increase access to information and communication technology.

3. **SDG 11: Sustainable Cities and Communities:**

   Target 11.6: Reduce the adverse environmental impact of cities, paying special attention to air quality, municipal and other waste management.

4. **SDG 13: Climate Action:**

   Target 13.3: Improve education, awareness-raising, and human and institutional capacity on climate change mitigation, adaptation, impact reduction, and early warning.

5. **SDG 14: Life Below Water:**

   Target 14.1: Prevent and significantly reduce marine pollution of all kinds, particularly from land-based activities, including marine debris and nutrient pollution.

By providing real-time water quality predictions and timely alerts, this project contributes to the achievement of these goals by promoting sustainable water management, environmental protection, and the use of technology for monitoring and early warning systems.

A.2 SOURCE CODE

```
import pandas as pd
import numpy as np
import seaborn as sns

df = pd.read_csv('E:/Ana/Files/WaterQuality forecasting/WaterQuality-
main/data/water_potability.csv')

df.head()

df.isnull().sum()

df.drop_duplicates(inplace=True)

df.dropna(how='all', inplace=True)

df.describe()

df.Potability.value_counts()

import matplotlib.pyplot as plt
plt.figure(figsize=(10,10))
sns.heatmap(data=df.corr(), annot=True, cmap='BrBG')

idx1 = df.query('Potability == 1')['ph'][df.ph.isna()].index
df.loc[idx1, 'ph'] = df.query('Potability == 1')['ph'][df.ph.notna()].mean()
idx0 = df.query('Potability == 0')['ph'][df.ph.isna()].index
df.loc[idx0,'ph'] = df.query('Potability==0')['ph'][df.ph.notna()].mean()

idx1 = df.query('Potability == 1')['Sulfate'][df.Sulfate.isna()].index
df.loc[idx1, 'Sulfate'] = df.query('Potability == 1')['Sulfate'][df.Sulfate.notna()].mean()
idx0 = df.query('Potability == 0')['Sulfate'][df.Sulfate.isna()].index
df.loc[idx0,'Sulfate'] = df.query('Potability==0')['Sulfate'][df.Sulfate.notna()].mean()

idx1 = df.query('Potability == 1')['Trihalomethanes'][df.Trihalomethanes.isna()].index
df.loc[idx1, 'Trihalomethanes'] = df.query('Potability ==
1')['Trihalomethanes'][df.Trihalomethanes.notna()].mean()
idx0 = df.query('Potability == 0')['Trihalomethanes'][df.Trihalomethanes.isna()].index
df.loc[idx0,'Trihalomethanes'] =
df.query('Potability==0')['Trihalomethanes'][df.Trihalomethanes.notna()].mean()

df.loc[~df.ph.between(6.5, 8.5), 'Potability'] = 0
```

```
df.isna().sum()

X = df.drop(['Potability'], axis = 1).values
y = df['Potability'].values

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X = sc.fit_transform(X)

from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.2)

#Hyperparameter tuning ;)

lr = LogisticRegression(random_state=42)

knn = KNeighborsClassifier()

#xgb =XGBClassifier(eval_metric = 'logloss')

para_knn = {'n_neighbors':np.arange(1, 50)}  #parameters of knn
grid_knn = GridSearchCV(knn, param_grid=para_knn, cv=5) #search knn for 5 fold cross validation

#XGBoost
#parameters for xgboost
#params_xgb = {'n_estimators': [50,100,250,400,600,800,1000], 'learning_rate': [0.2,0.5,0.8,1]}
#rs_xgb =  RandomizedSearchCV(xgb, param_distributions=params_xgb, cv=5)
#Hyperparameter tuning ;)

lr = LogisticRegression(random_state=42)

knn = KNeighborsClassifier()

#xgb =XGBClassifier(eval_metric = 'logloss')
```

```
para_knn = {'n_neighbors':np.arange(1, 50)}  #parameters of knn
grid_knn = GridSearchCV(knn, param_grid=para_knn, cv=5) #search knn for 5 fold cross validation

#XGBoost
#parameters for xgboost
#params_xgb = {'n_estimators': [50,100,250,400,600,800,1000], 'learning_rate': [0.2,0.5,0.8,1]}
#rs_xgb =  RandomizedSearchCV(xgb, param_distributions=params_xgb, cv=5)
from xgboost import XGBClassifier
from sklearn.model_selection import RandomizedSearchCV

# XGBoost
xgb = XGBClassifier(eval_metric='logloss')

# Additional hyperparameter tuning for XGBoost
params_xgb = {
    'n_estimators': [200, 300, 400, 500, 600],
    'learning_rate': [0.01, 0.05, 0.1, 0.2],
    'max_depth': [3, 4, 5, 6, 7],
    'min_child_weight': [1, 2, 3, 4],
    'subsample': [0.8, 0.9, 1.0],
    'gamma': [0, 0.1, 0.2, 0.3],
    'colsample_bytree': [0.8, 0.9, 1.0],
    'reg_alpha': [0, 0.1, 0.5, 1.0],
    'reg_lambda': [0, 0.1, 0.5, 1.0],
}

rs_xgb = RandomizedSearchCV(xgb, param_distributions=params_xgb, cv=5, n_iter=200, random_state=42)
rs_xgb.fit(X_train, y_train)

from xgboost import XGBClassifier

grid_knn.fit(X_train, y_train)
rs_xgb = XGBClassifier(n_jobs=-1, random_state=42)  # Assuming use_label_encoder is not explicitly needed
rs_xgb.fit(X_train, y_train)

print("Best parameters for KNN:", grid_knn.best_params_)
print("Best parameters for XGBoost:", rs_xgb.get_params())

lr = LogisticRegression(random_state=42)
```

```
knn = KNeighborsClassifier(n_neighbors=16)

xgb = XGBClassifier(n_estimators= 100, learning_rate= 0.3)

classifiers = [('Logistic Regression', lr), ('K Nearest Neighbours', knn),
               ('XGBoost', xgb)]

from sklearn.metrics import accuracy_score

for classifier_name, classifier in classifiers:

    # Fit clf to the training set
    classifier.fit(X_train, y_train)

    # Predict y_pred
    y_pred = classifier.predict(X_test)
    accuracy = accuracy_score(y_test,y_pred)
    # Evaluate clf's accuracy on the test set
    print('{:s} : {:.2f}'.format(classifier_name, accuracy))

from sklearn.metrics import classification_report

y_pred_rf= xgb.predict(X_test)
print(classification_report(y_test, y_pred_rf))

from sklearn.metrics import classification_report, precision_score, recall_score,
confusion_matrix
print(precision_score(y_test, y_pred_rf))
print(recall_score(y_test, y_pred_rf))
print(confusion_matrix(y_test, y_pred_rf))


# CNN ALGORITHM

import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler

# Assuming X is your feature matrix and y is your target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
# Standardize the feature values
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Define the number of features
num_features = X_train.shape[1]

# Define the number of classes
num_classes = len(set(y_train))

# Define the dense neural network model
model = tf.keras.Sequential([
    tf.keras.layers.Dense(32, activation='relu', input_shape=(num_features,)),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(num_classes, activation='softmax')
])

# Compile the model
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train, epochs=50, validation_data=(X_test, y_test))

# Evaluate the model on the test set
test_loss, test_acc = model.evaluate(X_test, y_test, verbose=2)
print('\nTest accuracy:', test_acc)

y_pred = model.predict(X_test)
y_pred_classes = tf.argmax(y_pred, axis=1).numpy()

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred_classes)
print(f'Accuracy: {accuracy * 100:.2f}%')

conf_matrix = confusion_matrix(y_test, y_pred_classes)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
```

```
plt.title('Confusion Matrix')
plt.show()

#print accuracies for all algorithms

from sklearn.metrics import accuracy_score

# Assuming classifiers is a list of tuples containing classifier_name and classifier
accuracies = {}

for classifier_name, classifier in classifiers:
    # Fit clf to the training set
    classifier.fit(X_train, y_train)

    # Predict y_pred
    y_pred = classifier.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    # Store the accuracy in the dictionary
    accuracies[classifier_name] = accuracy
# Now print the accuracies for all algorithms
for classifier_name, accuracy in accuracies.items():
    print('{:s} : {:.2f}'.format(classifier_name, accuracy))

# Print CNN accuracy
print('CNN : {:.2f}'.format(test_acc))

import plotly.graph_objects as go
from sklearn.metrics import accuracy_score

# Colors for each classifier
colors = ['skyblue', 'green', 'orange', 'red', 'purple']

# Assuming classifiers is a list of tuples containing classifier_name and classifier
classifier_names, accuracies = zip(*[(classifier_name,
accuracy_score(classifier.predict(X_test), y_test))
                                     for classifier_name, classifier in classifiers])

# Calculate CNN accuracy
# Assuming cnn_model is your trained CNN model
accuracy_cnn = accuracy_score(model.predict(X_test).argmax(axis=1), y_test)
classifier_names += ('CNN',)
accuracies += (accuracy_cnn,)
```

```
# Now print the accuracies for all algorithms
for classifier_name, accuracy in zip(classifier_names, accuracies):
    print('{:s} : {:.2f}'.format(classifier_name, accuracy))

# Create an interactive radar chart with Plotly
fig = go.Figure()

for classifier_name, color, accuracy in zip(classifier_names, colors, accuracies):
    fig.add_trace(go.Scatterpolar(
        r=[accuracy],
        theta=[classifier_name],
        fill='toself',
        name=classifier_name,
        line=dict(color=color),
    ))

fig.update_layout(
    polar=dict(
        radialaxis=dict(
            visible=True,
            range=[0, 1]
        )
    ),
    showlegend=True,
    title='Comparison of Classifier Accuracies',
)

# Show the interactive chart
fig.show()
import plotly.graph_objects as go
from sklearn.metrics import accuracy_score

# Colors for each classifier
colors = ['skyblue', 'green', 'orange', 'red', 'purple']
# Assuming classifiers is a list of tuples containing classifier_name and classifier
accuracies = {}
for i, (classifier_name, classifier) in enumerate(classifiers):
    # Fit clf to the training set
    classifier.fit(X_train, y_train)
    # Predict y_pred
    y_pred = classifier.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    # Store the accuracy in the dictionary
```

```
    accuracies[classifier_name] = accuracy

# Calculate CNN accuracy
# Assuming cnn_model is your trained CNN model
y_pred_cnn = model.predict(X_test).argmax(axis=1)
accuracy_cnn = accuracy_score(y_test, y_pred_cnn)
accuracies['CNN'] = accuracy_cnn
# Now print the accuracies for all algorithms
for classifier_name, accuracy in accuracies.items():
    print('{:s} : {:.2f}'.format(classifier_name, accuracy))
# Create an interactive bar chart with Plotly
fig = go.Figure()
fig.add_trace(go.Bar(x=list(accuracies.keys()), y=list(accuracies.values()),
marker_color=colors))
fig.update_layout(title='Comparison of Classifier Accuracies', xaxis_title='Classifiers',
yaxis_title='Accuracy', yaxis_range=[0, 1])
# Show the interactive chart
fig.show()
import pickle
filename = 'xgboost.sav'
pickle.dump(xgb, open(filename, 'wb'))
# load the model from disk
loaded_model = pickle.load(open('xgboost.sav', 'rb'))

filename = 'scaler.sav'
pickle.dump(sc, open(filename, 'wb'))
# load the model from disk
scc = pickle.load(open('scaler.sav', 'rb'))
import pickle
loaded_model = pickle.load(open('xgboost.sav', 'rb'))
scc = pickle.load(open('scaler.sav', 'rb'))
data = df.iloc[3:4, :-1].values
data
sc_data = scc.transform(data)
sc_data
loaded_model.predict(sc_data)
df
def plot_confusion_matrix(model):
    y_pred = model.predict(X_test)
    cm = confusion_matrix(y_test, y_pred)
    conf_matrix = pd.DataFrame(data = cm,columns = ['Predicted:0','Predicted:1'], index =
['Actual:0','Actual:1'])
    sns.heatmap(conf_matrix, annot = True, fmt = 'd', cmap ="winter", cbar = False,
```

```
            linewidths = 0.1, annot_kws = {'size':25})
    plt.xticks(fontsize = 20)
    plt.yticks(fontsize = 20)

    # display the plot
    plt.show()

from sklearn.metrics import classification_report
# Assuming you have already trained your logistic regression model (lr) and have X_test,
y_test

# Make predictions
y_pred_lg = lr.predict(X_test)

# Print classification report
print(classification_report(y_test, y_pred_lg, zero_division=1))  # You can set
zero_division to 'warn', 1, or any other value

# Plot confusion matrix
plot_confusion_matrix(lr)

from sklearn.metrics import classification_report
y_pred_knn= knn.predict(X_test)
print(classification_report(y_test, y_pred_knn))
plot_confusion_matrix(knn)

from sklearn.metrics import classification_report
y_pred_rf= xgb.predict(X_test)
print(classification_report(y_test, y_pred_rf))
plot_confusion_matrix(xgb)

lr = LogisticRegression(random_state=42)
lr.fit(X_train, y_train)
y_pred_lr = lr.predict(X_test)

xgb = XGBClassifier(n_estimators=250, learning_rate=0.2)
xgb.fit(X_train, y_train)
y_pred_xgb = xgb.predict(X_test)

import seaborn as sns
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
```

```
# Logistic Regression
cm_lr = confusion_matrix(y_test, y_pred_lr)
sns.heatmap(cm_lr, annot=True, fmt='d', cmap='Blues', xticklabels=['0', '1'],
yticklabels=['0', '1'])
plt.title('Logistic Regression - Confusion Matrix')
plt.show()

# K Nearest Neighbours
cm_knn = confusion_matrix(y_test, y_pred_knn)
sns.heatmap(cm_knn, annot=True, fmt='d', cmap='Blues', xticklabels=['0', '1'],
yticklabels=['0', '1'])
plt.title('K Nearest Neighbours - Confusion Matrix')
plt.show()

# XGBoost
cm_xgb = confusion_matrix(y_test, y_pred_xgb)
sns.heatmap(cm_xgb, annot=True, fmt='d', cmap='Blues', xticklabels=['0', '1'],
yticklabels=['0', '1'])
plt.title('XGBoost - Confusion Matrix')
plt.show()

# CNN
cm_cnn = confusion_matrix(y_test, y_pred_cnn)
sns.heatmap(cm_cnn, annot=True, fmt='d', cmap='Blues', xticklabels=['0', '1'],
yticklabels=['0', '1'])
plt.title('CNN - Confusion Matrix')
plt.show()
```

A.3 SCREENSHOTS

Fig A.3.1 Output Screen

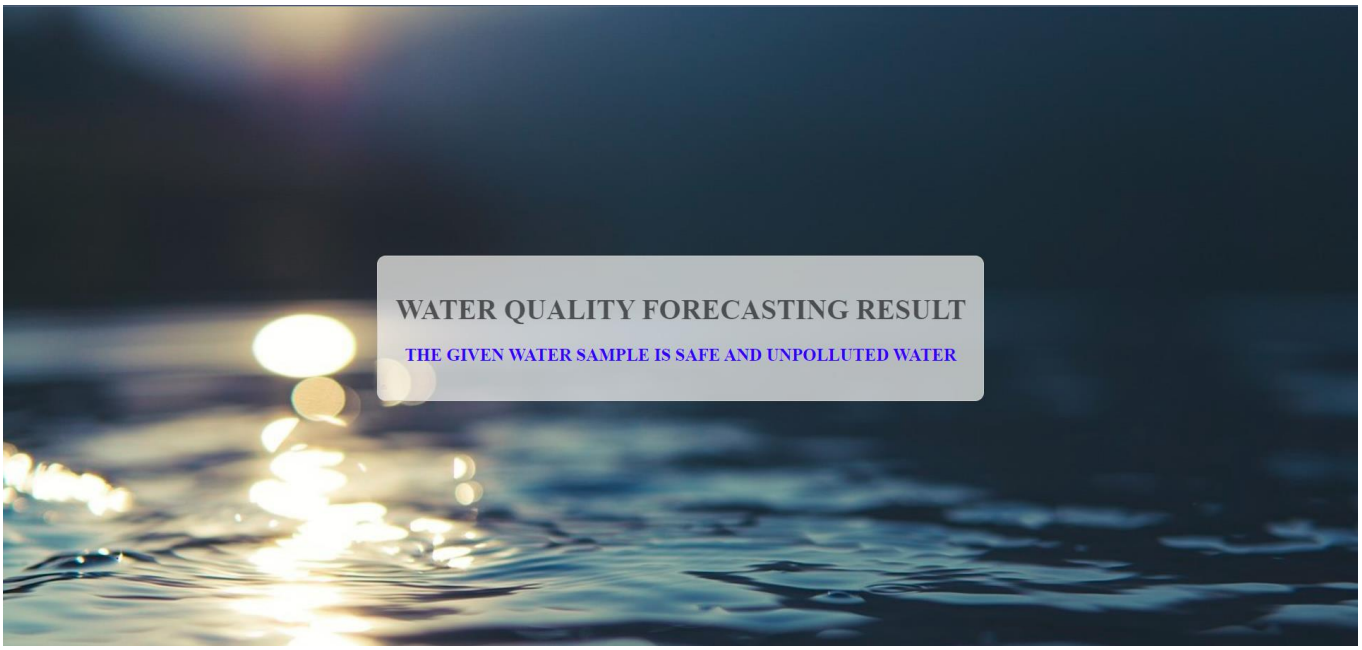


Fig A.3.2 Result Screen – Safe Water

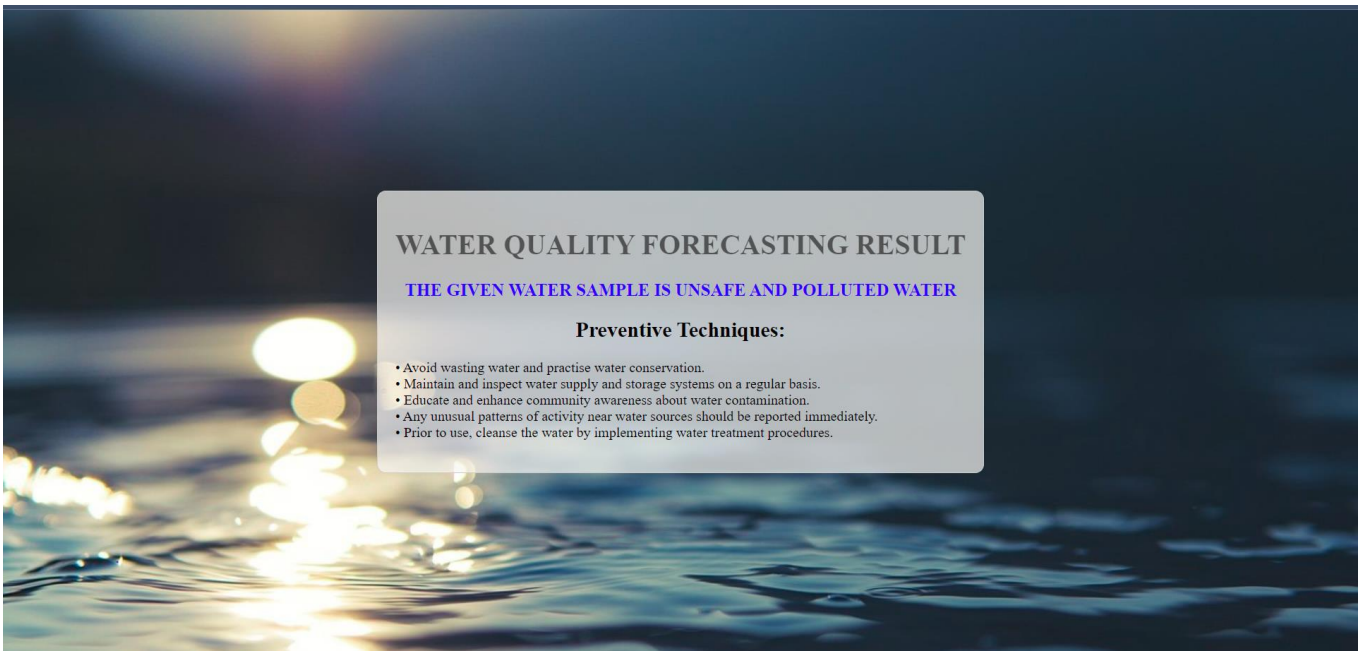


Fig A.3.3 Result Screen – Unsafe Water

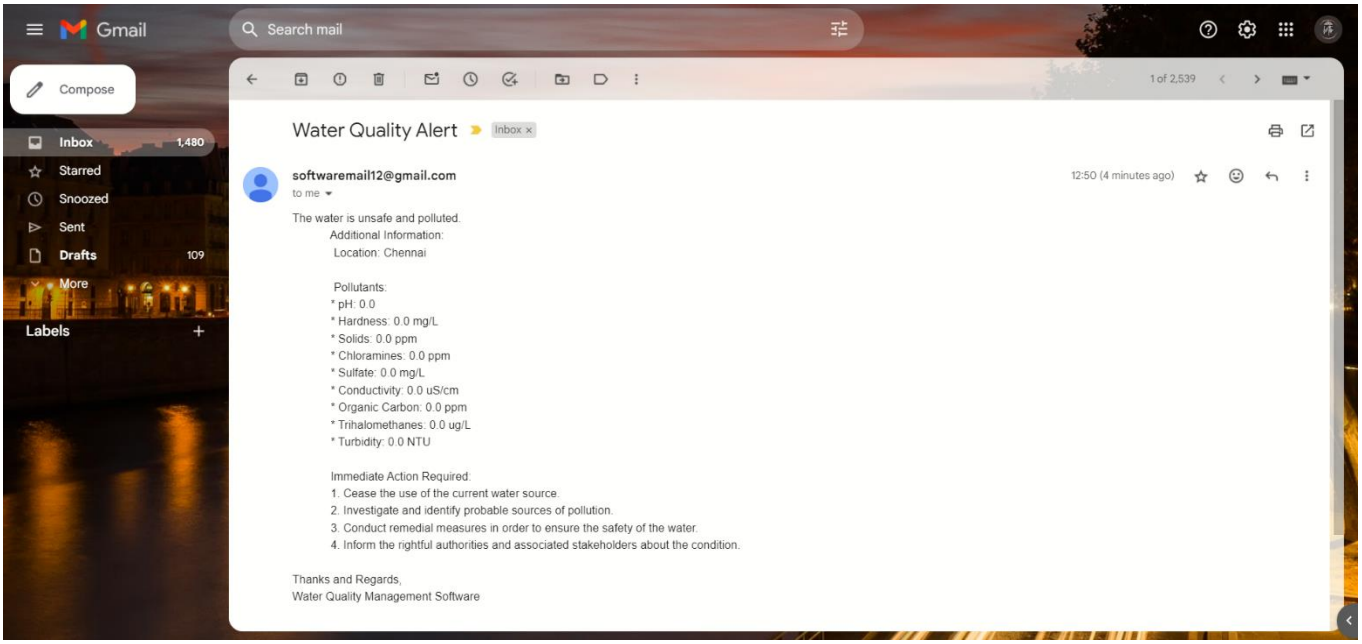
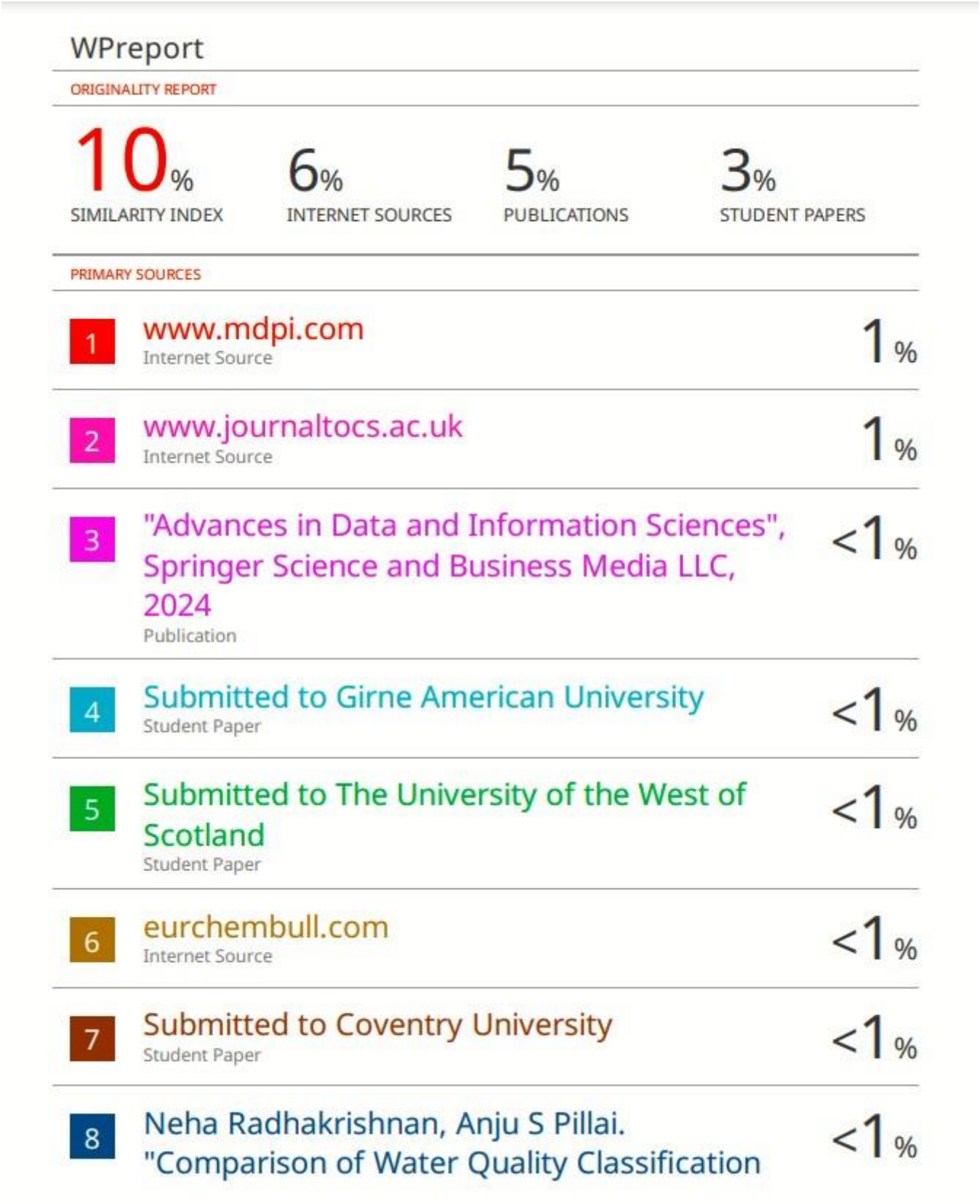


Fig A.3.3 Autogenerated Email Screen

A.4 PLAGIARISM REPORT



WPreport

ORIGINALITY REPORT

10%	6%	5%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.mdpi.com Internet Source	1%
2	www.journaltocs.ac.uk Internet Source	1%
3	"Advances in Data and Information Sciences", Springer Science and Business Media LLC, 2024 Publication	<1%
4	Submitted to Girne American University Student Paper	<1%
5	Submitted to The University of the West of Scotland Student Paper	<1%
6	eurchembull.com Internet Source	<1%
7	Submitted to Coventry University Student Paper	<1%
8	Neha Radhakrishnan, Anju S Pillai. "Comparison of Water Quality Classification	<1%

A.5 PAPER PUBLICATION

Title: Water Pollution Forecasting and Alert System Using XGBoost Classifier.

Conference: 7th International Conference on Intelligent Computing (ICONIC2024)

Indexing: The conference proceedings will be indexed in Scopus, ensuring wide visibility and accessibility to the academic community.

Authors: Thelma Princy M, Kanchana A

Affiliation: Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India.

Conference Track: ICONIC2024

Conference Date: 22nd & 23rd March 2024

Abstract Submission Date: 4th March 2024

Paper Submission Date: 4th March 2024

REFERENCES

[1] Jalagam, Lokesh, Nathaniel Shepherd, Jingyi Qi, Nicole Barclay, and Michael Smith. "Water Quality Predictions for Urban Streams Using Machine Learning." In SoutheastCon 2023, pp. 217-223. IEEE, 2023.

[2] Wang, Mingyan, Qing Xu, Yingying Cao, Shahbaz Gul Hassan, Wenjun Liu, Min He, Tonglai Liu et al. "An Ensemble Model for Water Temperature Prediction in Intensive Aquaculture." IEEE Access 11 (2023): 137285-137302.

[3] Liu, Wenjun, Shuangyin Liu, Shahbaz Gul Hassan, Yingying Cao, Longqin Xu, Dachun Feng, Liang Cao et al. "A Novel Hybrid Model to Predict Dissolved Oxygen for Efficient Water Quality in Intensive Aquaculture." IEEE Access 11 (2023): 29162-29174.

[4] Omambia, Andrew, Benard Maake, and Anthony Wambua. "Water quality monitoring using IoT & machine learning." In 2022 IST-Africa Conference (IST-Africa), pp. 1-8. IEEE, 2022.

[5] Zhu, Mengyuan, Jiawei Wang, Xiao Yang, Yu Zhang, Linyu Zhang, Hongqiang Ren, Bing Wu, and Lin Ye. "A review of the application of machine learning in water quality evaluation." Eco-Environment & Health (2022).

[6] Kenchannavar, Harish H., Prasad M. Pujar, Raviraj M. Kulkarni, and Umakant P. Kulkarni. "Evaluation and Analysis of Goodness of Fit for Water Quality Parameters Using Linear Regression Through the Internet-of-Things-Based Water Quality Monitoring System." IEEE Internet of Things Journal 9, no. 16 (2021): 14400-14407.

[7] Nair, Jitha P., and M. S. Vijaya. "Predictive models for river water quality using machine learning and big data techniques-a Survey." In 2021 international conference on artificial intelligence and smart systems (ICAIS), pp. 1747-1753. IEEE, 2021.

[8] Rostam, Nur Aqilah Paskhal, Nurul Hashimah Ahamed Hassain Malim, Rosni Abdullah, Abdul Latif Ahmad, Boon Seng Ooi, and Derek Juinn Chieh Chan. "A complete proposed framework for coastal water quality monitoring system with algae predictive model." IEEE Access 9 (2021): 108249-108265.

[9] Radhakrishnan, Neha, and Anju S. Pillai. "Comparison of water quality classification models using machine learning." In 2020 5th International Conference on Communication and Electronics Systems (ICCES), pp. 1183-1188. IEEE, 2020.

[10] Wu, Di, Hao Wang, Hadi Mohammed, and Razak Seidu. "Quality risk analysis for sustainable smart water supply using data perception." IEEE transactions on sustainable computing 5, no. 3 (2019): 377-388.