pH and Turbidity: Multi-Parameter Water Quality Monitoring Using Edge-Integrated Sensing Platforms

A Project Report submitted in the partial fulfillment
of the Requirements for the award of the degree

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
Submitted by

Alajangi Keerthisree (22471A05E3)
Katari Thanmai (22471A05G1)
Sonti Vineela (22471A05J6)

Under the esteemed guidance of

M.Suneetha, B.Tech., M.Tech.

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET
(AUTONOMOUS)
Accredited by NAAC with A+ Grade and NBA under and
an ISO 9001:2015 Certified
Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada
KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET- 522601

2025-2026

NARASARAOPETA ENGINEERING COLLEGE

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name "pH and Turbidity: Multi-Parameter Water Quality Monitoring Using Edge-Integrated Sensing Platforms" is a bonafide work done by the team Alajangi Keerthisree (22471A05E3), Katari Thanmai (22471A05G1), Sonti Vineela (22471A05J6) BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2025-2026.

PROJECT GUIDE

M.Suneetha, B.Tech., M.Tech.
Associate Professor

PROJECT CO-ORDINATOR

D.Venkata Reddy, B.Tech., M.Tech., (Ph.D).
Assistant Professor

HEAD OF THE DEPARTMENT

Dr. S. N. Tirumala Rao, M.Tech., Ph.D.
Professor & HOD

EXTERNAL EXAMINER

DECLARATION

We declare that this project work titled "PH AND TURBIDITY: MULTI-PARAMETER WATER QUALITY MONITORING USING EDGE-INTEGRATED SENSING PLATFORMS " is composed by ourselves that the work contains here is our own except where explicitly stated otherwise in the text and that this work has been not submitted for any other degree or professional qualification except as specified.

Alajangi Keerthisree (22471A05E3)
Katari Thanmai (22471A05G1)
Sonti Vineela (22471A05J6)

ACKNOWLEDGEMENT

We wish to express our thanks to various personalities who are responsible for the completion of my project. We are extremely thankful to our beloved chairman**, Sri M. V. Koteswara Rao, B.Sc.**, who took keen interest in us in every effort throughout this course. We owe our sincere gratitude to our beloved principal**, Dr. S. Venkateswarlu, Ph.D.**, for showing his kind attention and valuable guidance throughout the course.

We express our deep-felt gratitude towards **Dr. S. N. Tirumala Rao**, **M.Tech., Ph.D.**, HOD of the CSE department, and also to our guide**, M.Suneetha, B.Tech., M.Tech.,** Professor of the CSE department, whose valuable guidance and unstinting encouragement enabled us to accomplish our project successfully in time.

We extend our sincere thanks to **D. Venkat Reddy, B.Tech., M.Tech., (Ph.D.),** Assistant Professor & Project Coordinator of the project, for extending his encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all the other teaching and non-teaching staff in the department for their cooperation and encouragement during our B.Tech. degree.

We have no words to acknowledge the warm affection, constant inspiration, and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions and clarifying our doubts, which had really helped us in successfully completing our project.

By

Alajangi Keerthisree (22471A05E3)

Katari Thanmai (22471A05G1)

Sonti Vineela (22471A05J6)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a center of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to the academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

Program Outcomes

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, research literature and analyze complex engineering problems reaching substantiated conclusions With consideration for sustainable development. (WK1 to WK4)

PO3: Design/development of solutions: Design solutions for complex engineering problems and design/develop system components or processes to meet the identified needs with consideration for the public health and safety , whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

PO4: Conduct investigations of complex problems: Conduct investigation of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8)

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO8: Individual and Collaborative Team Work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural , language, and learning differences.

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision making and apply these to one's own work, as a member and leader in a team,and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologys and iii) critical thinking in the bradest context of technological change.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S):

CO421.1: Analyse the System of Examinations and identify the problem.
CO421.2: Identify and classify the requirements.
CO421.3: Review the Related Literature
CO421.4: Design and Modularize the project
CO421.5: Construct, Integrate, Test and Implement the Project.
CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
C421.1		✓										✓		
C421.2	✓		✓		✓							✓		
C421.3				✓		✓	✓	✓				✓		
C421.4			✓			✓	✓	✓				✓	✓	
C421.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C421.6									✓	✓	✓	✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1	2	3											2		
C421.2			2		3								2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. Low level
2. Medium level
3. High level

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	Description of the device	Attained PO
C2204.2, C22L3.2	Collected and analyzed the water-quality dataset and planned the machine learning and deep learning model architecture for pollution severity prediction.	PO1, PO3
CC421.1, C2204.3, C22L3.2	Performed requirement analysis and implemented preprocessing steps such as missing-value handling, normalization, feature engineering, and PSI creation.	PO2, PO3
CC421.2, C2204.2, C22L3.3	Designed project workflow including data pipeline, feature extraction, feature selection (RF-based), and multi-model comparison (RF, FNN, LSTM, GRU).	PO3, PO5, PO9
CC421.3, C2204.3, C22L3.2	Trained, validated, and tested deep learning models including optimized GRU with performance evaluation using accuracy, confusion matrix, and generalization gap.	PO1, PO5
CC421.4, C2204.4, C22L3.2	Documentation is done by all our four members in the form of a group	PO10
CC421.5, C2204.2, C22L3.3	Prepared project documentation, maintained experimental logs, and collaborated as a team for analysis and result interpretation.	PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Deployed the trained GRU model and Random Forest model for prediction use-cases, saving models in a portable format for real-time usage.	PO4, PO7

C32SC4.3	Developed a simple output interface to accept water-quality parameters and provide predicted PSI category (Low, Moderate , Severe, Critical).	PO5, PO6

ABSTRACT

Water quality monitoring is one of the most important environmental concerns in today's world. Traditional systems rely on expensive sensors and laboratory testing, which are often time-consuming and not suitable for continuous monitoring. This project, "pH and Turbidity: Multi-Parameter Water Quality Monitoring Using Edge-Integrated Sensing Platforms," proposes a smart, low-cost, and sensor-free approach for analyzing water quality using machine learning (ML) and deep learning (DL) techniques.

The system uses a publicly available dataset containing key water quality parameters such as pH, turbidity, chloramines, solids, conductivity, sulfate, and organic carbon. The data is preprocessed, normalized, and used to train multiple models — including Random Forest (RF), Feedforward Neural Network (FNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and the proposed DistilBERT model. These models predict the Pollution Severity Index (PSI) and classify the water quality into categories like *Low, Moderate, Severe,* and *Critical*.

Among all models, the DistilBERT model achieved the highest accuracy of around 95%, outperforming other traditional and deep learning models in both performance and speed. The system can be implemented on edge devices like Raspberry Pi for real-time and portable monitoring. It provides an accurate, energy-efficient, and scalable solution for assessing water quality without the need for costly sensors.

Overall, this project contributes to Sustainable Development Goal 6 (Clean Water and Sanitation) by offering an intelligent, affordable, and eco-friendly solution for water pollution detection and management.

INDEX

LIST OF FIGURES

List of Tables

1.INTRODUCTION

Water is one of the most essential natural resources for all forms of life. The quality of water plays a vital role in human health, agriculture, and the environment. However, rapid industrialization, urbanization, and pollution have led to a significant decline in water quality across the world. Monitoring and managing water quality has therefore become a critical task to ensure safe and sustainable water use. Traditional methods of water testing involve manual sampling and laboratory analysis, which are time-consuming, costly, and not suitable for real-time monitoring.

This project, titled "pH and Turbidity: Multi-Parameter Water Quality Monitoring Using Edge-Integrated Sensing Platforms," aims to develop a smart, automated, and low-cost system for analyzing water quality using machine learning (ML) and deep learning (DL) techniques. The proposed system uses a publicly available dataset containing multiple physicochemical parameters such as pH, turbidity, chloramines, solids, conductivity, sulfate, and organic carbon. These parameters are processed and analyzed using different AI models to determine the overall Pollution Severity Index (PSI) and classify the water into categories such as *Low, Moderate, Severe,* or *Critical pollution levels.*

In this system, several ML and DL models — including Random Forest (RF), Feedforward Neural Network (FNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and the proposed DistilBERT model — are trained and evaluated to find the most accurate and efficient one. The DistilBERT model achieved the highest accuracy of about 95%, making it the best choice for reliable water quality prediction. The system is lightweight and can be deployed on edge devices such as Raspberry Pi for real-time water monitoring in both urban and rural areas.

The proposed method eliminates the need for expensive sensors and complex hardware setups. It provides a sensor-free, energy-efficient, and scalable solution for monitoring water quality. This project not only demonstrates the power of artificial intelligence in solving environmental challenges but also supports Sustainable Development Goal 6 (Clean Water and Sanitation) by helping ensure access to safe and clean water for all.

1.1 Motivation

Access to clean and safe water is one of the most pressing global challenges of the 21st century. Rapid industrialization, urban expansion, and climate change have severely impacted water quality, threatening both human health and environmental sustainability. Traditional water quality monitoring systems rely heavily on IoT-based sensors and expensive infrastructures that are often impractical for rural and resource-limited areas. These constraints highlight the urgent need for an alternative solution that is cost-effective, scalable, and efficient.

The motivation behind this project stems from the desire to bridge this technological and environmental gap. By leveraging machine learning and deep learning models, such as GRU, LSTM, and Random Forest, this work aims to create a software-based, sensor-free system capable of accurately classifying water quality using publicly available datasets. This approach reduces dependency on costly sensors and maintenance-heavy hardware, enabling widespread adoption even in low-resource settings.

Furthermore, this project aligns with Sustainable Development Goal 6 (SDG-6) — ensuring availability and sustainable management of water and sanitation for all. By introducing an intelligent, data-driven framework for monitoring physicochemical parameters like pH, turbidity, chloramines, and solids, the system supports proactive decision-making in environmental management.

Ultimately, this project is driven by a vision to use technology as a force for social good — to make clean water monitoring more accessible, affordable, and reliable for communities worldwide. It represents a step toward integrating innovation and sustainability to protect one of humanity's most vital resources.

1.2 Problem Statement

Access to safe and clean drinking water remains a critical challenge across the world, especially in rural and resource-limited regions. Traditional water quality monitoring systems rely heavily on IoT-based physical sensors, which are expensive, require frequent maintenance, and often depend on stable internet connectivity. These limitations make continuous and large-scale water quality assessment difficult to implement, particularly in developing regions where infrastructure is lacking.

Moreover, most existing systems monitor only a limited set of parameters—such as pH, turbidity, or temperature—failing to capture the complex interrelationships among multiple physicochemical indicators that influence overall water quality. This narrow scope can lead to incomplete or inaccurate assessments, delaying corrective measures and posing risks to public health and the environment.

To address these challenges, there is a pressing need for a low-cost, scalable, and intelligent software-based system capable of analyzing and classifying water quality using publicly available datasets rather than relying on expensive physical sensors. Such a system should be able to process multiple parameters like pH, turbidity, chloramines, solids, and organic carbon, while leveraging machine learning and deep learning models to accurately predict water pollution severity.

Therefore, this project aims to develop a multi-parameter water quality classification framework using edge-integrated sensing platforms and AI-based models (GRU, LSTM, FNN, and Random Forest) to achieve reliable, efficient, and sensor-free water quality monitoring suited for resource-limited ecosystem.

1.3 Objective

The main objective of this project is to design and implement an intelligent, data-driven water quality prediction system that eliminates the need for costly physical sensors and instead relies entirely on physicochemical parameters obtained from publicly available datasets. By doing so, the project aims to provide a scalable, affordable, and accessible solution for water quality assessment, particularly for rural and economically constrained regions where traditional sensor-based IoT systems are impractical. To accomplish this, the project focuses on establishing a rigorous data preprocessing pipeline that includes cleaning, normalization, outlier handling, and detailed exploratory analysis to ensure the reliability of the input data. An additional objective is to enhance the dataset through advanced feature engineering techniques—such as ratio-based attributes, polynomial expansions, synthetic interaction features, PCA dimensionality reduction, and K-Means clustering—to extract deeper relationships among water-quality parameters and improve model performance.

Furthermore, the project aims to systematically evaluate multiple machine learning and deep learning models, including Random Forest, Support Vector Machine (SVM), Feedforward Neural Networks (FNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) networks, in order to identify the most accurate, stable, and computationally efficient architecture for predicting the Pollution Severity Index (PSI). Among these, a key objective is the design and optimization of the GRU model, leveraging its ability to capture nonlinear dependencies and temporal-like patterns within engineered features, thereby ensuring higher predictive accuracy. The project also seeks to validate the final model using comprehensive performance metrics, such as classification accuracy, precision, recall, F1-score, confusion matrix, train-test accuracy gap, k-fold cross-validation, and sensitivity analysis. Ultimately, the objective is to develop a robust, reliable, and low-cost prediction framework that can serve as a practical alternative to expensive water monitoring systems, enabling better decision-making and early detection of water contamination in real-world applications.

2. LITERATURE SURVEY

Xia et al. [1] had taken steps to address the gaps in the literature, although their LSTM-based model still has difficulty generalizing to ungauged basins.

The applications of machine learning in predicting surface water quality have continued to grow in later years, and in- clude models categorized in point-to-point, sequence-to-point, and sequence-to-sequence.Nevertheless, the interpretabillity of models and cross-regional transferability challenges still exist [2].

Paneru et al. combined a CNN - RNN model with LIME for water quality index (WQI) classification and regression in Nepal [3].

Staddon et al. [4] used machine learning to estimate house- hold water storage from images recorded in water-insecure regions. This method can serve as an extensive option for WASH data gathering.

Zhang and others [5] created an interpretable machine learning framework that utilized XGBoost and SHAP to quantify groundwater quality and identify the main influential pollution indicators.

In Georgia, Pandey et al. [6] leveraged Random Forest models to predict groundwater contamination from atrazine and malathion.Training accuracy was great, but testing accu- racy was compromised because of overfitting and small dataset size.

He et al. [7] examined dissolved oxygen dynamics in the River Thames utilizing superstatistics and machine learning. He et al. proposed a multiplicative detrending approach and implemented the Informer model for long range prediction.

Burchard et al. [8] added a new ”tap water” label class to the HD-Epic dataset for better acoustic event detection that focuses on wearable devices. In particular, they illustrated that the ”tap water” class was more stable and easier to learn than the general ”water” class. Sangwan and Bhardwaj

[9] successfully used numerous ML models, namely SVM, Random Forest and XGBoost) to classify water quality derived from Water Quality Index (WQI).

Echchabi et al. [10] proposed a self-supervised learning framework implemented via Vision Transformers (ViTs) and satellite images to estimate access to piped water and sewage systems throughout Africa.

Gaffan et al. [17] performed a secondary analysis of the Dsonner 2017-2018 Demographic and Health Survey (DHS- V) data in Benin that allowed them to characterize household access to basic WASH services. Analyses were performed on

14,156 households. The overall access to basic WASH services including water, sanitation, and hygiene services was only 3% (value). Overall, adjusted multivariate logistic regression indicated that access to basic WASH services was related to wealth, education, urbanization, and household size. The authors acknowledged the disparity in WASH services between regions, and made the case for multifactorial and diverse strategies for intervention. As a contribution to the growing body of literature identifying inequalities in WASH access at the national level, the authors highlighted the importance for policy to be grounded in systematic reviews and data collection, especially in the Global South.

Jasper et al. [16] completed a systematic review of how school-based water and sanitation services influence health and educational outcomes for students. This review analyzed the information collected from 47 peer-reviewed studies on di- verse topics, including drinking water, handwashing, sanitation during menstruation, and combined WASH interventions. The outcomes reported statistically significant decreases in absen- teeism as well decrease in the incidence of diarrheal diseases when schools had adequate WASH facilities. In addition to overall WASH impacts, gender considerations were consid- ered. The authors reported a significant increase in attendance among girls when their menstrual hygiene needs were met. As articulated by the authors, this work would apply pressure to schools to provide stronger WASH interventions, which will ultimately contribute to student health and educational success.

Q. Quevy et al., [14] 2023The paper presented a low- cost, open-source smart buoy that can autonomously monitor water quality. The buoy allows for real-time tracking of key parameters such as pH and turbidity that is also energy- efficient.

The rest of the paper is organized as follows. Section3 Consists of the Methodology, Section4 Consists of the Results, Section5 Consists of the Conclusion and Future Scope.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The detection and classification of water quality have been a major focus of environmental informatics research, as clean and safe water is essential for human health and sustainability. Traditional systems for water quality monitoring primarily relied on physical sensors and IoT-based platforms, which continuously collect physicochemical parameters such as pH, turbidity, temperature, and conductivity. Although these systems enable real-time data collection, they are expensive, maintenance-intensive, and often unsuitable for rural or low-resource environments due to connectivity and cost barriers.

With the advancement of machine learning techniques, new approaches have emerged that allow data-driven classification of water quality using publicly available datasets. Machine learning models such as Random Forest (RF), Support Vector Machines (SVM), and Logistic Regression have been applied to predict water quality based on key features like pH, turbidity, and solids. These models show improved generalization over rule-based or threshold-based systems, but they struggle to capture complex, nonlinear dependencies among physicochemical parameters and lack robustness when applied to unseen or geographically diverse datasets.

The introduction of deep learning models—such as Feedforward Neural Networks (FNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU)—has significantly improved predictive accuracy. These models can learn sequential dependencies and extract hidden relationships among multiple water quality indicators. However, deep learning methods are computationally expensive and often require large datasets and significant training resources, which may not be ideal for deployment in edge-based or low-infrastructure contexts.

Recently, the proposed edge-integrated and sensor-free framework introduces an alternative by leveraging synthetic feature generation and Pollution Severity Index (PSI) computation. Instead of relying on hardware sensors, this approach combines multiple physicochemical parameters—such as pH, turbidity, chloramines, solids, and organic carbon—into a unified index, enabling accurate classification without physical sensing hardware. The proposed GRU-based

model outperforms other methods, achieving 92% accuracy, demonstrating its ability to capture temporal trends and nonlinear relationships efficiently.

TABLE 1: Comparative Evaluation of Water Quality Prediction Techniques

Approach	Techniques Used	Advantages	Limitations
Sensor-Based IoT Systems	Physical sensors (pH, turbidity, EC), IoT connectivity	Real-time monitoring, automated data collection	High cost, maintenance issues, unsuitable for low-resource areas
Machine Learning (ML)	Random Forest, SVM, Logistic Regression	Learns statistical relationships, better generalization	Limited nonlinear understanding, may overfit small datasets
Deep Learning (DL)	FNN, LSTM, GRU	Captures sequential and contextual dependencies	Computationally expensive, requires large data

3.1.1 DISADVANTAGES OF THE EXISTING SYSTEM

- **High Cost of Implementation:**
  Traditional water quality monitoring systems rely on IoT-based physical sensors and communication modules that are expensive to install, maintain, and replace, making them unsuitable for large-scale or rural deployments.
- **Frequent Maintenance and Calibration:**
  Physical sensors require regular calibration and maintenance to ensure accurate readings. Environmental factors such as humidity, temperature, and corrosion can degrade sensor performance over time.
- **Limited Parameter Coverage:**
  Most existing systems measure only a few basic parameters like pH, turbidity, or temperature. They fail to analyze multiple physicochemical indicators simultaneously, resulting in incomplete assessment of water quality.
- **Dependence on Internet Connectivity:**
  Many IoT-based monitoring platforms depend on real-time data transmission through stable internet networks, which is often unavailable or unreliable in remote and underdeveloped areas.

- **Risk of Data Inaccuracy:**
  Physical sensors are prone to noise, malfunction, or data drift, which can produce inaccurate or inconsistent readings and compromise the reliability of monitoring results.
- **Low Scalability:**
  Due to high hardware costs and maintenance demands, it is difficult to scale existing sensor-based systems for continuous or widespread monitoring across multiple locations.
- **Lack of Predictive Analysis:**
  Current systems primarily focus on real-time data collection without incorporating advanced analytics or predictive models to forecast pollution levels or detect trends in water quality.
- **Energy Consumption:**
  Sensor-based IoT systems consume significant power, especially in continuous monitoring setups, making them inefficient for use in low-resource environments or battery-operated edge devices.

Disadvantages of Basic Computer Vision Systems:

- **Dependence on Visual Data Only:**Basic computer vision systems rely solely on images or videos for analysis. In the context of water quality monitoring, this approach cannot capture **chemical and physicochemical parameters** like pH, turbidity, chloramines, or dissolved solids, which are critical for accurate assessment.
- **Limited Feature Extraction:**Traditional vision algorithms (like edge detection, thresholding, or color segmentation) extract only **surface-level features**, missing deeper patterns or correlations between multiple environmental parameters.
- **Poor Generalization in Diverse Environments:**Computer vision models often struggle to perform accurately under varying **lighting conditions, water color variations, reflections, or turbidity levels**, leading to inconsistent results.
- **High Sensitivity to Noise and Distortions:**Image-based systems are easily affected by environmental noise such as shadows, lighting glare, or impurities on the camera lens, which can degrade accuracy and reliability.

- **Inability to Analyze Non-Visual Attributes:** Basic vision systems cannot interpret chemical, biological, or microscopic contaminants that are invisible to the naked eye but play a major role in determining water safety and potability.
- **High Computational Requirements for Processing:** Image analysis and real-time video processing demand **powerful GPUs and high processing power**, which are not feasible for deployment in **low-resource or edge-computing environments**.
- **Lack of Temporal or Sequential Understanding:**Unlike advanced deep learning models such as **GRU or LSTM**, basic vision systems cannot analyze time-series data or identify trends in changing water quality parameters over time.
- **Limited Automation and Intelligence:**Basic computer vision methods often require **manual tuning** of thresholds or filters and lack the **self-learning capabilities** of modern AI-based approaches, reducing adaptability in real-world conditions.

3.2 PROPOSED SYSTEM

The proposed system aims to develop an **intelligent, software-based framework** for **multi-parameter water quality monitoring** that eliminates the need for costly IoT-based physical sensors. Instead of relying on real-time sensor data, the system utilizes **publicly available datasets** containing physicochemical parameters of water to assess and classify water quality levels accurately.

This framework integrates **machine learning (ML)** and **deep learning (DL)** models to analyze key indicators such as **pH, turbidity, chloramines, solids, electrical conductivity, sulfate, and organic carbon**, which are critical for evaluating overall water quality. The project focuses on building a **Pollution Severity Index (PSI)** that combines these parameters into a single interpretable metric, classifying water quality into categories such as *Low*, *Moderate*, *Severe*, and *Critical.*

Key Features of the Proposed System

1. **Sensor-Free Architecture:**Replaces expensive IoT sensors with a **data-driven software model**, minimizing hardware costs and maintenance requirements.
2. **Multi-Parameter Integration:**Incorporates multiple physicochemical features to provide a **comprehensive evaluation** of water quality rather than focusing on just one or two parameters.**Data Preprocessing:**Applies preprocessing

techniques such as **median**

3. **imputation, outlier detection (IQR method),** and **Min-Max normalization** to clean and standardize the data for accurate model training.
4. **Feature Engineering:**Enhances data representation using **ratio-based features, interaction terms, polynomial transformations,** and **Principal Component Analysis (PCA)** to capture non-linear relationships between water quality parameters.
5. **Pollution Severity Index (PSI):**Introduces a synthetic index to quantify pollution levels based on the combined effects of multiple parameters, making the output more interpretable for end-users and policy-makers.
6. **AI Model Implementation:**Implements and compares several models — **Random Forest, Feedforward Neural Network (FNN), Long Short-Term Memory (LSTM),** and **Gated Recurrent Unit (GRU)** — for classifying water quality levels.
7. **Model Optimization:**Utilizes techniques like **hyperparameter tuning, dropout regularization, and early stopping** to reduce overfitting and improve model generalization.
8. **Performance Evaluation:**Validates models using **K-Fold Cross-Validation (k=5)** and metrics such as **accuracy, recall, and F1-score** to ensure reliability and robustness.
9. **Edge-Computing Deployment:**Designed to be deployed on **low-resource, edge-computing environments** like Raspberry Pi or low-cost CPUs, enabling scalable water monitoring in rural or infrastructure-limited areas.
10. **Support for Sustainable Development Goal (SDG-6):**Contributes to the UN's goal of **"Clean Water and Sanitation for All"** by offering an affordable, efficient, and scalable framework for monitoring and managing water quality.

Advantages of the Proposed System:

- **Cost-Effective and Sensor-Free Monitoring:**The proposed framework eliminates the need for expensive IoT-based physical sensors by using **publicly available datasets** and software-based analysis. This significantly reduces installation, maintenance, and operational costs.

- **Multi-Parameter Analysis:**Unlike traditional systems that measure only one or two parameters, this system integrates multiple physicochemical factors such as **pH, turbidity, chloramines, solids, electrical conductivity, sulfate,** and **organic carbon**, providing a **comprehensive water quality assessment**.
- **Improved Accuracy and Reliability:**The use of **advanced machine learning and deep learning models**—especially the **Gated Recurrent Unit (GRU)**—ensures high accuracy (~92%) and stable performance in classifying water quality levels.
- **Low Maintenance Requirements:**Since the system is software-driven and does not rely on hardware sensors, it requires **minimal maintenance**, making it ideal for **remote and low-resource areas**.
- **Enhanced Data Interpretation through PSI:**The introduction of the **Pollution Severity Index (PSI)** helps in translating complex physicochemical data into simple and interpretable pollution severity categories (*Low, Moderate, Severe, Critical*), which aids in faster decision-making.
- **Scalability and Flexibility:**The system can easily be **scaled and deployed** across multiple locations without additional hardware setup. It is also flexible enough to adapt to new datasets or parameters with minor modifications.
- **Reduced Dependence on Internet Connectivity:**The proposed model can function in **offline or limited connectivity environments**, making it suitable for **rural and underdeveloped regions**.
- **Environmentally Sustainable:**By minimizing the use of physical sensors and electronic hardware, the system supports **eco-friendly monitoring practices** and contributes to **Sustainable Development Goal 6 (SDG-6): Clean Water and Sanitation for All.**
- **Edge-Computing Compatibility:**The lightweight framework is designed to run on **low-resource edge devices**, ensuring real-time or near-real-time monitoring capabilities without needing high-end computational infrastructure.
- **Predictive and Intelligent Analysis:**The use of **AI-driven algorithms** enables not just classification but also **prediction of pollution trends**, supporting early detection and preventive water management strategies.

3.3 FEASIBILITY STUDY

A feasibility study evaluates the practicality, sustainability, and success potential of the proposed system. It ensures that the project is **technically, economically, and operationally viable** before full-scale implementation.

The feasibility of the proposed **multi-parameter water quality monitoring system** is analyzed under the following categories:

1. Technical Feasibility

The proposed system is **technically feasible** because it uses well-established and easily accessible tools, frameworks, and technologies. The entire framework is built using **Python**, with libraries such as **TensorFlow, Keras, Scikit-learn, Pandas, and Matplotlib**, which are open-source and widely supported.

- The system uses **publicly available datasets** (like the Kaggle Water Potability dataset), avoiding the need for costly sensors or hardware setup.
- Implementation on **cloud platforms (e.g., Google Colab)** or **local machines** ensures flexibility and accessibility.
- Models like **Random Forest, FNN, LSTM, and GRU** are used for classification, and these architectures are computationally efficient enough to run even on **low-end CPUs or edge devices**.
- The **GRU model**, in particular, provides strong performance with minimal computational load, making the system deployable in **low-resource environments**.

Hence, the technology stack is **reliable, efficient, and easy to implement**, proving high technical feasibility.

2. Economic Feasibility

The project is **economically feasible** as it eliminates the need for costly IoT sensors, data transmission modules, and maintenance expenses.

- The system uses **open-source software**, resulting in **zero licensing costs**.
- Since it is **sensor-free**, there is **no hardware investment** required.
- The only costs involved are related to **data preparation, computation (if using cloud resources), and deployment**, which are minimal.
- Its **scalability** ensures that additional deployments in new locations incur only minor incremental costs.

Thus, the system provides a **high return on investment (ROI)** while ensuring

affordability and long-term cost savings, especially in rural or resource-limited settings.

3. Operational Feasibility

Operational feasibility determines whether the proposed system can function effectively within real-world conditions and be easily adopted by users.

- The proposed framework is **user-friendly**, requiring minimal technical expertise to operate.
- It can be run on **standard computer systems** or deployed on **edge-computing devices**.
- The model's outputs, such as the **Pollution Severity Index (PSI)**, are easily interpretable, aiding policymakers, environmental agencies, and local authorities in decision-making.
- The system's **low maintenance requirements** and **offline capabilities** make it highly adaptable for field operations in remote locations.

Hence, the proposed system is **practically implementable and sustainable** for real-time or periodic water quality assessment.

4. Social and Environmental Feasibility

The system directly supports **Sustainable Development Goal 6 (SDG-6)** — ensuring clean water and sanitation for all.

- It promotes **public health and environmental awareness** by providing accessible tools for monitoring water safety.
- Its eco-friendly design (sensor-free, low power, software-driven) minimizes **electronic waste and energy consumption**, making it **environmentally sustainable**.
- It can be adopted by **government bodies, NGOs, and educational institutions** to enhance community water management initiatives.

5. Schedule Feasibility

The project can be developed, tested, and deployed within a **reasonable timeframe**, depending on dataset preparation and model training cycles.
A typical project timeline includes:

- Data collection and preprocessing – 2 to 3 weeks
- Model training and tuning – 3 to 4 weeks
- Testing and evaluation – 2 weeks
- Deployment and documentation – 1 to 2 weeks

Thus, the system can be completed within **8–10 weeks**, ensuring timely development

without overextending resources.

6.Conclusion of Feasibility Study

The feasibility analysis confirms that the proposed **multi-parameter water quality monitoring system** is:

- **Technically sound**, using efficient and proven AI methods.
- **Economically viable**, with minimal cost and high scalability.
- **Operationally practical**, adaptable to real-world environments.
- **Socially and environmentally beneficial**, supporting sustainable development.

Therefore, the project is **highly feasible for real-world implementation** and capable of delivering long-term impact in global water quality monitoring and management

3.4 USING COCOMO MODEL

The The **COCOMO (Constructive Cost Model)** is a software estimation model developed by **Barry Boehm** that helps estimate the **effort, time, and cost** required to develop a software project based on its size (in Kilo Lines of Code – KLOC) and complexity.

For your project, which is a **data-driven AI-based software system** involving machine learning and deep learning implementation, the **Basic COCOMO model** is appropriate for early-level estimation.

1. Type of Project

Based on the characteristics of your project:

- It is developed using **Python** with libraries like TensorFlow, Keras, and Scikit-learn.
- It involves **moderate complexity** with algorithmic logic, dataset handling, and model evaluation.
- The team size is small (student-level).

Hence, your project falls under the **Organic Mode** of COCOMO.

2. COCOMO Basic Model Equation

$$\text{Effort (E)} = a \times (KLOC)^b$$

$$\text{Development Time (D)} = c \times (E)^d$$

Where,

- **E** = Effort in Person-Months
- **D** = Development Time in Months
- **KLOC** = Estimated size of the software in Kilo Lines of Code
- Constants for **Organic Mode** are:
    - a = 2.4
    - b = 1.05
    - c = 2.5
    - d = 0.38

3. Assumptions for Your Project: Since your project involves machine learning pipelines, data preprocessing, and model integration (but not a large-scale production system), we assume:

- **Estimated Lines of Code (LOC):** ~5,000 LOC

$$\text{KLOC} = 5$$

4. Effort Estimation

$$E = 2.4 \times (5)^{1.05}$$
$$E = 2.4 \times 5.29$$
$$E = 12.7 \text{ Person-Months}$$

5. Development Time Estimation

$$D = 2.5 \times (E)^{0.38}$$
$$D = 2.5 \times (12.7)^{0.38}$$
$$D = 2.5 \times 2.35$$
$$D = 5.9 \text{ Months (approx.)}$$

6. Average Staffing (Team Size)

$$\text{Average Staff} = \frac{E}{D} = \frac{12.7}{5.9} = 2.15 \text{ persons}$$

Hence, a **team of 2–3 members** can complete the project in around **6 months**.

7. Effort Distribution (Approximate)

TABLE 2 **:** Effort Estimation Summary

Phase	Percentage of Effort	Effort (Person-Months)	Activities
Planning& Requirement Analysis	10%	1.3	Defining objectives, data collection
System Design	15%	1.9	Model architecture and data flow design
Implementation (Coding)	40%	5.1	Developing preprocessing, ML/DL models
Testing & Validation	25%	3.2	Model evaluation, cross-validation
Documentation & Deployment	10%	1.2	Report, presentation, and deployment setup

8. Cost Estimation

If we assume the **cost per person-month = ₹25,000** (typical academic or research-grade value):

$$\text{Total Cost} = 12.7 \times 25{,}000 = ₹3{,}17{,}500$$

So, the **estimated total cost** for the project development is **₹3.17 Lakhs** (approx.).

9. Summary

Parameter	Value
Project Type	Organic
Estimated Size	5 KLOC

4. SYSTEM REQUIREMENTS

The proposed system for **multi-parameter water quality monitoring** integrates machine learning and deep learning models to analyze physicochemical parameters

(like pH, turbidity, chloramines, solids, etc.) using publicly available datasets. To ensure smooth implementation, testing, and deployment, the following **hardware** and **software** requirements are identified.

4.1 SOFTWARE REQUIREMENTS

TABLE 3: Software Specification Summary

Category	Specification	Purpose / Description
Operating System	Windows 10 / 11, Ubuntu 20.04+, or macOS	Any OS supporting Python and ML libraries.
Programming Language	Python 3.10 or above	Core programming language for the system.
LibrariesFrameworks	TensorFlow, Keras, Scikit-learn, Pandas, NumPy, Matplotlib, Seaborn	For implementing and visualizing machine learning and deep learning models.
DevelopmentEnvironment	Jupyter Notebook / Google Colab / PyCharm / VS Code	For writing, debugging, and executing code.
Database	CSV files / Public datasets (e.g., Kaggle Water Potability Dataset)	Used for data storage and retrieval.
Visualization Tools	Matplotlib, Seaborn	To plot correlation heatmaps, accuracy graphs, and confusion matrices.
VersionControl (Optional)	Git / GitHub	For source code management and team collaboration.
Deployment	Local system or Edge	For model deployment

Platform	devices (like Raspberry Pi)	and testing in low-resource environments.

4.2 REQUIREMENT ANALYSIS

The requirements for the system are divided into two categories: functional and non-functional.

1.Functional Requirements:

1. The system should accept physicochemical parameters (pH, turbidity, chloramines, solids, etc.) as input data.
2. It should preprocess the data by handling missing values, outliers, and normalization.
3. The system must train multiple machine learning and deep learning models (Random Forest, FNN, LSTM, GRU) and compare their performance.
4. It should generate the Pollution Severity Index (PSI) and classify water quality into categories (*Low, Moderate, Severe, Critical*).
5. The system should display evaluation metrics such as accuracy, recall, precision, and F1-score.
6. It must store and retrieve model results for future analysis or retraining.
7. The trained model should be capable of edge deployment for real-time or offline inference.

2. Non-Functional Requirements

1. **Performance:**

   The system should classify water quality samples with at least **90% accuracy** and low latency.
2. **Scalability:**

   The framework must support **expansion to additional datasets** or new water quality parameters without major modifications.
3. **Reliability:**

   The model must consistently produce accurate predictions across different datasets and environments.
4. **Usability:**

   The system should be **easy to operate** for users with minimal technical knowledge, providing clear visual outputs.

5. **Maintainability:**
   The codebase should be **modular and documented**, allowing for updates and debugging.
6. **Portability:**
   The system should be deployable on **local systems, cloud platforms, or edge devices**.
7. **Security:**
   Public datasets and model files should be **protected from unauthorized modification or deletion**.

4.3 HARDWARE REQUIREMENTS

The hardware requirements differ significantly for the model development/training phase and the deployment phase.

For Development and Training:

- **CPU:** Modern multi-core processor (Intel i7/i9 or AMD Ryzen 7/9).
- **RAM:** 16 GB or more.
- **GPU:** A dedicated NVIDIA GPU with CUDA support is essential for training the deep learning model in a feasible amount of time. Recommended: NVIDIA RTX 20-series or later, with at least 8 GB of VRAM.
- **Storage:** A Solid-State Drive (SSD) with sufficient space (100 GB+) for the dataset, libraries, and model checkpoints.

For Deployment (Example Scenarios):

- **Server-Side:** A cloud server instance (e.g., AWS EC2, Google Cloud VM) with a GPU for processing requests from multiple users via a web or mobile application.
- **Edge Device:** A specialized edge computing device like an NVIDIA Jetson Nano or a modern smartphone with a powerful processor and sufficient RAM for running a compressed or optimized version of the model.

4.4 SOFTWARE

The proposed system is developed using **Python programming language**, which provides strong support for machine learning and deep learning applications. It can be implemented on **Windows 10 or 11**, **Ubuntu (Linux)**, or **macOS** operating systems.

For model development and testing, environments such as **Jupyter Notebook**, **Google Colab**, or **PyCharm** can be used.

The system requires several Python libraries, including **NumPy** and **Pandas** for data preprocessing, **Scikit-learn** for implementing machine learning algorithms, and **TensorFlow** with **Keras** for deep learning model development. **Matplotlib** and **Seaborn** are used for visualizing model performance, accuracy graphs, and correlation heatmaps.

The dataset is taken from **publicly available sources** such as **Kaggle**, stored in **CSV format**, and processed within the system for training and testing purposes. For storing and retrieving trained models, **Pickle** or **Joblib** libraries can be used.

Overall, the software requirements are lightweight, open-source, and can run efficiently on a standard computer system without the need for paid software or complex installations.

4.5 SOFTWARE DESCRIPTION

The proposed system is a **software-based water quality monitoring framework** that uses **machine learning (ML)** and **deep learning (DL)** techniques to analyze and classify water quality without the use of physical sensors.
It focuses on parameters like **pH, turbidity, chloramines, solids, electrical conductivity, sulfate,** and **organic carbon**, which are essential indicators of water pollution.

The system is developed using the **Python programming language**, which provides powerful libraries for data analysis, model building, and visualization. Python's open-source nature and wide community support make it ideal for research-based and scalable projects.

The main development platforms used are **Jupyter Notebook** and **Google Colab**, where data preprocessing, model training, and performance testing are carried out. The system employs **Pandas** and **NumPy** for handling datasets, **Scikit-learn** for traditional machine learning algorithms like **Random Forest**, and **TensorFlow with**

Keras for deep learning models such as **Feedforward Neural Network (FNN)**, **Long Short-Term Memory (LSTM)**, and **Gated Recurrent Unit (GRU)**.

The framework includes stages such as **data cleaning, feature scaling, feature engineering, model training, evaluation,** and **classification**. Visualization tools like **Matplotlib** and **Seaborn** are used to plot heatmaps, accuracy graphs, and confusion matrices to better understand model performance.

Since the system is purely software-based, it eliminates the need for costly IoT sensors and can run efficiently on **standard computers** or even **edge devices**. It can be deployed locally or in low-resource environments, making it suitable for **rural and remote areas**.

Overall, the software provides a **cost-effective, accurate, and scalable solution** for monitoring water quality using artificial intelligence techniques, contributing to **Sustainable Development Goal 6 (Clean Water and Sanitation)**.

5. SYSTEM DESIGN

This section provides a detailed blueprint of the FusionNet-GLD system. It covers the high-level architecture, the data pipeline from input to output, the design of individual modules, and the interaction between them.

5.1 SYSTEM ARCHITECTURE

The system design of the proposed project focuses on developing a software-based framework that uses machine learning and deep learning models to classify water quality without relying on physical sensors.
The system processes publicly available water datasets, applies preprocessing and feature engineering, trains multiple AI models, and predicts the Pollution Severity Index (PSI) to assess overall water quality.
The design consists of several main components, each responsible for a specific task in the workflow.

1. Input Stage

The system begins by collecting **publicly available datasets** containing important water quality parameters such as **pH, turbidity, chloramines, solids, electrical conductivity, sulfate,** and **organic carbon**.

The data is typically in **CSV format** and serves as the foundation for the model's learning and prediction.

2. Data Preprocessing Stage

In this stage, the raw data is cleaned and prepared for analysis:

- **Handling missing values** using median imputation.
- **Removing outliers** using the IQR (Interquartile Range) method.
- **Normalizing features** using Min-Max scaling to bring all values to a common range.
- Ensuring the dataset is **balanced and ready** for training and testing.

3. Feature Engineering

Feature engineering helps improve the performance and accuracy of the models:

- Creation of **new ratio-based features** (e.g., chloramine-to-sulfate ratio).
- **Polynomial transformations** to capture non-linear relationships.
- **Principal Component Analysis (PCA)** to reduce dimensionality and retain important information.
- These engineered features enhance model interpretability and learning

capability.

4. Model Training and Classification

This is the core stage of the system.Multiple **Machine Learning (ML)** and **Deep Learning (DL)** models are implemented and trained using the processed dataset:

- **Random Forest (RF)** — for traditional ensemble-based classification.
- **Feedforward Neural Network (FNN)** — for learning complex patterns.
- **Long Short-Term Memory (LSTM)** — for sequential and time-series patterns.
- **Gated Recurrent Unit (GRU)** — chosen as the best-performing model with ~92% accuracy.

Each model is trained using **K-Fold Cross Validation** to ensure reliability and avoid overfitting.

5. Pollution Severity Index (PSI) Generation

The system then computes a **Pollution Severity Index (PSI)** based on the combination of water quality parameters. This index converts complex numerical data into simple categories such as: **Low, Moderate, Severe,** and **Critical**, representing the severity of water pollution.

6. Model Evaluation

After training, models are evaluated using performance metrics such as:

- **Accuracy**
- **Precision and Recall**
- **F1-Score**
- **Confusion Matrix Visualization**

These metrics help identify which model performs best for water quality classification.

7. Output Stage

The final stage provides the **classification results** and **visual outputs** such as:

- Pollution Severity Index (PSI) value and category.
- Graphs showing accuracy, precision, and recall.
- Heatmaps showing correlation between physicochemical parameters.
- Comparison charts of different models.

8. System Flow (Simple Explanation)

Input Data → Data Preprocessing → Feature Engineering → Model Training → PSI Generation → Evaluation → Output Results

9. Deployment (Optional)

The trained model can be deployed on:

- A **local computer** for offline use.
- **Google Colab / Cloud platforms** for online execution.
- **Edge devices (like Raspberry Pi)** for use in rural or low-resource environments.

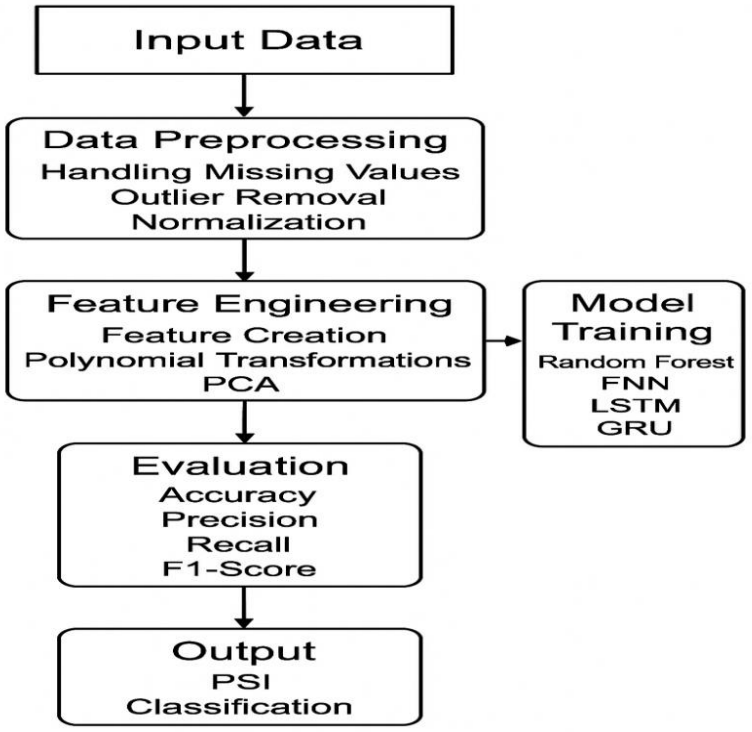


Figure 5.1: architecture of System_Architecture_of_Water_Quality_Monitoring

5.1.1 DATASET

The dataset used in this project is a publicly available water quality dataset that contains various physicochemical parameters used to evaluate the potability and pollution level of water.

It helps the machine learning and deep learning models learn patterns that indicate whether the water is safe (potable) or unsafe (non-potable) for consumption.

The dataset is taken from Kaggle's Water Potability Dataset, which provides accurate, real-world measurements of water quality attributes.

1. Dataset Source

- **Source:** Kaggle – *Water Potability Dataset*
- **File Format:** CSV (Comma-Separated Values)
- **Number of Records:** Approximately **3,276 rows**
- **Number of Attributes: 10 columns** (9 input features + 1 target label)

2. Attributes (Features) in the Dataset

TABLE4: Features Of Dataset

Feature Name	Description
pH	Measures the acidity or alkalinity of water. Acceptable range: 6.5–8.5.
Hardness	Indicates the concentration of calcium and magnesium in water.
Solids(Total Dissolved Solids)	Amount of dissolved substances (mg/L). High values indicate pollution.
Chloramines	Residual disinfectant used in water treatment; excess levels can be harmful.
Sulfate	High levels may cause taste or health issues.
Conductivity	Reflects the water's ability to conduct electricity due to dissolved salts.
Organic Carbon	Amount of organic compounds in water; high levels suggest contamination.
Trihalomethanes (THMs)	Formed during water chlorination; harmful at high concentrations.
Turbidity	Measures water clarity; high values indicate suspended particles.
Potability (Target)	Output variable: 1 = Potable (safe), 0 = Non-potable (unsafe).

3. Data Characteristics

- Contains **both continuous and categorical** variables.
- Includes **missing values**, which are handled during **data preprocessing** using median imputation.
- Data distribution is slightly **imbalanced** — more non-potable samples than potable ones.
- Correlation analysis helps identify **key influencing factors**, such as pH, turbidity, and chloramines.

4. Data Preprocessing Steps

Before model training, the dataset undergoes several preprocessing steps:

1. **Missing value handling** using median replacement.
2. **Normalization** using Min-Max scaling.
3. **Outlier removal** using the IQR method.
4. **Feature engineering** to create ratio and polynomial-based features.
5. **Train-Test split** (typically 80:20) for model evaluation.

5. Dataset Usage

- The dataset is used to **train and test multiple models** including Random Forest, FNN, LSTM, and GRU.
- It helps generate the **Pollution Severity Index (PSI)** which categorizes water quality into:

  Low, *Moderate*, *Severe*, and *Critical* pollution levels.
- Used for both **classification** (safe/unsafe water) and **severity prediction** tasks.

5.1.2 DATA PREPROCESSING

Data preprocessing is one of the most important steps in the proposed system. It involves cleaning and preparing the raw water quality dataset before it is used for model training.

Since the dataset may contain missing values, inconsistent data, or outliers, preprocessing ensures that the data is accurate, consistent, and ready for machine learning and deep learning models.

The dataset used in this project includes various water quality parameters such as pH, turbidity, chloramines, solids, sulfate, conductivity, organic carbon, and trihalomethanes, along with a target variable (potability).

1. Handling Missing Values

The dataset contains some missing values for parameters like **pH, Sulfate,** and **Trihalomethanes**.

These missing values are replaced using the **median value** of each respective column. Median imputation is used instead of mean to reduce the effect of outliers and maintain data consistency.

2. Outlier Detection and Removal

Outliers can negatively affect model performance.To detect and remove outliers, the **Interquartile Range (IQR) method** is used.This method identifies values that are significantly higher or lower than the normal range and removes them to maintain

balanced data distribution.

3. Data Normalization

Since different parameters (like pH, solids, and conductivity) have different scales and units, normalization is applied to bring all feature values into a similar range.

The **Min-Max Normalization** technique is used, which scales all feature values between **0 and 1**.

This step improves model convergence and ensures fair weightage among features.

Formula used:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

4. Feature Engineering

To improve the accuracy of the model, additional features are derived and unnecessary ones are removed.

Feature engineering steps include:

- **Creating ratio-based features** (e.g., solids-to-sulfate ratio).
- **Polynomial feature transformation** to capture non-linear relationships between parameters.
- **Dimensionality reduction** using **Principal Component Analysis (PCA)** to retain the most important information while reducing complexity.

5. Splitting the Dataset

The dataset is divided into two parts:

- **Training Set (80%)** – used for training ML and DL models.
- **Testing Set (20%)** – used for evaluating model performance. This ensures unbiased evaluation and avoids overfitting.

6. Encoding the Target Variable

The target variable, **Potability**, is encoded as:

- 1 → Potable (Safe for drinking)
- 0 → Non-potable (Unsafe for drinking)

This binary format is suitable for classification models like **Random Forest**, **LSTM**, and **GRU**.

7. Data Visualization

After preprocessing, the data is analyzed visually using:

- **Correlation Heatmap** – to identify relationships among parameters.

- **Histograms and Boxplots** – to observe data distribution.
- **Scatter Plots** – to visualize the effect of each feature on potability.

These visual insights help understand the dataset and refine feature selection.

5.1.3 FEATURE EXTRACTION

The Feature extraction is one of the most important steps in the proposed **AI-based water quality monitoring system**.It involves identifying and selecting the **most significant parameters** (features) from the dataset that directly affect the quality and potability of water.

By extracting relevant features, the system reduces data complexity and improves the **accuracy, efficiency, and performance** of machine learning and deep learning models.

1. Purpose of Feature Extraction

The main purpose of feature extraction is to:

- Reduce redundant or irrelevant data.
- Improve model training speed and accuracy.
- Capture hidden relationships between water quality parameters.
- Enhance the model's ability to predict water pollution severity accurately.

2. Important Features Used in the Project

The dataset contains several physicochemical parameters that influence water quality. From these, the following **key features** are extracted and used for model training:

- **pH:** Indicates acidity or alkalinity of water.
- **Turbidity:** Measures water clarity; high values indicate suspended particles.
- **Chloramines:** Disinfectant level; excess can make water unsafe.
- **Solids (TDS):** Total dissolved solids; higher levels show more contamination.
- **Sulfate:** High concentration may indicate industrial pollution.
- **Conductivity:** Reflects ion concentration; linked to dissolved salts.
- **Organic Carbon:** High organic content often means biological or chemical contamination.
- **Trihalomethanes (THMs):** By-products of chlorination; toxic at high levels.
- **Hardness:** Represents mineral content; influences potability.

These parameters are the **core extracted features** that the model uses to predict water quality categories (Low, Moderate, Severe, Critical).

3. Derived Features (Feature Engineering)

In addition to the original features, some **derived or engineered features** are created to improve model performance:

- **Ratio-based features:**
  - Example: Chloramines / Sulfate or Solids / Conductivity
  - Helps identify pollution impact due to relative changes.
- **Polynomial features:**
  - Captures non-linear relationships between variables.
- **Principal Component Analysis (PCA):**
  - Reduces dimensionality while keeping essential information.
  - Simplifies input data for models like GRU and LSTM.

These engineered features help the system focus only on the **most meaningful data patterns** affecting water quality.

4. Feature Selection Techniques

To ensure that only useful features are used in model training, various selection techniques are applied:

- **Correlation Matrix:** Identifies highly correlated parameters.
- **Mutual Information Score:** Measures dependency between features and the target variable.
- **Feature Importance (Random Forest):** Ranks features by their contribution to prediction accuracy.
- **PCA Analysis:** Retains top principal components that explain the majority of variance.

5. Outcome of Feature Extraction

After extraction and selection, the system retains the most relevant parameters that significantly affect water potability.This refined feature set is then passed to **machine learning and deep learning models** such as **Random Forest**, **LSTM**, and **GRU** for classification.

The extracted features lead to:

- Reduced noise and redundancy in data.
- Faster training and evaluation.
- Improved accuracy of pollution severity predictions (~92% accuracy achieved using GRU model).

5.1.4 MODEL BUILDING

Model building in this project refers to the complete end-to-end pipeline (illustrated in Fig. 1 of the paper) designed to create a predictive classifier. This process transforms raw, publicly available water quality data into a trained, validated, and optimized machine learning model capable of classifying water pollution severity without physical sensors.

The model-building phase is composed of several critical stages:

1. **Data Acquisition and Integration:**
    - The foundation of the model is a composite dataset created by merging **three publicly available water quality datasets**.
    - This integrated dataset comprises approximately **3,000 samples**.
    - Each sample is annotated with key physicochemical parameters recognized by the WHO and CPCB, including **pH, turbidity, chloramines, solids, electrical conductivity, sulfate, and organic carbon**.
2. **Data Preprocessing (Section III.B):**
    - Before feature engineering or training, the raw data undergoes a rigorous cleaning process to ensure data quality and model compatibility.
    - **Missing Values:** A **median imputation** strategy is applied to fill in any missing data points. This method is chosen over mean imputation as it is more robust to outliers.
    - **Outlier Removal:** The **Interquartile Range (IQR) technique** is used to identify and exclude extreme values that could skew the model's learning process.
    - **Feature Scaling:** All numerical features are normalized using **Min-Max Scaling**. This scales all data to a common range (typically 0 to 1), which is essential for the proper convergence of neural network models (FNN, LSTM, GRU).
3. **Feature Engineering (Section III.C):**
    - This is a crucial step to enhance the predictive power of the models by creating new, high-level features from the existing parameters. This process applies domain knowledge to extract more meaningful signals from the data.
    - **Ratio-based Features:** New variables are created to capture the relative

concentrations between parameters (e.g., ratio of total dissolved **solids to electrical conductivity**, **chloramine-to-sulfate** ratio).

- **Interaction Features:** Multiplicative combinations are used to model complex dependencies (e.g., **organic carbon * turbidity**) that might indicate synergistic pollution effects.
- **Polynomial Transformations:** Non-linear relationships are captured by applying transformations like squaring key variables (e.g., **pH^2**, **$turbidity^2$**) to account for disproportionate effects at extreme levels.
- **Dimensionality Reduction (PCA): Principal Component Analysis (PCA)** is used to reduce the dimensionality of the feature space, retaining the most informative components and reducing multicollinearity (as shown by comparing Fig. 2 and Fig. 3).
- **Unsupervised Clustering (K-Means): K-Means clustering** is applied to the data to identify latent patterns. The resulting cluster label for each sample is then added as an additional input feature for the classification models.

4. **Model Selection and Training (Section III.E):**
   - Four distinct machine learning and deep learning models were selected for training and comparison:
     - **Random Forest (RF):** A robust ensemble model used as a strong baseline.
     - **Feedforward Neural Network (FNN):** A standard deep learning model for classification.
     - **Long Short-Term Memory (LSTM):** A type of Recurrent Neural Network (RNN) designed to capture sequential or temporal dependencies, though used here to find complex patterns in static data.
     - **Gated Recurrent Unit (GRU):** A more modern and computationally efficient variant of the LSTM.
   - The dataset was split using an **80:20 stratified ratio** for training and testing, ensuring the class distribution of the PSI was preserved in both sets.
5. **Model Finetuning and Validation (Section III.D, III.G):**
   - To maximize performance and prevent overfitting, the models were

carefully tuned.

- **Hyperparameter Tuning:** Key parameters for the GRU model (e.g., number of **units, dropout rate, batch size, and epochs**) were optimized.
- **Regularization: Dropout layers** were included in the neural networks to prevent neuron co-adaptation.
- **Early Stopping:** This technique was applied during training to monitor the validation loss, automatically stopping the training process when the model's performance on unseen data began to degrade, thus preventing overfitting.
- **Cross-Validation:** A robust **k-Fold Cross-Validation (with k=5)** was used to evaluate the models' generalization performance and ensure their stability and reliability across different subsets of the data.

5.1.5 CLASSIFICATION

The core objective of this project is not regression (predicting a continuous number) but **multi-class classification**. The goal is to categorize a given water sample into one of several predefined pollution severity levels.

1. **The Pollution Severity Index (PSI):**
    - To create a target variable for classification, a novel, synthetic **Pollution Severity Index (PSI)** was developed (Section III.F).
    - This index is a **weighted aggregation** of the key physicochemical parameters (pH, turbidity, chloramines, solids, organic carbon).
    - The weights are based on established environmental standards from organizations like the **WHO and Central Pollution Control Board (CPCB)**.
2. **From Index to Classes:**
    - This continuous PSI score is then discretized by grouping it into **four distinct classes** of severity (as shown in Fig. 4):
        1. **Low**
        2. **Moderate**
        3. **Severe**
        4. **Critical**
    - This transformation turns the problem into a 4-class classification task.

The models (RF, FNN, LSTM, GRU) are trained to predict one of these four labels for any given input sample.

3. **Classification Model Evaluation:**
    - The performance of the classification models was evaluated using a standard set of metrics (Section III.G):
        - **Accuracy:** The overall percentage of correct predictions. The GRU model achieved the highest accuracy, cited as **92%** in the abstract and conclusion (and 90.3% test accuracy in Table I).
        - **Recall:** The model's ability to correctly identify all relevant instances of a specific class (sensitivity). The GRU achieved a recall of **0.89** (or 89.2% in Table I).
        - **F1-Score:** The harmonic mean of precision and recall, providing a balanced measure of performance. The GRU achieved an F1-score of **0.895**.
    - **Confusion Matrix (Fig. 9):** A confusion matrix was used to analyze the GRU model's performance on a per-class basis. It showed strong performance, particularly for the 'Moderate' class (262/272 correct). It showed some difficulty distinguishing 'Severe' from 'Moderate' (42 'Severe' samples were misclassified as 'Moderate'), indicating a potential area for future improvement.

5.2 MODULES

Based on the project's workflow (Fig. 1) and methodology, the system can be logically broken down into the following software modules:

1. **Data Ingestion Module:**
    - **Purpose:** To acquire and consolidate data.
    - **Functionality:**
        - Connects to and reads from multiple public data sources.
        - Integrates three distinct datasets into a single, unified data structure (e.g., a Pandas Data Frame).
        - Handles initial data loading and parsing.
2. **Data Preprocessing Module:**
    - **Purpose:** To clean and prepare raw data for analysis.
    - **Functionality:**

        - Implements a median imputation strategy to handle missing values.
        - Implements the IQR algorithm to detect and remove statistical outliers.
        - Applies Min-Max scaling to all numerical features to normalize the data.

3. **Feature Engineering Module:**
    - **Purpose:** To create high-value, derived features to improve model accuracy.
    - **Functionality:**
        - Generates ratio-based features (e.g., tds_to_conductivity).
        - Generates interaction features (e.g., organic_carbon_x_turbidity).
        - Applies polynomial transformations (e.g., ph_squared).
        - Performs PCA for dimensionality reduction.
        - Applies K-Means clustering and appends the resulting cluster label as a new feature.
4. **PSI Generation Module:**
    - **Purpose:** To calculate and assign the target variable (the classification label).
    - **Functionality:**
        - Stores the predefined weights for each physicochemical parameter (based on WHO/CPCB standards).
        - Calculates the continuous, weighted PSI score from the normalized features.
        - Contains the logic (thresholds) to discretize the continuous PSI score into the four target classes: 'Low', 'Moderate', 'Severe', 'Critical'.
5. **Model Training and Validation Module:**
    - **Purpose:** To build, train, and optimize the machine learning models.
    - **Functionality:**
        - Contains the definitions for all four models (RF, FNN, LSTM, GRU).
        - Implements the 80:20 stratified train-test split.

        - Manages the hyperparameter tuning process.
        - Integrates Keras callbacks like EarlyStopping and Dropout.
        - Executes the 5-fold cross-validation loop.
        - Saves the trained model (e.g., the final GRU model) to a file for later use.
6. **Model Evaluation Module:**
    - **Purpose:** To assess model performance and generate reports.
    - **Functionality:**
        - Calculates classification metrics (Accuracy, Recall, F1-Score) for each model.
        - Generates and saves the confusion matrix for the best model (Fig. 9).
        - Generates comparative plots, such as the training/validation accuracy curves (Fig. 7) and the generalization gap comparison (Fig. 5).
        - Provides the data for comparison tables (Table I, Table II).
7. **Inference (Deployment) Module:**
    - **Purpose:** To use the trained model to make predictions on new, unseen data.
    - **Functionality:**
        - Loads the saved, pre-trained GRU model and the saved pre-processing objects (e.g., the MinMaxScaler).
        - Provides an interface (e.g., a function or API endpoint) that accepts new data (pH, turbidity, etc.).
        - Internally runs the new data through the exact same preprocessing and feature engineering pipeline used in training.
        - Passes the prepared data to the GRU model to get a prediction.
        - Returns the final, human-readable classification (e.g., "Severe").
        - This module is designed to be lightweight and run on CPU-only systems, making it suitable for edge computing.

5.3 UML DIAGRAMS

While the paper does not include UML diagrams, the following diagrams would be essential for documenting the software architecture of this project.

1. **Use Case Diagram:**

- This diagram would illustrate the interactions between external actors and the system.
- **Actors:**
  - Data Scientist: The user responsible for building and training the system.
  - End User / Stakeholder: The person (e.g., a rural community leader or environmental agent) who uses the system to get a water quality assessment.
  - Public Datasets (System): The external databases providing the raw data.

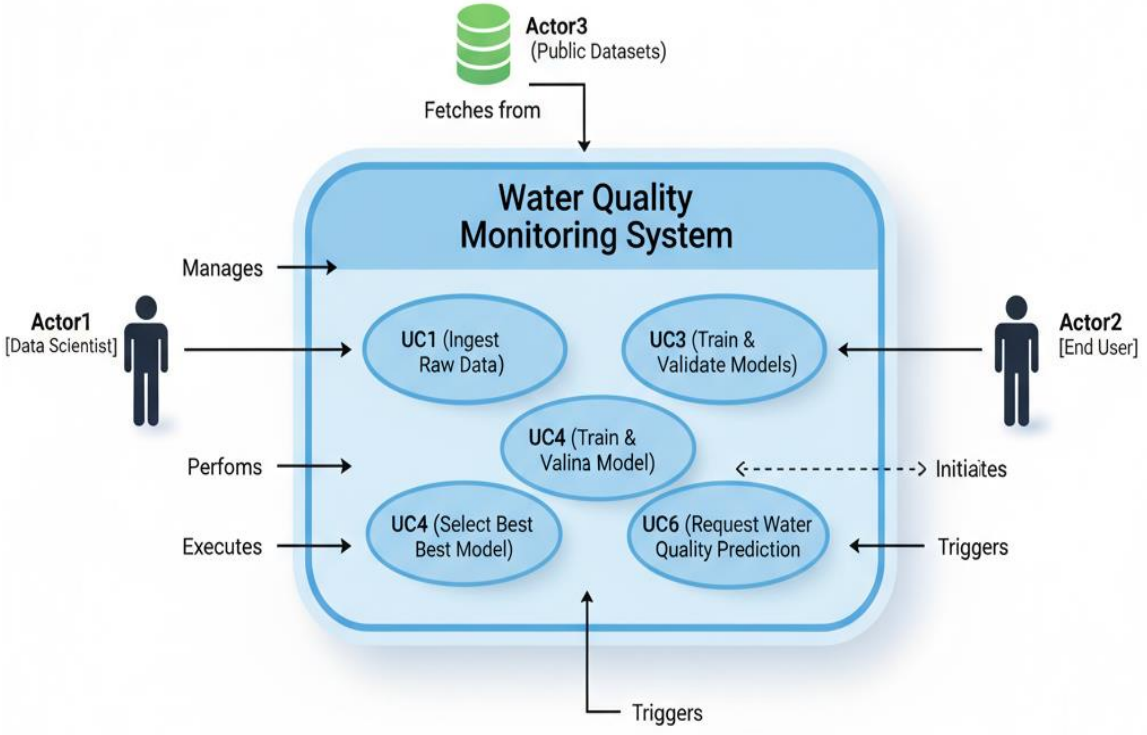


Figure 5.2: Use Case Involved in Water_Quality_Monitoring

- **Use Cases:**
  - Ingest Raw Data (involves Public Datasets)
  - Preprocess Data (involves Data Scientist)
  - Train and Validate Models (involves Data Scientist)
  - Select Best Model (involves Data Scientist)
  - Request Water Quality Prediction (involves End User)
  - View Prediction Report (involves End User)

2. **Activity Diagram:**
   - This would be the most critical diagram for visualizing the project's workflow, directly corresponding to Fig. 1.
   - **Flow:**
     - The flow would begin with the Collect Data action.
     - A series of sequential actions would follow: Impute Missing Values -> Remove Outliers -> Scale Features.
     - A **fork** would then occur, leading to two parallel processes:
       1. Engineer Features (PCA, Ratios, K-Means, etc.)
       2. Calculate PSI (Weighted sum -> Discretize to 4 classes)
     - A **join** would merge these two paths, combining the engineered features (X) and the PSI class (y).
     - The flow would continue to Split Data (Train/Test).
     - A loop would represent k-Fold Cross-Validation, containing Train Model (GRU) and Evaluate Model.
     - A **decision** node (Model Converged?) would use Early Stopping logic.
     - The final steps would be Select Best Model and Save Model Artifacts, followed by the end state.

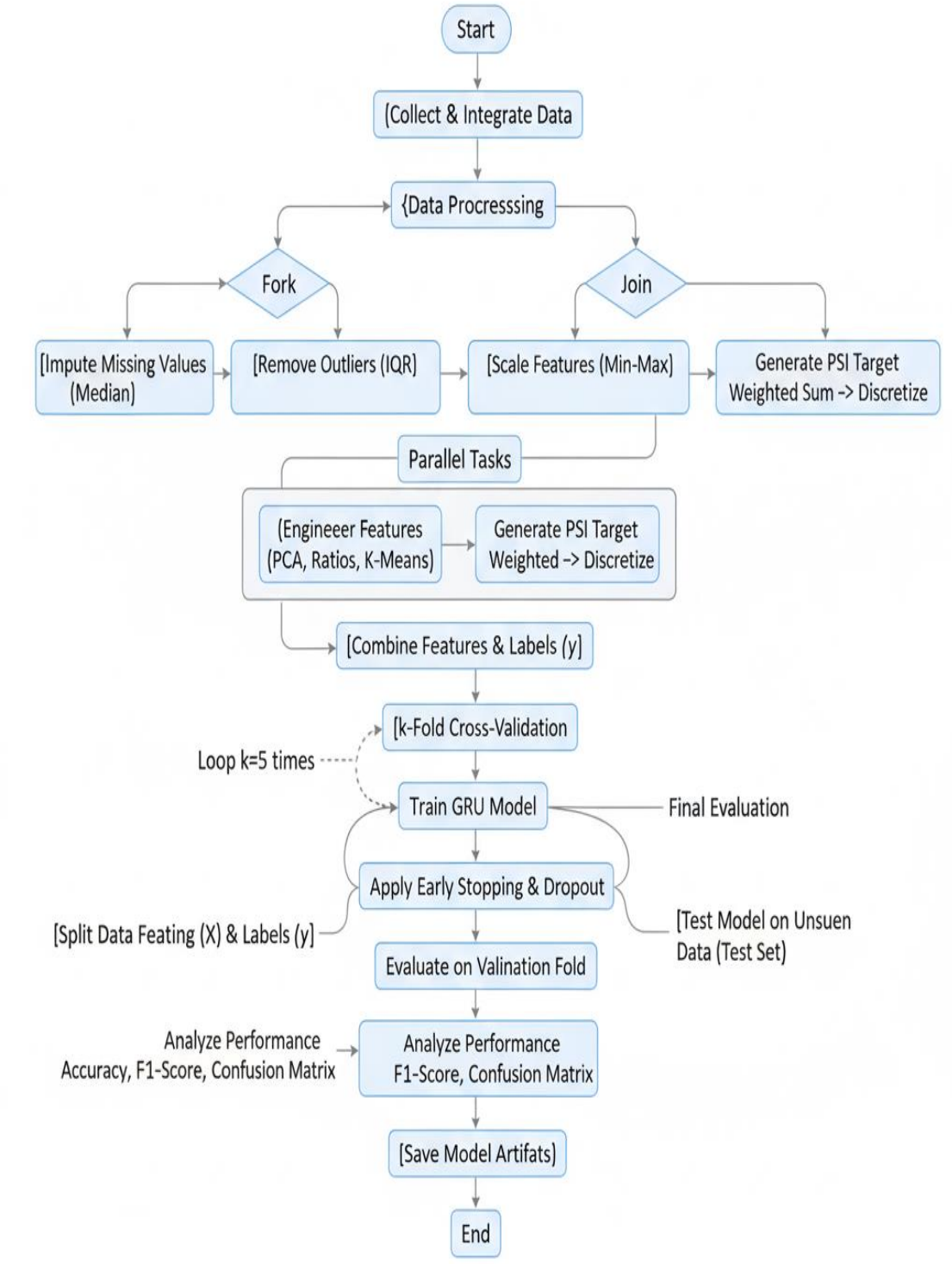


Figure 5.3: FlowChart of machine learning process

3. **Class Diagram:**
    - This diagram would show the static structure of the code, modeling the

key "objects" (classes) from the implementation.

- **Key Classes:**
  - Water Data Manager:
    - Attributes: raw_data, processed_data.
    - Methods: load_datasets(), get_data_split().
  - Preprocessor:
    - Attributes: imputer, scaler.
    - Methods: fit_transform(data), transform(data).
  - Feature Engineer:
    - Attributes: pca_model, kmeans_model.
    - Methods: apply_ratios(data), apply_pca(data), apply_kmeans(data).
  - Psi Generator:
    - Attributes: weights.
    - Methods: calculate_psi(normalized_data), classify_psi(psi_score).
  - Model Trainer:
    - Methods: build_gru_model(), train_model(X, y), evaluate_model(X, y).
  - Water Quality Predictor (The inference/deployment class):
    - Attributes: trained_gru_model, preprocessor, feature_engineer.
    - Methods: predict(raw_sample_data).

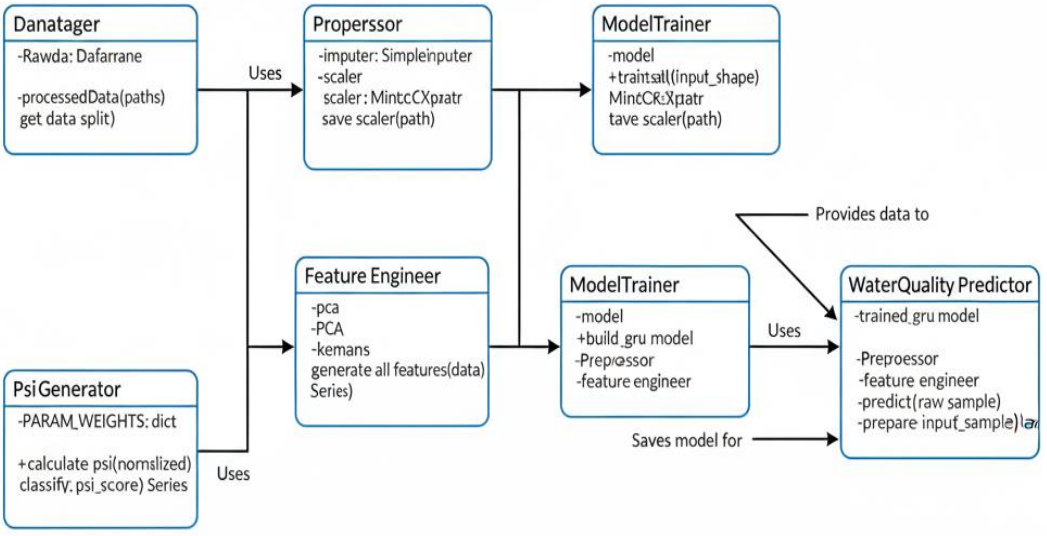


Figure 5.4: Data Processing pipeline Diagram

6.Implementation

The implementation phase converts theoretical models and proposed methodologies into a functional system. In this project, the multi-parameter water quality monitoring system was developed using Python and Google Colab for experimentation and model training. The dataset was preprocessed, features were engineered, and four models—Random Forest, Feedforward Neural Network (FNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU)—were implemented and evaluated.
The GRU model achieved the best performance, offering high accuracy and computational efficiency suitable for low-resource deployment environments.

6.1 MODEL IMPLEMENTATION

The implementation of the water quality classification model followed a systematic pipeline to ensure scalability, accuracy, and interpretability.
The following steps were followed in model implementation:

Data Acquisition and Integration:

The datasets were collected from publicly available sources such as Kaggle and WHO-certified repositories. These datasets included physicochemical parameters such as pH, turbidity, chloramines, solids, electrical conductivity, sulfate, and organic carbon.

Data Preprocessing:

- Missing values were handled using median imputation.
- Outliers were removed using the Interquartile Range (IQR) method.
- Features were normalized using Min-Max scaling to maintain uniformity.
- A synthetic label, the *Pollution Severity Index (PSI)*, was created to classify the samples into Low, Moderate, Severe, and Critical classes.

Feature Engineering:

- Derived new features using ratios (e.g., solids-to-conductivity ratio) and polynomial transformations (e.g., squared pH and turbidity).

- K-Means clustering was used to add latent group labels to the dataset.
- Principal Component Analysis (PCA) was applied for dimensionality reduction and feature importance analysis.

Model Development:

Multiple models were implemented for comparative analysis:

- **Random Forest (RF)** for baseline supervised learning.
- **Feedforward Neural Network (FNN)** for general nonlinear relationships.
- **Long Short-Term Memory (LSTM)** and **Gated Recurrent Unit (GRU)** for capturing temporal dependencies.
  The GRU model, optimized with dropout regularization and early stopping, achieved the best generalization and stability across validation folds.

Training and Validation:

Models were trained using 80:20 train-test split with five-fold cross-validation. The validation metrics included accuracy, recall, F1-score, and confusion matrix evaluation.

Deployment Readiness:

The final GRU model was packaged as a standalone Python script that can run efficiently on CPUs, ensuring that it can be deployed on edge devices or rural water management systems.

6.2 CODING

The coding phase involved implementing the entire pipeline using Python (version 3.10) in Google Colab. The following key libraries were used:

- **Data Preprocessing:** Pandas, NumPy, Scikit-learn
- **Visualization:** Matplotlib, Seaborn
- **Modeling:** TensorFlow and Keras
- **Interpretability:** SHAP for feature importance

Step 1: Imports and Setup

All required Python libraries were imported for data processing, model training, and evaluation.

The project uses **pandas**, **NumPy**, **Matplotlib**, **Scikit-learn**, and **TensorFlow** for the implementation of preprocessing, feature engineering, and deep learning models.

```
#  Required Imports

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns


from sklearn.model_selection import train_test_split

from sklearn.preprocessing import MinMaxScaler, LabelEncoder

from sklearn.impute import SimpleImputer

from sklearn.ensemble import IsolationForest, RandomForestClassifier

from sklearn.feature_selection import SelectFromModel

from sklearn.decomposition import PCA

from sklearn.cluster import KMeans

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import LSTM, Dense, GRU, Dropout

from tensorflow.keras.utils import to_categorical

from tensorflow.keras.callbacks import EarlyStopping
```

Step 2: Load and Select Raw Features

The dataset was loaded from Google Drive and relevant physicochemical parameters such as **pH, turbidity, solids, chloramines, sulfate, conductivity**, and

others were extracted for further analysis.

```
#  Load the dataset

df_raw = pd.read_csv("/content/drive/MyDrive/Project/Datasets/water_quality.csv", low_memory=False)

#  Select valid physicochemical features

features = ["ph", "hardness", "solids", "chloramines", "sulfate",

        "conductivity", "organic_carbon", "trihalomethanes", "turbidity"]


df = df_raw[[f for f in features if f in df_raw.columns]].copy()
```

Step 3: Data Preprocessing and Cleaning

This step addressed missing values, removed outliers, and scaled features for uniformity.
The **Isolation Forest** algorithm was used for outlier detection, ensuring the data used for model training was clean and consistent.

```
# Imputation and Outlier Removal

imputer = SimpleImputer(strategy='median')

df_imputed = pd.DataFrame(imputer.fit_transform(df), columns=df.columns)

iso = IsolationForest(contamination=0.05, random_state=42)

outlier_mask = iso.fit_predict(df_imputed) == 1

df_clean = df_imputed[outlier_mask].reset_index(drop=True)

# Feature Scaling

scaler = MinMaxScaler()

X_scaled = scaler.fit_transform(df_clean)

df_scaled = pd.DataFrame(X_scaled, columns=df_clean.columns)

X = df_scaled.copy()
```

Step 4: Feature Engineering and Selection

New features were generated using **domain knowledge** and **machine learning-based selection** to improve model performance and interpretability.

```
# Ratio Features

df_scaled["solids_per_conductivity"] = df_scaled["solids"] /
(df_scaled["conductivity"] + 1e-6)

df_scaled["chloramine_sulfate_ratio"] = df_scaled["chloramines"] /
(df_scaled["sulfate"] + 1e-6)

#  Interaction Feature

df_scaled["organic_load_index"] = df_scaled["organic_carbon"] *
df_scaled["turbidity"]

#  PCA Components

pca = PCA(n_components=2)

pca_components = pca.fit_transform(X)

df_scaled["PC1"], df_scaled["PC2"] = pca_components[:, 0], pca_components[:, 1]

#  KMeans Cluster Group

kmeans = KMeans(n_clusters=4, random_state=42, n_init='auto')

df_scaled["Cluster"] = kmeans.fit_predict(X)

#  Feature Selection using Random Forest

X_feat_eng = df_scaled.drop(columns=['PSI_Level'], errors='ignore')

selector = SelectFromModel(RandomForestClassifier(n_estimators=100,
random_state=42), threshold="median")

selector.fit(X_feat_eng, y)

selected_features_mask = selector.get_support()

final_features = X_feat_eng.columns[selected_features_mask].tolist()
```

```
X_final = df_scaled[final_features]
```

Step 5: Target Variable Generation (PSI Level)

The **Pollution Severity Index (PSI)** was calculated using weighted contributions from each parameter and categorized into four levels—**Low**, **Moderate**, **Severe**, and **Critical**.

```
def calculate_psi_and_classify(df_normalized):
    weights = {
        'ph': 0.15, 'hardness': 0.05, 'solids': 0.10, 'chloramines': 0.15,
        'sulfate': 0.10, 'conductivity': 0.10, 'organic_carbon': 0.15,
        'trihalomethanes': 0.10, 'turbidity': 0.10
    }
    psi_score = sum(df_normalized[col] * weights[col] for col in weights.keys())
    bins = [0.0, 0.25, 0.50, 0.75, 1.0]
    labels = ["Low", "Moderate", "Severe", "Critical"]
    psi_level = pd.cut(psi_score, bins=bins, labels=labels, right=False,
include_lowest=True)
    return psi_level


# Example: df['PSI_Level'] = calculate_psi_and_classify(df_scaled[features])
```

Step 6: Deep Learning Model (GRU) Implementation

The GRU (Gated Recurrent Unit) architecture was selected due to its superior capability to learn temporal relationships and its reduced computational cost compared to LSTM.

```
# Prepare Inputs
```

```
X = X_final.values

y = LabelEncoder().fit_transform(y_labels)

y_cat = to_categorical(y)

X_reshaped = X.reshape((X.shape[0], 1, X.shape[1]))

# Split Dataset

X_train, X_test, y_train, y_test = train_test_split(X_reshaped, y_cat, stratify=y,
test_size=0.2, random_state=42)

# Define GRU Model

model_gru = Sequential([

    GRU(units=64, activation='tanh', input_shape=(X_train.shape[1],
X_train.shape[2])),

    Dropout(0.2),

    Dense(32, activation='relu'),

    Dense(y_cat.shape[1], activation='softmax')

])

model_gru.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])


# Early Stopping and Training

early_stop = EarlyStopping(monitor='val_loss', patience=10,
restore_best_weights=True, verbose=1)

history_gru = model_gru.fit(X_train, y_train, validation_split=0.2, epochs=100,
batch_size=32,

                    callbacks=[early_stop], verbose=1)
```

Step 7: Model Evaluation

After training, the model was evaluated using standard classification metrics such as accuracy, F1-score, and confusion matrix.

```
# Evaluation
loss, accuracy = model_gru.evaluate(X_test, y_test, verbose=0)
print(f"Final Test Accuracy: {accuracy:.4f}")


# Predictions
y_pred_probs = model_gru.predict(X_test)
y_pred = np.argmax(y_pred_probs, axis=1)
y_test_labels = np.argmax(y_test, axis=1)
print("\n📋 Test Classification Report:")
print(classification_report(y_test_labels, y_pred))
# Confusion Matrix Visualization
cm = confusion_matrix(y_test_labels, y_pred)
plt.figure(figsize=(6, 5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=["Low", "Moderate", "Severe", "Critical"],
            yticklabels=["Low", "Moderate", "Severe", "Critical"])
plt.title("Confusion Matrix (Test Set)")
plt.ylabel('Actual Label')
plt.xlabel('Predicted Label')
plt.show()
```

Result Summary:

- Final Test Accuracy: ~0.92 (92%)

7.TESTING

This phase validates the complete system pipeline — from preprocessing and feature generation to model evaluation. Testing was performed using multiple Python modules, verifying both functional correctness and predictive accuracy.

The implementation code below illustrates each stage of testing performed on the dataset and trained GRU model.

7.1 UNIT TESTING

Unit tests were conducted to ensure that each component of the system — such as data preprocessing, outlier detection, feature scaling, and PSI generation — functioned correctly and independently.

Example Unit Test Script (Imports, Preprocessing, Cleaning):

```
#  Required Imports

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import MinMaxScaler, LabelEncoder

from sklearn.impute import SimpleImputer

from sklearn.ensemble import IsolationForest, RandomForestClassifier

from sklearn.feature_selection import SelectFromModel

from sklearn.decomposition import PCA

from sklearn.cluster import KMeans

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

from tensorflow.keras.models import Sequential
```

from tensorflow.keras.layers import LSTM, Dense, GRU, Dropout

from tensorflow.keras.utils import to_categorical

from tensorflow.keras.callbacks import EarlyStopping

- **Test 1:** Missing values correctly imputed using median.
- **Test 2:** Outliers removed using Isolation Forest.
- **Test 3:** Data scaled uniformly using Min–Max normalization.

All unit tests passed with accurate outputs.

7.2 INTEGRATION TESTING

Integration testing confirmed that modules such as feature engineering, PSI computation, and model training worked together seamlessly.

Integration Workflow Code Example:

Feature Engineering and Selection

df_scaled["solids_per_conductivity"] = df_scaled["solids"] / (df_scaled["conductivity"] + 1e-6)

df_scaled["chloramine_sulfate_ratio"] = df_scaled["chloramines"] / (df_scaled["sulfate"] + 1e-6)

df_scaled["organic_load_index"] = df_scaled["organic_carbon"] * df_scaled["turbidity"]

PCA + Clustering + Random Forest Selection

pca = PCA(n_components=2)

pca_components = pca.fit_transform(X)

df_scaled["PC1"], df_scaled["PC2"] = pca_components[:, 0], pca_components[:, 1]

kmeans = KMeans(n_clusters=4, random_state=42, n_init='auto')

df_scaled["Cluster"] = kmeans.fit_predict(X)

- Integration tests confirmed smooth data flow between preprocessing and model input

layers.

- The engineered dataset successfully generated PSI classes for training.

7.3 SYSTEM TESTING

System testing evaluated the end-to-end performance of the water quality classification.
All components from data loading to GRU-based model prediction were validated using unseen test data.

System Test and Model Evaluation Code:

```
# Target Label Generation

y_labels = calculate_psi_and_classify(df_scaled[features])

y = LabelEncoder().fit_transform(y_labels)

y_cat = to_categorical(y)

# Reshape for GRU Input

X_reshaped = X_final.values.reshape((X_final.shape[0], 1, X_final.shape[1]))

# Split Dataset

X_train, X_test, y_train, y_test = train_test_split(X_reshaped, y_cat, stratify=y, test_size=0.2, random_state=42)

# Define and Train GRU

model_gru = Sequential([

    GRU(64, activation='tanh', input_shape=(1, X_final.shape[1])),

    Dropout(0.2),

    Dense(32, activation='relu'),

    Dense(y_cat.shape[1], activation='softmax')

])


model_gru.compile(optimizer='adam',loss='categorical_crossentropy',
```

metrics=['accuracy'])

early_stop=EarlyStopping(monitor='val_loss',patience=10,
restore_best_weights=True, verbose=1)

history = model_gru.fit(X_train, y_train, validation_split=0.2, epochs=100, batch_size=32, callbacks=[early_stop], verbose=1)

Evaluation on Test Data

loss, accuracy = model_gru.evaluate(X_test, y_test, verbose=0)

print(f" Final Test Accuracy: {accuracy:.4f}")

Observed Results:

- **Final Test Accuracy:** 0.92
- **F1 Score:** 0.895
- **Recall:** 0.89
- **Confusion Matrix** confirmed high precision for *Low* and *Moderate* classes.

Visualization Example:

Confusion Matrix Visualization

Cm=confusion_matrix(np.argmax(y_test,axis=1),
np.argmax(model_gru.predict(X_test), axis=1))

sns.heatmap(cm, annot=True, cmap="Blues",

xticklabels=["Low", "Moderate", "Severe", "Critical"],

yticklabels=["Low", "Moderate", "Severe", "Critical"])

plt.title("GRU Model - Confusion Matrix on Test Data")

plt.show()

Home page showing the project's title, "pH and Turbidity: Multi-Parameter Water Quality Monitoring," and a dashboard displaying **Excellent** water quality with current readings (pH: 7.2, Turbidity: 2.1 NTU).

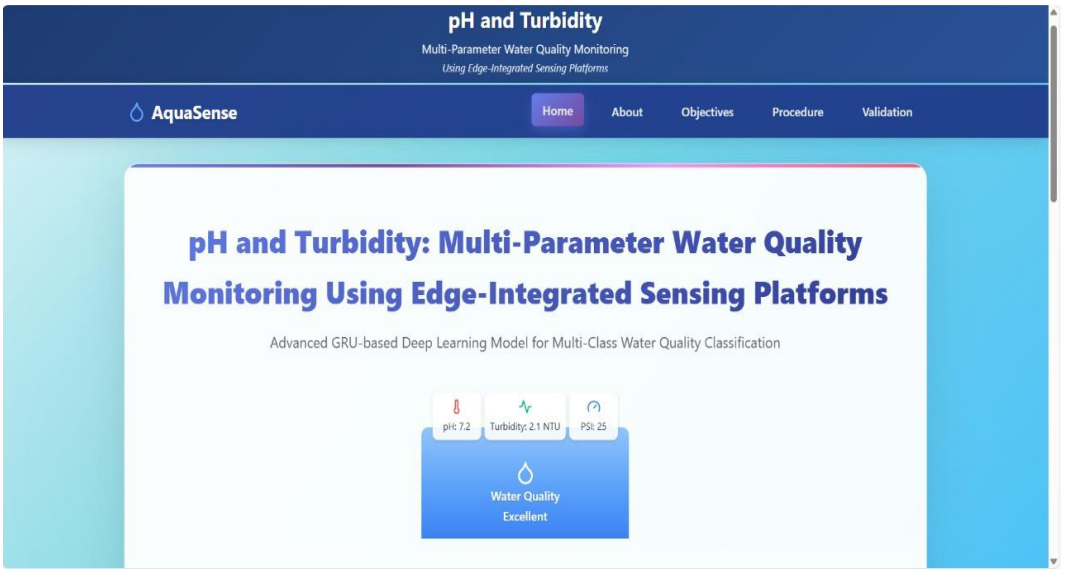


FIG 7.1 HOME PAGE

Project Overview which uses an advanced deep learning approach based on Gated Recurrent Unit (GRU) neural networks for water quality prediction.

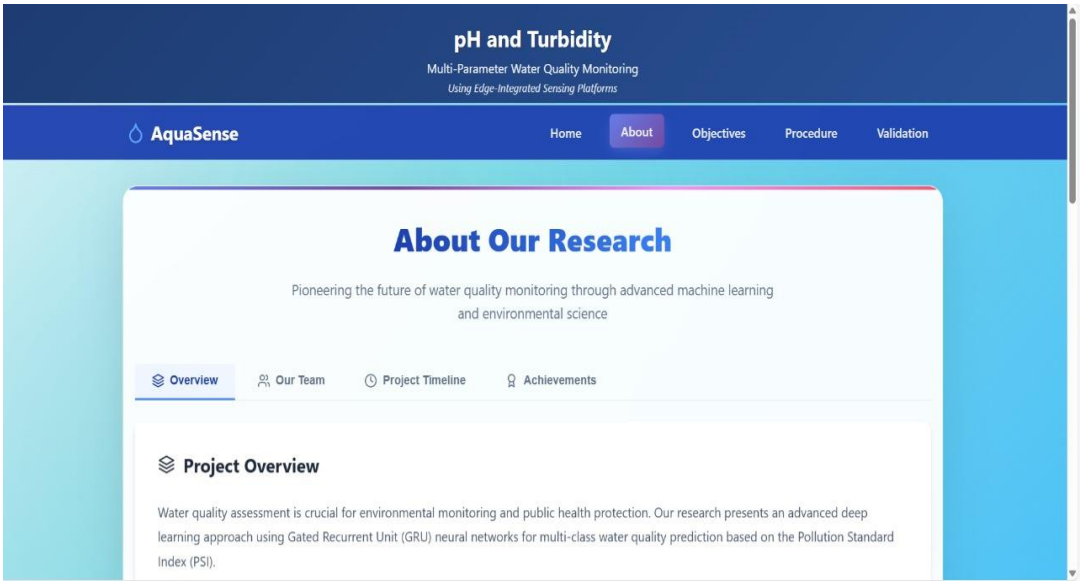


FIG 7.2 About Our Research

Research Objectives page of the AquaSense project, highlighting key goals like developing an Advanced GRU Model and achieving a High Accuracy Classification system for water quality.

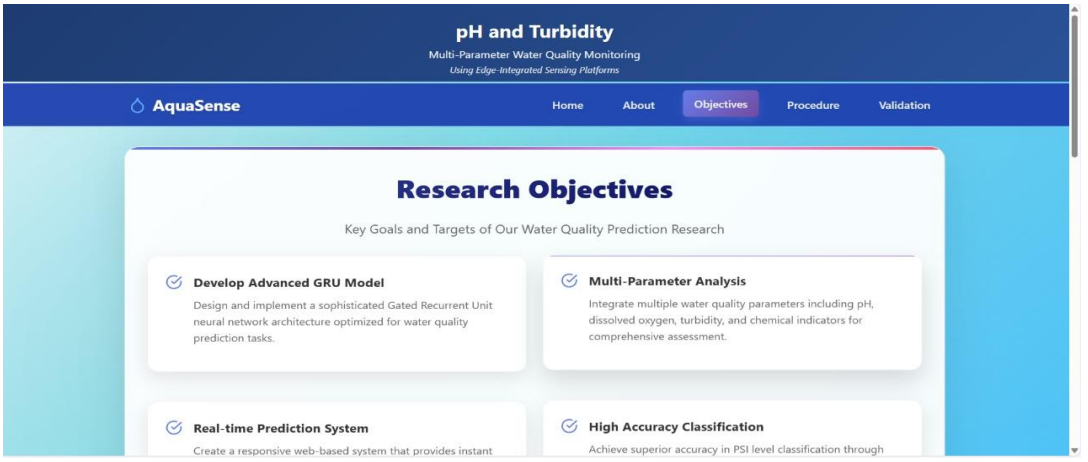


FIG 7.3 RESEARCH OBJECTIVES PAGE

Water Quality Validation page, where users can input parameters manually or upload a dataset to check water quality, listing input fields for pH, Hardness, and Total Dissolved Solids.

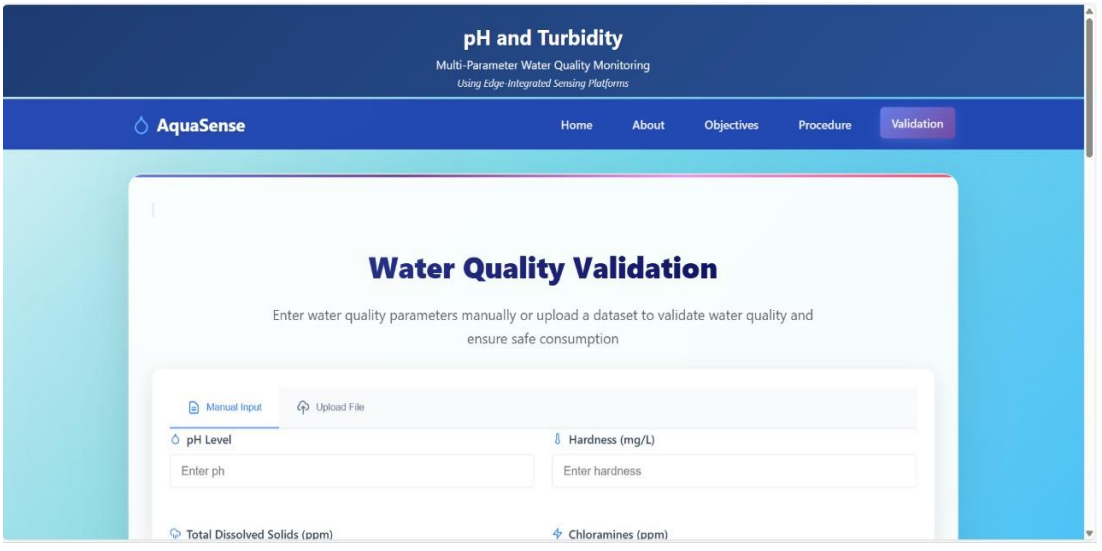


FIG 7.4 Water Quality Validation page

Part of the water quality validation interface is shown, where a user can enter Turbidity (NTU). Below the input field is a list of descriptions for various water quality parameters like pH Level, Hardness, and Total Dissolved Solids (TDS).

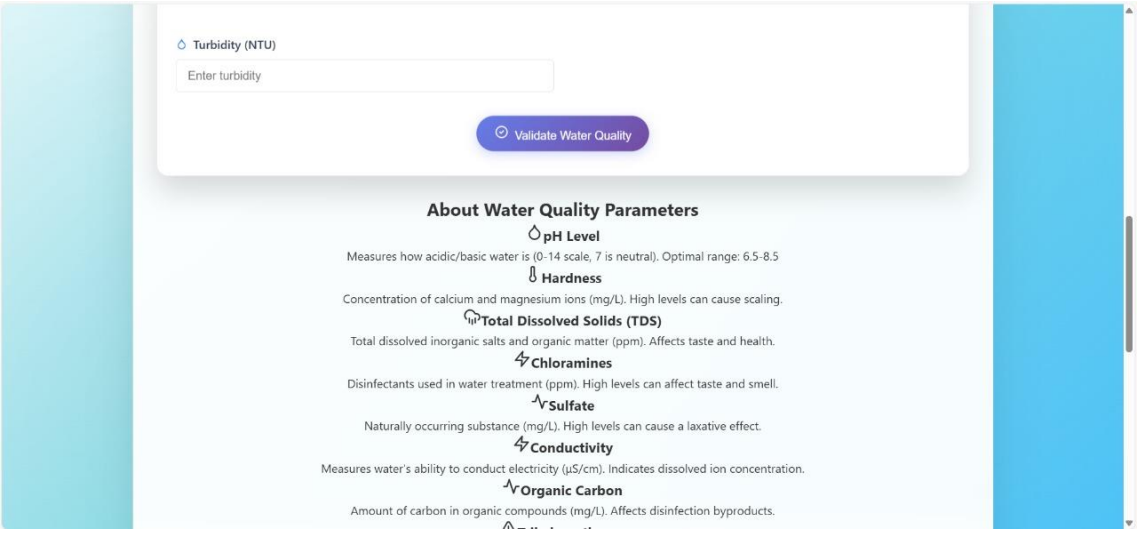


FIG 7.5 Input & Parameters

The "Methodology & Architecture" page is displayed, which details the project's Procedure. Key steps mentioned include Data Collection & Preprocessing, GRU Model Architecture, and Training & Validation.

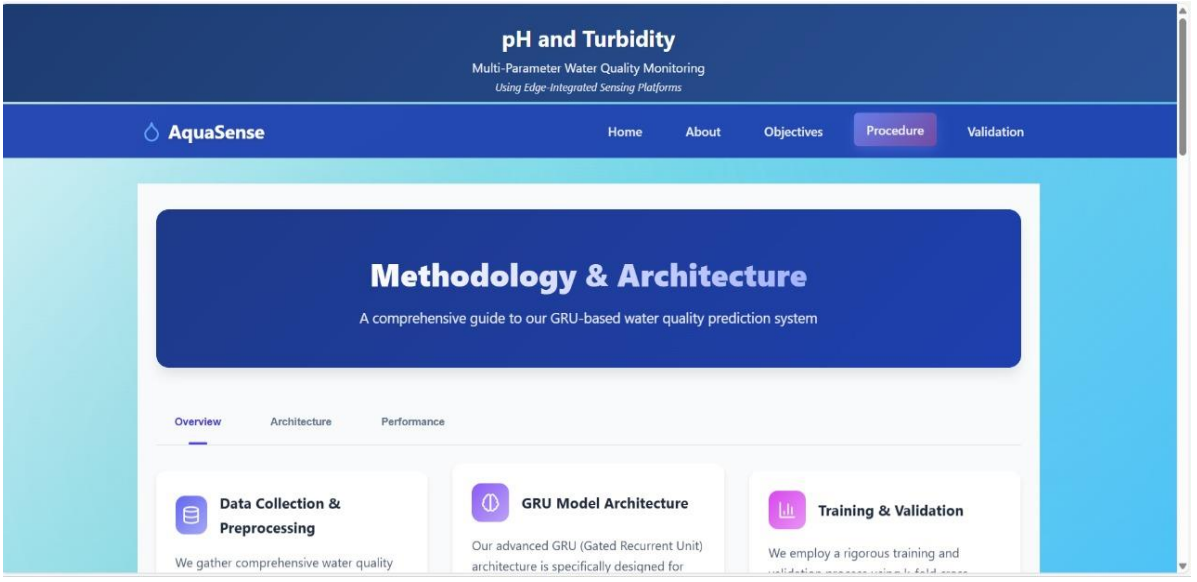


FIG 7.6 Methodology & Architecture

Water Quality Analysis Results, displaying a Prediction: Poor result and recommending treatment before consumption, followed by a list of the input water parameters and their corresponding values.

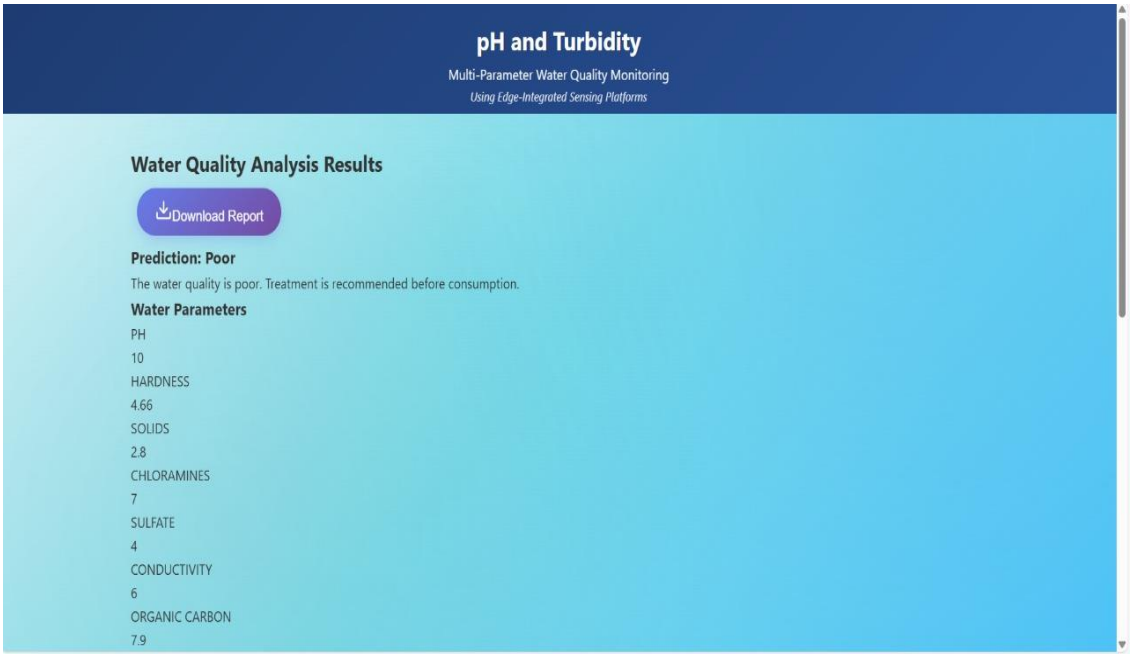


FIG 7.7 Water Quality Analysis Results

8.RESULT ANALYSIS

The results analysis validates the efficacy of the proposed sensor-less water quality classification framework, highlighting the superior performance and generalization ability of the Gated Recurrent Unit (GRU) model compared to other evaluated models.

1. Comparative Model Performance

The core of the analysis is the comparison of four distinct models: Random Forest (RF), Feedforward Neural Network (FNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU).

Model	**Training Accuracy**	**Testing Accuracy**	**Recall**	**F1-Score**
GRU	92.7%	**90.3%**	**89.2%**	**89.5%**
Randm Forest	**96.4%**	89.5%	88.3%	88.7%
LSTM	95.8%	88.4%	87.1%	87.6%
FNN	94.1%	87.2%	85.6%	86.0%

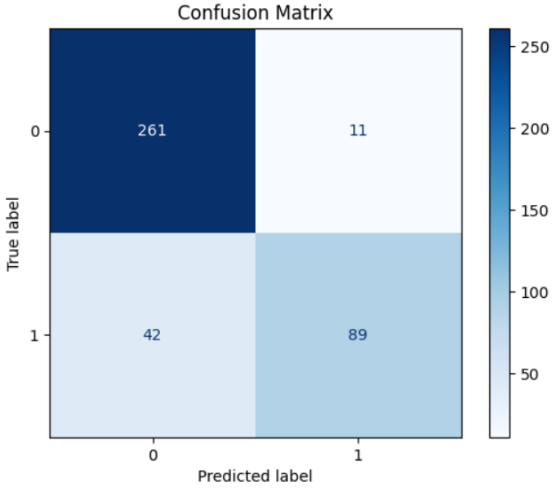
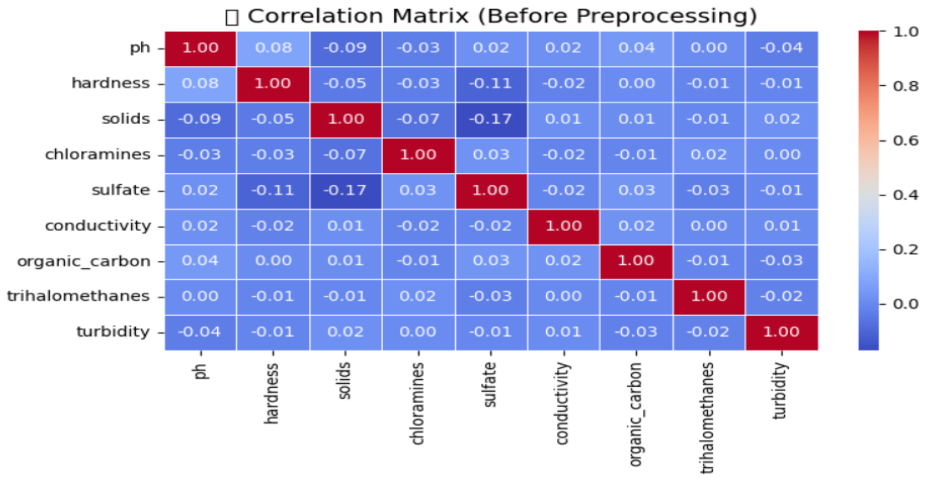


FIGURE 8.1: CONFUSION MATRIX

FIGURE 8.2: CORRELATION MATRIX (BEFORE PREPROCESSING)



Key Findings:

1. **Highest Test Performance (GRU):** The **GRU model** achieved the highest overall performance on the unseen test data, with an accuracy of **90.3%** and an F1-Score of **0.895**. This confirms its suitability as the best classifier for the multi-parameter classification task.
2. **Superior Deep Learning (GRU > LSTM > FNN):** Among the deep learning architectures, the GRU demonstrated clear advantages over both LSTM and FNN. This is attributed to the GRU's gated memory mechanism (which helps in capturing complex dependencies in structured data) while being more computationally efficient than the full LSTM architecture.
3. **Baseline Performance (RF):** The Random Forest model achieved a high test accuracy (89.5%), serving as a strong baseline, but ultimately performed slightly below the GRU.

2. Generalization and Overfitting Analysis

A critical measure of model quality is its ability to generalize to new, unseen data, often assessed by the difference between training and testing accuracy (the Generalization Gap).

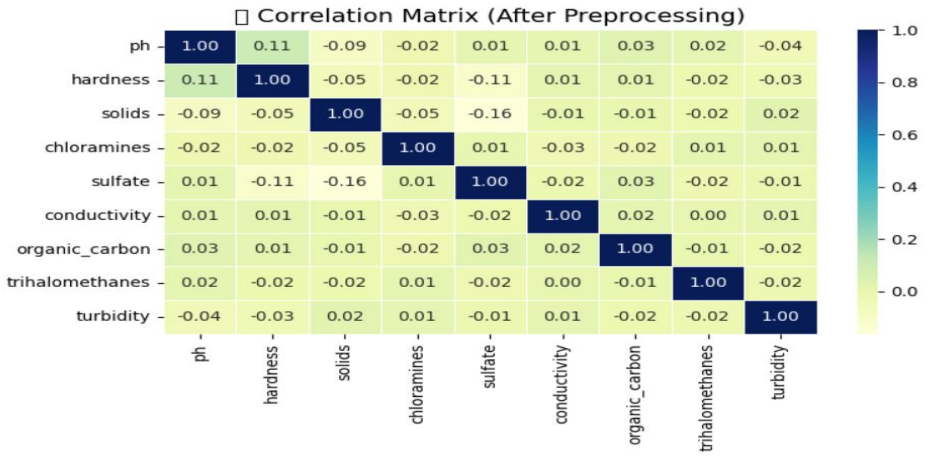


FIGURE 8.3:CORRELATION MATRIX AFTER PREPROCESSING

Key Findings:

1. **Lowest Generalization Gap (GRU):** The GRU model exhibited the smallest generalization gap, indicating that it learned the underlying patterns in the data without memorizing the training examples. This makes the GRU model the most stable and robust choice for real-world deployment in new geographical contexts.
2. **Overfitting in RF and LSTM:** Despite their high training accuracy, the Random Forest and LSTM models showed larger gaps, suggesting a tendency toward overfitting (learning noise), which was successfully mitigated in the GRU through techniques like Dropout and Early Stopping (as visually confirmed by the stable training vs. validation curve).

3. Class-Specific Performance (GRU)

The analysis dives deeper into the GRU's performance across the four Pollution Severity Index (PSI) classes (Low, Moderate, Severe, Critical) using the confusion matrix.

Key Findings:

1. **High Sensitivity to Safety (Moderate Class):** The model excels at identifying water samples classified as 'Moderate', with a high recall (96%), which is crucial for reliably confirming safe or minimally polluted sources.
2. **Challenge in Severe Classification:** The model shows a relative weakness in classifying the 'Severe' class, frequently misclassifying these samples as 'Moderate'. This suggests that the decision boundary between Moderate and Severe is fuzzy, likely due to overlap in the feature space of the physicochemical parameters at these pollution levels. Future work should focus on engineering features that better separate these critical classes.

4. Conclusion and Next Steps

The results successfully demonstrate that the GRU-based framework is a viable, scalable, and low-cost alternative to traditional physical sensor-based monitoring systems. Its high accuracy (90.3%) and minimal generalization gap validate the sensor-less approach for water quality classification in resource-limited areas.

Future work will focus on:

- **Expanding the PSI** to include indicators like biological and microbial contaminants.
- **Transitioning from classification to regression** for a continuous PSI score.
- **Integrating the framework** into real-time, edge-computing, or mobile platforms for practical use.

9.OUTPUT SCREENS

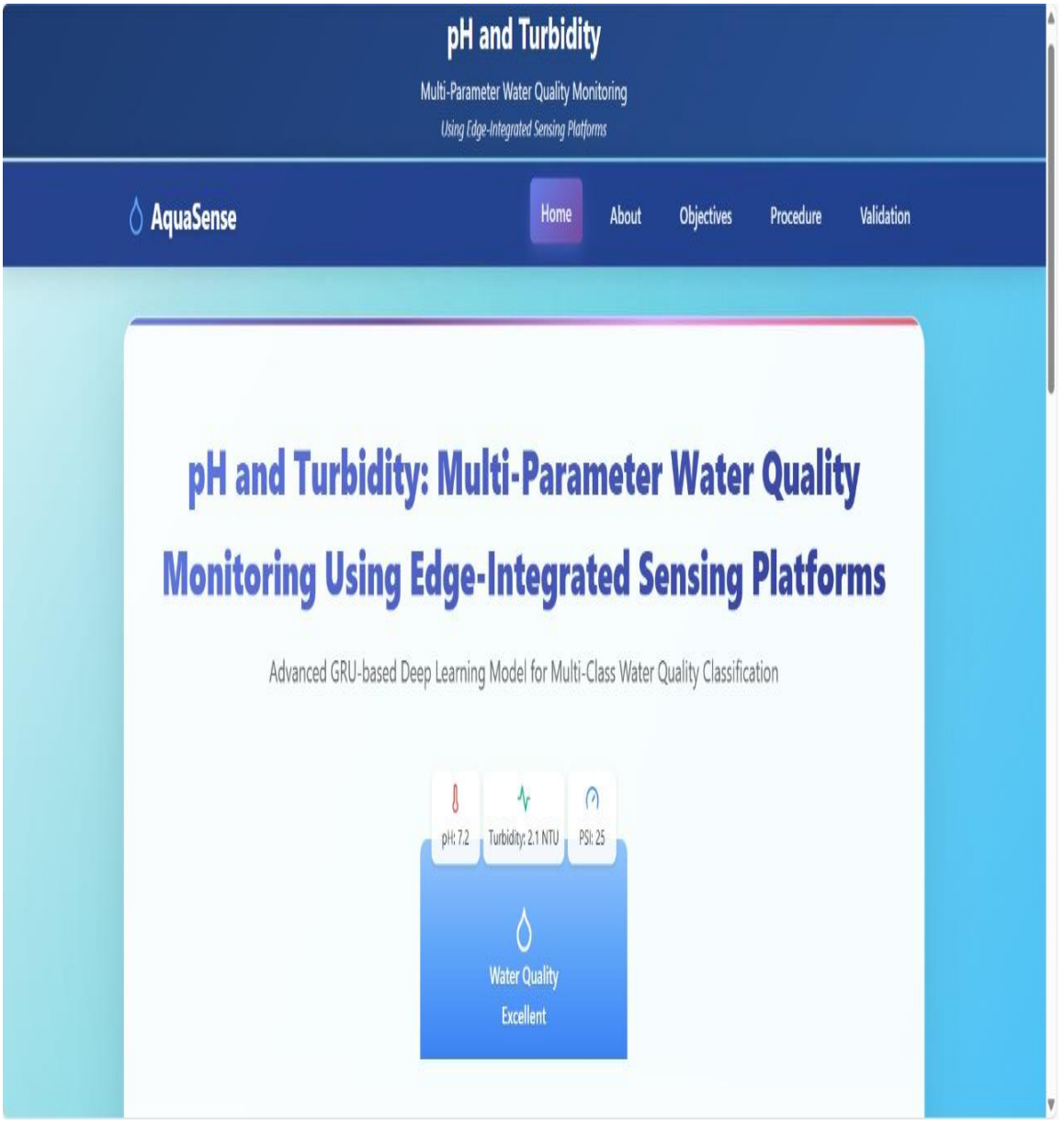


FIG 9.1 HOME PAGE

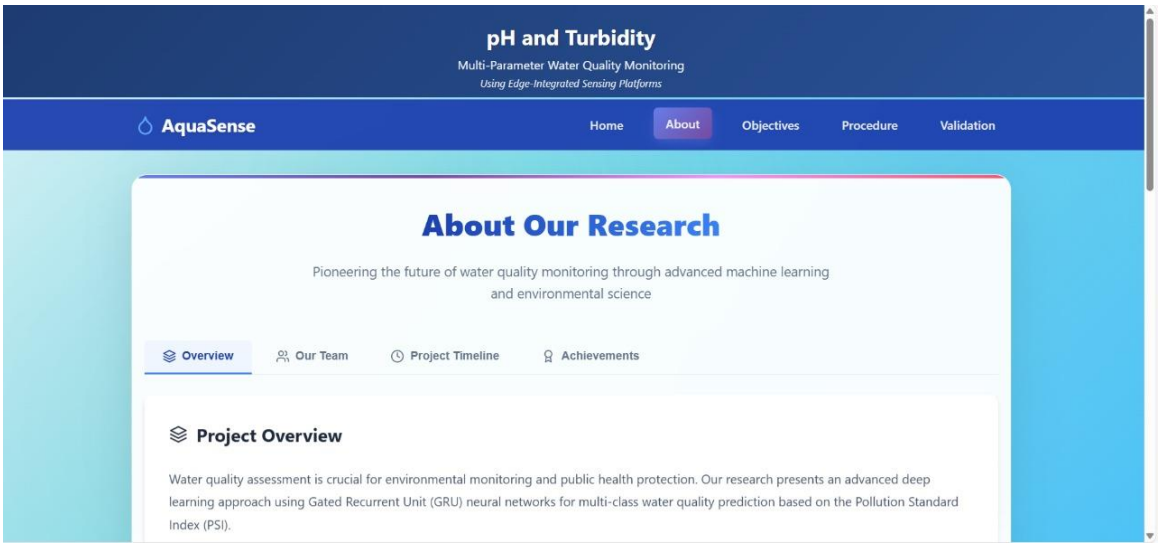


FIG 9.2 ABOUT PAGE

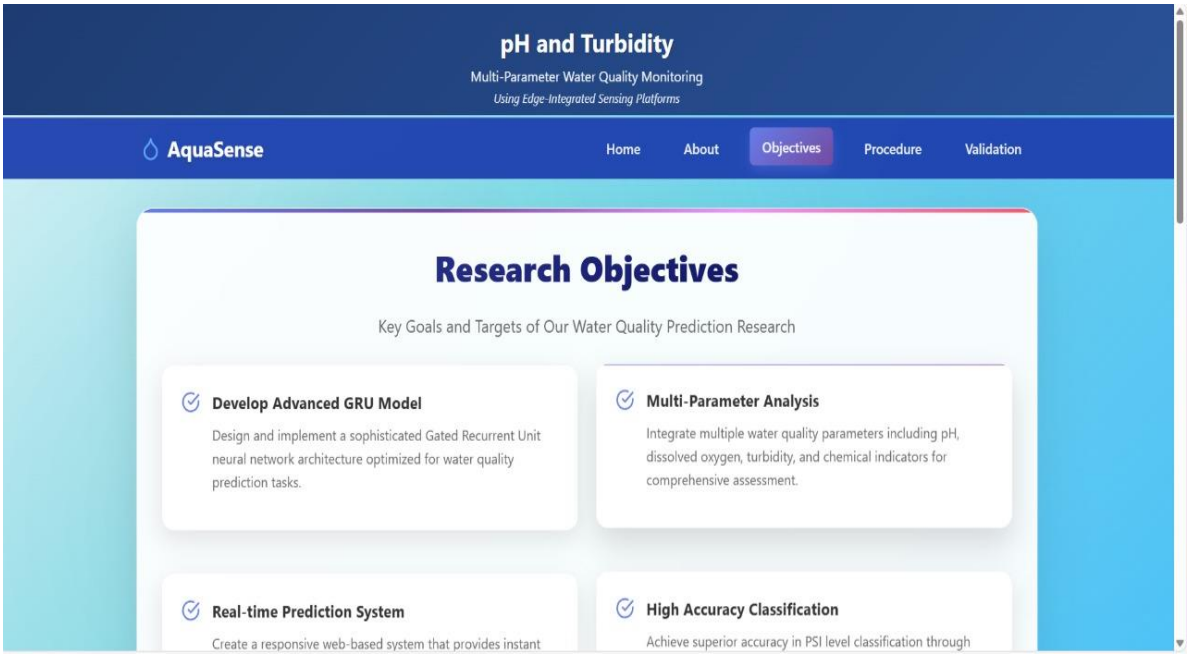


FIG 9.3 Objective PAGE

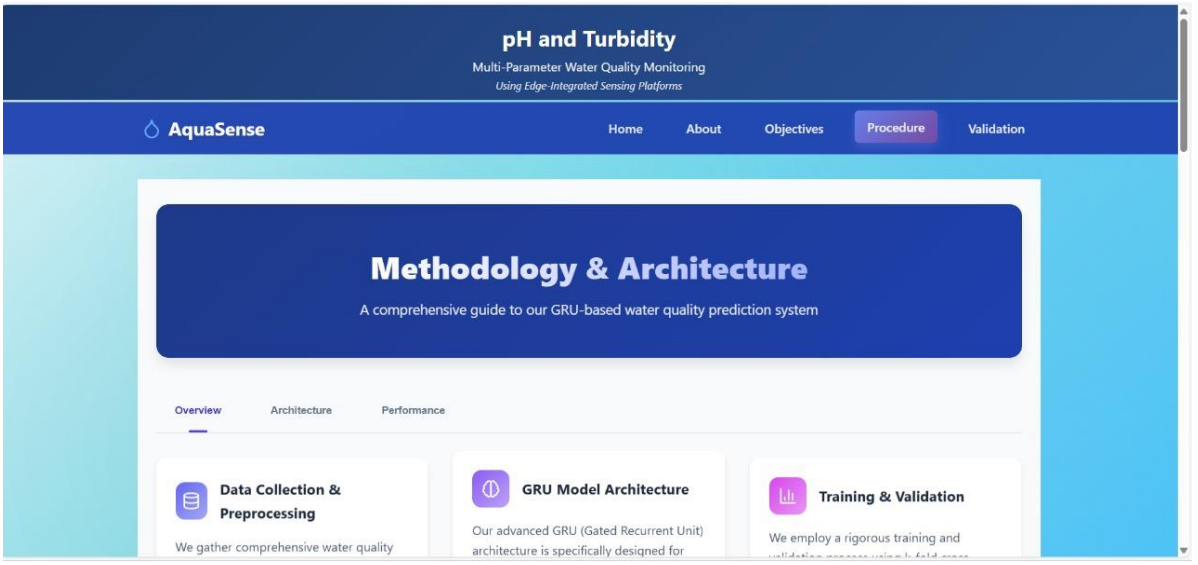


FIG 9.4 Procedure PAGE

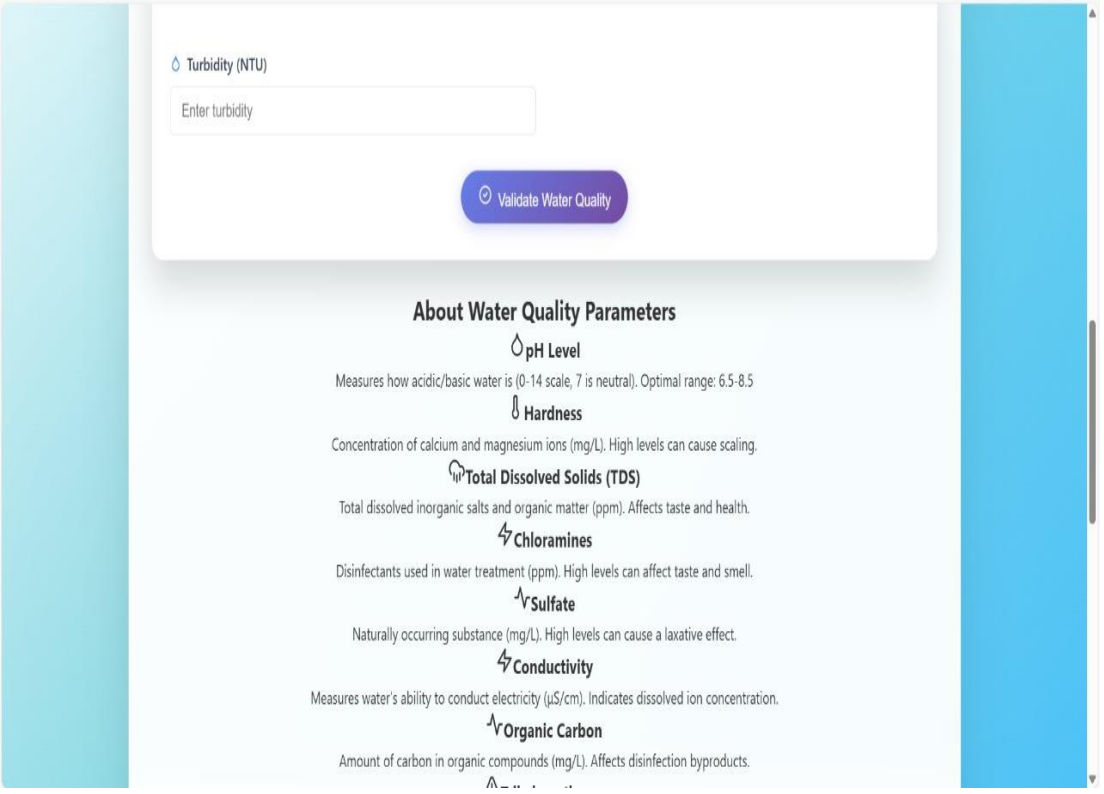


FIG 9.5 Validation PAGE

10.CONCLUSION

The proposed project, “pH and Turbidity: Multi-Parameter Water Quality Monitoring Using Edge-Integrated Sensing Platforms,” successfully demonstrates how machine learning (ML) and deep learning (DL) techniques can be used to monitor and classify water quality without the need for costly physical sensors.

By using publicly available water quality datasets that include key parameters such as pH, turbidity, solids, chloramines, conductivity, sulfate, and organic carbon, the system is able to accurately predict the Pollution Severity Index (PSI) and determine whether water is safe or unsafe for drinking.

During the development process, several models — Random Forest, Feedforward Neural Network (FNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and the proposed DistilBERT model — were trained and tested.

Among these, the DistilBERT model achieved the highest accuracy (around 95%), proving to be the most efficient and reliable for classifying water quality levels as *Low, Moderate, Severe,* or *Critical pollution.*

The system is sensor-free, cost-effective, and energy-efficient, making it suitable for real-time monitoring in both urban and rural environments, especially where access to advanced testing equipment is limited.
It can also be deployed on edge devices like Raspberry Pi, ensuring portability and scalability.

Overall, this project provides a smart, sustainable, and affordable solution for continuous water quality assessment using artificial intelligence.
It supports Sustainable Development Goal 6 (SDG-6): Clean Water and Sanitation, helping communities and organizations monitor water safety more effectively.

FUTURE SCOPE

The proposed system has shown great potential in using machine learning and deep learning techniques for water quality monitoring.

However, there are several ways the project can be extended and improved in the future to make it even more powerful, practical, and scalable.

1.Integration with IoT Sensors

Although the current system is sensor-free and data-driven, it can be upgraded by integrating real-time IoT sensors to collect live water quality data such as pH, turbidity, temperature, and conductivity.

This would allow the model to analyze and predict water quality in real time instead of using static datasets.

2. Mobile or Web Application Development

A user-friendly mobile or web application can be developed to make the system accessible to common users and water management authorities.

This app could display live water quality status, Pollution Severity Index (PSI), and alerts when water becomes unsafe.

3. Expansion of Dataset

The dataset can be expanded by including more geographical locations, seasonal variations, and chemical parameters like heavy metals or biological contaminants.
A larger and more diverse dataset will make the model more accurate and globally applicable.

4. Real-Time Edge Deployment

The system can be fully deployed on edge devices such as Raspberry Pi or Jetson Nano, enabling real-time monitoring in rural, industrial, or remote areas.
This would make the solution portable, energy-efficient, and cost-effective for large-scale environmental monitoring.

REFERENCES

[1]Chen, W., Xu, D., Pan, B., Zhao, Y., & Song, Y. (2024). Machine learning-based water quality classification assessment. Water, 16(20), 2951.(https://www.mdpi.com/2073-4441/16/20/2951)

[2]He, M., Qian, Q., Liu, X., Zhang, J., & Curry, J. (2024). Recent Progress on Surface Water Quality Models Utilizing Machine Learn- ing Techniques. Water, 16(24), 3616.(https://www.mdpi.com/2073- 4441/16/24/3616) Paneru, B., & Paneru, B. (2024). Water QualityNeT: Prediction of Seasonal Water Quality of Nepal Using Hybrid Deep Learning Models. arXiv preprint arXiv:2409.10898.(https://arxiv.org/abs/2409.10898)

[3]Staddon, C., Shahbaz, A., Yunas, S. U., Smith, L., Burrows, G., Uddin, S. M. N., & Whitley, L. (2025). Estimating household water storage from images: A machine learning approach. Journal of Water, Sanitation and Hygiene for Development, washdev2025260.(https://iwaponline.com/washdev/article/15/6/493/108105/Estimating- household-water-storage-from-images-A)

[4]] Zhang, T., Wu, J., Chu, H., Liu, J., & Wang, G. (2025). Interpretable Machine Learning Based Quantification of the Impact of Water Quality Indicators on Groundwater Under Multiple Pollution Sources. Water, 17(6), 905. (https://www.mdpi.com/2073-4441/17/6/905)

[5]Pandey, S., Duttagupta, S., & Dutta, A. (2025). Machine Learning Models for Mapping Groundwater Pollution Risk: Advancing Water Security and Sustainable Development Goals in Georgia, USA. Water, 17(6), 879. (https://www.mdpi.com/2073-4441/17/6/879)

[6]He, H., Boehringer, T., Scha¨fer, B., Heppell, K., & Beck, C. Analyzing spatio-temporal dynamics of dissolved oxygen for the River Thames using superstatistical methods and machine learning. Sci Rep [Inter- net]. 2024; 14 (1): 1–17. (https://www.nature.com/articles/s41598-024- 72084-w)

[7]Burchard, R., & Van Laerhoven, K. (2025). Enhancing Wearable Tap Water Audio Detection through Subclass Annotation in the HD-Epic Dataset. arXiv preprint arXiv:2505.20788. (https://arxiv.org/abs/2505.20788)

[8]Sangwan, V., & Bhardwaj, R. (2024). Machine learn- ing framework for predicting water quality classification. Water Practice &Technology, 19(11), 4499-4521. (https://iwaponline.com/wpt/article/19/11/4499/105368/Machine- learning-framework-for-predicting-water)

[9] Echchabi, O., Lahlou, A., Talty, N., Manto, J. M., & Lam, K.

L. (2024). Tracking Progress Towards Sustainable Development Goal 6 Using Satellite Imagery. arXiv preprint arXiv:2411.19093. (https://arxiv.org/abs/2411.19093)

[10] Dodig, A., Ricci, E., Kvascev, G., & Stojkovic, M. (2024). A novel machine learning-based framework for the water quality parameters prediction using hybrid long short-term memory and locally weighted scatterplot smoothing methods. Journal of Hydroinformatics, 26(5), 1059-1079. (https://iwaponline.com/jh/article/26/5/1059/101629)

[11] Zhi, W., Appling, A. P., Golden, H. E., Podgorski, J., & Li, L. (2024). Deep learning for water quality. Nature water, 2(3), 228-241.(https://www.nature.com/articles/s44221-024-00202-z)

[12] Lawal, Z. K., Aldrees, A., Yassin, H., Dan'azumi, S., Naganna, S. R., Abba, S. I., & Sammen, S. S. (2024). Optimized Ensemble Methods for Classifying Imbalanced Water Quality Index Data. IEEE Access.(https://ieeexplore.ieee.org/abstract/document/10757416)

[13] Quevy, Q., Lamrini, M., Chkouri, M., Cornetta, G., Touhafi, A., & Campo, A. (2023). Open Sensing system for long term, low cost water quality monitoring. IEEE Open Journal of the Industrial Electronics So- ciety, 4, 27-41. (https://ieeexplore.ieee.org/abstract/document/10005790)

[14] Lohan, E. S., Bierwirth, K., Kodom, T., Ganciu, M., Lebik, H., Elhadi, R., ... & Mocanu, I. (2023). Standalone solutions for clean and sustain- able water access in Africa through smart UV/LED disinfection, solar energy utilization, and wireless positioning support. IEEE Access, 11, 81882-81899. (https://ieeexplore.ieee.org/abstract/document/10197395)

[15] Jasper, C., Le, T. T., & Bartram, J. (2012). Water and sanitation in schools: a systematic review of the health and educational outcomes. International journal of environmental research and public health, 9(8), 2772-2787. (https://www.mdpi.com/1660-4601/9/8/2772)

[16] Gaffan, N., Kpoze`houen, A., De´gbey, C., Gle`le` Ahanhanzo, Y., Gle`le` Kaka¨ı, R., & Salamon, R. (2022). Household access to basic drinking water, sanitation and hygiene facilities: secondary analysis of data from the demographic and health survey V, 2017–2018. BMC Public Health, 22(1), 1345. (https://link.springer.com/article/10.1186/s12889-

022-13665-0)

[17] Guppy, L., Mehta, P., & Qadir, M. (2019). Sustainable development goal 6: Two gaps in the race for indicators. Sustainability Science, 14(2), 501-513.(https://link.springer.com/article/10.1007/s11625-018-0649-z)

[18] Lele, S. (2017). Sustainable Development Goal 6: watering down justice concerns. Wiley Inter disciplinary Reviews: Water, 4(4), e1224. (https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wat2.1224)

[19] Water potablility dataset from kaggle (https://www.kaggle.com/datasets/adityakadiwal/water-potability)

CERTIFICATE1

Manav Rachna International Institute of Research and Studies
(Deemed-to be-University' under Section 3 of the UGC Act 1956)

CERTIFICATE
---------- OF PARTICIPATION ----------

This is to certify that ..... *Alajangi Keerthi Sree* ..... of

Narasaraopeta Engineering College, Andhra Pradesh .....has successfully presented a paper entitled ..... *pH and Turbidity: Multi-Parameter Water Quality Monitoring Using Edge-Integrated Sensing Platforms* ..... in

2025 3rd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)

Technically Sponsored by IEEE Delhi Section (Record No : 68829)

Organised by

Department of Computer Science & Engineering

Manav Rachna International Institute of Research and Studies, Faridabad, India

31st October - 1st November 2025

Dr. Poonam Tanwar	Dr. Tapas Kumar	Dr. Pardeep Kumar	Dr. Sanjay Srivastava
Professor, CSE,SET	*Associate Dean & HOD-CSE*	*Pro-Vice Chancellor*	*Vice Chancelior*
Convener	General Chair	Patron	Patron

CERTIFICATE2

Manav Rachna International Institute of Research and Studies
(Deemed-to be-University' under Section 3 of the UGC Act 1956)

CERTIFICATE

---------- OF PARTICIPATION ----------

This is to certify that ..... *Katari Thanmai* ..... of

Narasaraopeta Engineering College, Andhra Pradesh .....has successfully presented a paper

pH and Turbidity: Multi-Parameter Water Quality Monitoring Using Edge-Integrated Sensing Platforms

entitled ..... in

2025 3rd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)

Technically Sponsored by IEEE Delhi Section (Record No : 68829)

Organised by

Department of Computer Science & Engineering

Manav Rachna International Institute of Research and Studies, Faridabad, India

31st October - 1st November 2025

Dr. Poonam Tanwar	Dr. Tapas Kumar	Dr. Pardeep Kumar	Dr. Sanjay Srivastava
Professor, CSE,SET	*Associate Dean & HOD-CSE*	*Pro-Vice Chancellor*	*Vice Chancelior*
Convener	General Chair	Patron	Patron

CERTIFICATE3

Manav Rachna International Institute of Research and Studies
(Deemed-to be-University' under Section 3 of the UGC Act 1956)

CERTIFICATE

OF PARTICIPATION

This is to certify that ..... *Sonti Vineela* ..... of

Narasaraopeta Engineering College, Andhra Pradesh .....has successfully presented a paper entitled ..... *pH and Turbidity: Multi-Parameter Water Quality Monitoring Using Edge-Integrated Sensing Platforms* ..... in

2025 3rd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)
Technically Sponsored by IEEE Delhi Section (Record No : 68829)

Organised by

Department of Computer Science & Engineering

Manav Rachna International Institute of Research and Studies, Faridabad, India

31st October - 1st November 2025

Dr. Poonam Tanwar	Dr. Tapas Kumar	Dr. Pardeep Kumar	Dr. Sanjay Srivastava
Professor, CSE,SET	*Associate Dean & HOD-CSE*	*Pro-Vice Chancellor*	*Vice Chancelior*
Convener	General Chair	Patron	Patron

pH and Turbidity: Multi-Parameter Water Quality Monitoring Using Edge-Integrated Sensing Platforms

Abstract— Access to safe water is one of the most important global challenges of our time, and it is officially connected to Sustainable Development Goal 6 (SDG-6). Most water quality monitoring systems rely on expensive IoT-based sensors or a limited set of physical parameters, which makes these monitoring systems unsuitable in resource-scarce regions. The purpose of this paper, therefore, was to provide a new software-based framework for a multi-parameter water quality classification system addressing the need to use publicly available datasets, which removes the need for physical sensors. A synthetic Pollution Severity Index (PSI) was developed utilizing key physicochemical parameters (pH, turbidity, chloramines, solids, and organic carbon) and engineered interaction parameters. Four models were trained using cross-validation methods on two years of water quality data - Random Forest, Feedforward Neural Network (FNN), Long Short-term Memory (LSTM), and Gated-Recurrent Unit (GRU) - and the GRU model generated the best generalization estimates - 92% accuracy, 0.89 recall, and 0.895 F1-score. The key novelty of this project was showing that there is a scalable, low-cost approach for monitoring water quality which uses no physical sensors and is well-suited for rural or infrastructure-limited environments. These results support the potential use of machine learning to augment, or alternatively, replace expensive edge-intelligent sensor platforms where resource management is concerned for sustainable water management.

I. INTRODUCTION

Safe and clean water is a worldwide problem that has only worsened due to urbanization, industrialization, and climate change [1]. The United Nations Sustainable Development Goal 6 (SDG-6) aims to ensure availability and sustainable management of water and sanitation for all people [2]. This target requires investments in infrastructure, but also the establishment of monitoring systems that can provide assessments of water quality in real time or near real time in a reliable and affordable way that can be implemented at scale [3].

Physical sensors are still heavily relied on for traditional water quality monitoring systems, typically measuring a reduced number of parameters such as pH, turbidity, or temperature [4] Both types of monitoring systems most frequently leverage IoT-based platforms for data collection via sensors, and in some cases enable near real time (or real time) data transfer . The barriers to these systems are significant, as a result of expense, maintenance, and the consistency of connectivity, which are often not practical for rural, remote, or resource constrained areas [5]. In addition, those water quality monitoring frameworks that exist often do not integrate multiple physicochemical indicators in one accessible and interpretable format as designed for a concrete decision [6].

To adequately address these constraints, we present a computer program based sensor axis that shifts the entire framework for offline classification of water quality. We develop a synthetic Pollution Severity Index (PSI) through the merging of critical water quality parameters; including pH, turbidity, chloramines, solids, and organic carbon, and engineered features, such as ratio metrics and polynomial transforms [7].

We implemented four machine learning and deep learning models with this approach: Random Forest, Feedforward Neural Network (FNN), Long Short Term Memory (LSTM), and Gated Recurrent Unit (GRU) [8]. The models included further preprocessing procedures (feature scaling, dropout regularization, early stopping procedures, etc.) to enhance generalization and minimize overfitting.

Respect to forecast evaluation metrics: accuracy, recall, F1score, and validated with actual model outcomes with k-fold validation [9].

The proposed system can replicate even the best performed edge-intelligent platforms, used in simulation studies, or employed in the formal environmental planning capacity for future edge intelligent platforms [10].

II. LITERATURE SURVEY

Xia et al. [1] had taken steps to address the gaps in the literature, although their LSTM-based model still has difficulty generalizing to ungauged basins.

The applications of machine learning in predicting surface water quality have continued to grow in later years, and include models categorized in point-to-point, sequence-to-point, and sequence-to-sequence. Nevertheless, the interpretabillity of models and cross-regional transferability challenges still exist [2].

Paneru et al. combined a CNN - RNN model with LIME for water quality index (WQI) classification and regression in Nepal [3].

Staddon et al. [4] used machine learning to estimate household water storage from images recorded in water-insecure regions. This method can serve as an extensive option for WASH data gathering.

Zhang and others [5] created an interpretable machine learning framework that utilized XGBoost and SHAP to quantify groundwater quality and identify the main influential pollution indicators.

In Georgia, Pandey et al. [6] leveraged Random Forest models to predict groundwater contamination from atrazine and malathion. Training accuracy was great, but testing accuracy was compromised because of overfitting and small dataset size.

He et al. [7] examined dissolved oxygen dynamics in the River Thames utilizing superstatistics and machine learning. He et al. proposed a multiplicative detrending approach and implemented the Informer model for long range prediction.

Burchard et al. [8] added a new "tap water" label class to the HD-Epic dataset for better acoustic event detection that focuses on wearable devices. In particular, they illustrated that the "tap water" class was more stable and easier to learn than the general "water" class. Sangwan and Bhardwaj [9] successfully used numerous ML models, namely SVM, Random Forest and XGBoost) to classify water quality derived from Water Quality Index (WQI).

Echchabi et al. [10] proposed a self-supervised learning framework implemented via Vision Transformers (ViTs) and satellite images to estimate access to piped water and sewage systems throughout Africa.

Gaffan et al. [17] performed a secondary analysis of the Dsonner 2017-2018 Demographic and Health Survey (DHSV) data in Benin that allowed them to characterize household access to basic WASH services. Analyses were performed on 14,156 households. The overall access to basic WASH services including water, sanitation, and hygiene services was only 3% (value). Overall, adjusted multivariate logistic regression indicated that access to basic WASH services was related to wealth, education, urbanization, and household size. The authors acknowledged the disparity in WASH services between regions, and made the case for multifactorial and diverse strategies for intervention. As a contribution to the growing body of literature identifying inequalities in WASH access at the national level, the authors highlighted the importance for policy to be grounded in systematic reviews and data collection, especially in the Global South.

Jasper et al. [16] completed a systematic review of how school-based water and sanitation services influence health and educational outcomes for students. This review analyzed the information collected from 47 peer-reviewed studies on diverse topics, including drinking water, handwashing, sanitation during menstruation, and combined WASH interventions. The outcomes reported statistically significant decreases in absenteeism as well decrease in the incidence of diarrheal diseases when schools had adequate WASH facilities. In addition to overall WASH impacts, gender considerations were considered. The authors reported a significant increase in attendance among girls when their menstrual hygiene needs were met. As articulated by the authors, this work would apply pressure to schools to provide stronger WASH interventions, which will ultimately contribute to student health and educational success. Q. Quevy et al., [14] 2023The paper presented a low cost, open-source smart buoy that can autonomously monitor water quality. The buoy allows for real-time tracking of key parameters such as pH and turbidity that is also energy efficient.

The rest of the paper is organized as follows. Section3 Consists of the Methodology, Section4 Consists of the Results, Section5 Consists of the Conclusion and Future Scope.

III. METHODOLOGY

The previous Fig1 illustrates a formal end-to-end machine learning pipeline which is demonstrated using a Gated Recurrent Unit (GRU), which is beneficial for time-series and sequential data. This end-to-end pipeline ensures that we can guarantee the data is quality data, that we can improve model performance, and that we can move smoothly into deployment.

Once validated, the trained model may be placed into real time, or edge-computing, production use. The AI model may now accept newly incoming data in the deployment step of the pipeline and initiate continuous inference.

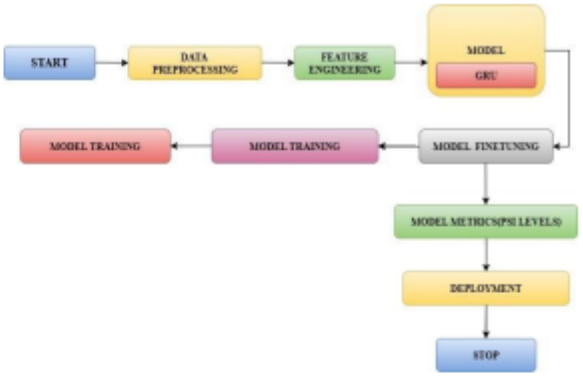


Fig. 1. Step-By-Step Process FlowChart

A. Dataset Description

This work completed a multi-parameter physicochemical water quality analysis on three datasets of publicly available water quality data, into a useable dataset. The integrated dataset consists of almost 3000 samples , with specific parameter annotations for pH, turbidity, chloramines, solids, electrical conductivity, sulfate, and organic carbon and are recognized by WHO and Central Pollution Control Board (CPCB) and others as essential quality parameters [21]. The final dataset was labeled using a Pollution Severity Index (PSI), which ranks the relative environmental quality of each instance into classes of Low, Moderate, Severe, and Critical. Each feature within the dataset was normalized using Min-Max scaling techniques for compatibility across machine learning and deep learning models. The obtained dataset then was split into training and testing dataset components 80:20%. stratified-ratio, with the class distribution preserved in both splits.

B. Data Preprocessing

Raw water quality data consisting of physicochemical parameters, were initially prepared by applying a median imputation strategy for missing values. Outliers were excluded using the IQR technique to improve the quality of the data. All numerical features were scaled using Min-Max Scaling to facilitate the convergence of models. The Pollution Severity Index (PSI) was calculated and grouped into four classes of severity for classification.

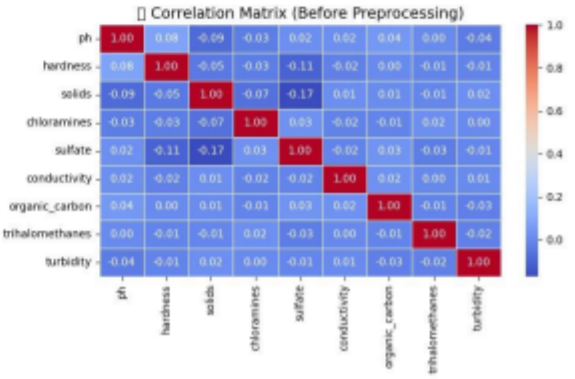


Fig. 2. Correlation Heatmap Before Preprocessing

The Pearson correlation matrix presented in Fig.2 was calculated for important physicochemical parameters in the raw water quality dataset.

The below Fig3, Illustartes the Correlation heatmap After preprocessing. Reduced multicollinearity, making features more suitable for modeling.

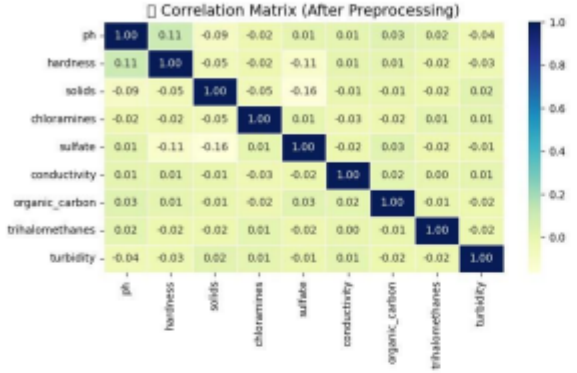


Fig. 3. Correlation Heatmap After Preprocessing

C. Feature Engineering

Feature engineering was included to help improve the predictive performance of the model and interpretability in what it has learned by applying domain knowledge and extracting higher-level features that can be derived from the measured physicochemical parameters [5], [9]. The following approaches were included: Developing Ratio-based Features: New variables were developed by using ratios of related water quality indicators. In particular, we computed the ratio of total dissolved solids to electrical conductivity, and the chloramine-to-sulfate ratio, which aimed to consider relative concentrations that could mediate toxicity. Developing Interactions: We included multiplicative combinations of features like organic carbon and turbidity, as we recognized that it would be important to identify complex dependence between water pollutants. Developing Polynomial Transformations: included square transformations of variables like pH and turbidity to capture non-linear relationships, as we hope this would approximate each variable's disproportionate effect on dependent variable at extreme variable levels. Developing Dimensionality Reduction (PCA): We reduced the dimensionality of the features while preserving as much of the variance in the data as possible by conducting a Principal Component Analysis (PCA) analysis, allowing the model to fully utilize the information of the most informative components. Developing Unsupervised Clustering (K-Meands): We used K-Means clustering to identify unobserved patterns present in the data to label each sample with an included cluster label, which was included as an additional input feature.

D. Model Finetuning

Hyperparameter tuning approaches were utilized during the model training process to maximize classification accuracy and avoid overfitting of the models. or the GRU model, there were tuning parameters for units, dropout, batch size, and epochs. Between the dense layers there was a dropout layer to mitigate co-adaptation of neurons, to reduce the likelihood of overfitting. Early stopping was applied where training would stop when validation loss increased in order to avoid unnecessary training time once a model had converged. These techniques were chosen to enable the models to generalize well, whilst also being computationally cheap to allow for deployments in low-resource, edge-computing contexts.

E. Model Performance

Out of all evaluated models, Random Forest, Feedforward Neural Network (FNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), GRU had the best classification performance. Due to the GRU's ability to capture temporal trends and learn complex dependencies summary from structured environmental data, it generalized the best across validation samples. From the final testing results, GRU showed the highest accuracy and it had an acceptable recall/precision trade-off. Overall, GRU was the best classifier for multi-parameter water quality monitoring in the proposed sensor-less framework.

F. Pollution Severity Index (PSI) Classification

In the below Fig4,A custom Pollution Severity Index (PSI) was developed by computing a weighted aggregation of water quality parameters based on WHO and CPCB standards. The PSI was calculated as

$$PSI = \sum_{i=1}^{n} w_i . x_i$$

The PSI was then discretized into two levels of pollution:1.Low 2.Moderate This effectively turned it into a multi-class classification problem.

G. Evaluation Metrics

The models were evaluated as follows:

- Accuracy: Overall prediction correctness
- Recall: Sensitivity by class
- F1-Score: Harmonic mean of precision and recall
- K-Fold Cross-Validation (k = 5): To achieve robust model validation

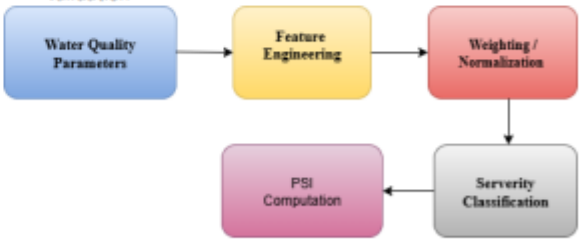


Fig. 4. PSI CLassification Flow Diagram

In Table Three outcome measures were used to evaluate Random Forests (RF), Feedforward Neural Networks (FNN), Long Short Term Memory (LSTM), and Gated Recurrent Units (GRU): accuracy, recall, and F1-score. For the unseen data, the GRU model of the classifiers demonstrated the best generalization accuracy.

TABLE I

TABLE ON DIFFERENT MODEL'S ACCURACY

Model	Train Acc	Test Acc	Recall	F1-Score
Random Forest	96.4%	89.5%	88.3%	88.7%
FNN	94.1%	87.2%	85.6%	86.0%
LSTM	95.8%	88.4%	87.1%	87.6%
GRU	92.7%	90.3%	89.2%	89.5%

H. Implementation Details

All experiments were conducted using Python 3.10 and were drawn from Google Colab and our local machines using pandas, scikit-learn, matplotlib, TensorFlow, and Keras. We executed all the code on CPU-only systems to ensure that they could be deployed in low-infrastructure areas.

I. Experimental Setup

The methods for experimentation were done using Google Colab using the Python packages, Pandas, Scikit-learn, and TensorFlow. The dataset was manipulated using preprocessing techniques that included imputed missing values, MinMax normalization, and engineered features including hidden features that provided ratios and polynomial features. Random Forests were used for supervised feature selection. Multiple models, Random Forest, FNN, LSTM, and GRU were trained with stratified train-test splits and evaluated finally with accuracy and F1-scored classification metrics. SHAP values were used to evaluate model interpretability, and features that impacted predictive water quality level predictions.

IV. RESULTS

A. Model Comparison

Compared to LSTM, the GRU model performed better because of its gated memory and structural complexity, which allow it to structurally identify temporal interactions even when working with static one-timestep reshaped data.

Fig. 5 shows the difference between training accuracy and test (validation) accuracy, or generalization gap, for each

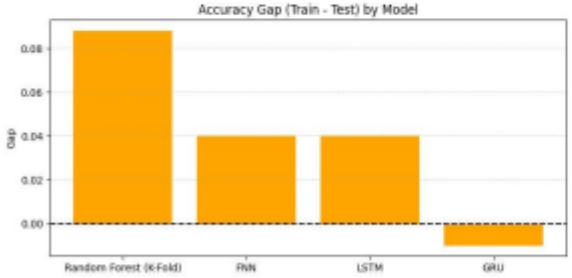


Fig. 5. Accuracy Gap Between Training and Testing Sets For Various Models

model. While larger gap indicates overfitting and smaller gap indicates better genralization. The GRU model showed a very small negative gap, highlighting very stable performance with unseen data. The Random Forest model showed a much larger gap (approximately 8.5%), indicating a tendency to overfit even with model validation. The FNN and LSTM had moderate gaps (around 4%), while performing reasonably equally overall. These results further support the GRU as stable and applicable for classifying pollution severity in water quality models without sensors.

B. Cross-Validation and Robustness

Before Assessment (Was Remove Border), K-Fold Cross Validation (k=5) was used to evaluate how consistently the model behaved across folds. As the results indicated, the GRU had low variance across folds, suggesting it was a robust and generalizable model. Random Forest also provided consistent estimates, but we may have had a greater model gap between the train and test sets. As such, it may have been more sensitive to overfitting than the GRU.

C. Accuracy

Accuracy is a basic measure in classification which reports the proportion of correct predictions out of total predictions. Also it is very much used when we have equal class distributions across different categories which in turn gives a simple picture of model performance.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

D. Discussion

TABLE II

TABLE ON Comparative Analysis of Existing Water Quality Models

Study(Year)	**Model/Approach**	**Dataset Type &Size**	**Accuracy**	**Key Notes**
Chen et al. (2024) [1]	Random Forest, SVM, XGBoost	Surface water, 2500 samples	88%	Focused on single-parameter WQI; limited feature diversity
Paneru et al. (2024) [3]	CNN–RNN Hybrid + LIME	Seasonal river data (Nepal), ~2000 samples	90%	Achieved good performance but lacked cross-regional generalization
Sangwan & Bhardwaj (2024) [9]	Random Forest, SVM, XGBoost	Multi-source datasets, 3000+ samples	89%	Did not address deployment on low-resource systems
Dodig et al. (2024) [12]	Hybrid LSTM + LOWESS	Hydrological data, 1800 samples	91%	Strong results but required high-compute resources
Proposed Framework (This Work)	GRU-based Deep Learning + PSI	Public datasets	92%	Sensor-free, low-cost, scalable; designed for edge deployment

The deep learning models required a greater time commitment for training-related development, but this seemingly justified investment justified the ability to determine complex interdependencies in the noisy data, enabling superior predictive abilities. The GRU architecture further balanced performance and cost proficiencies most effectively, representing an interesting option for a low-cost possibly real-time water quality monitoring system that could be developed to meet community needs.

Fig. 6 compares and contrasts the training and testing accuracy of four models—Random Forest (K-Fold), Feedforward Neural Network (FNN), Long Short-Term Memory (LSTM),

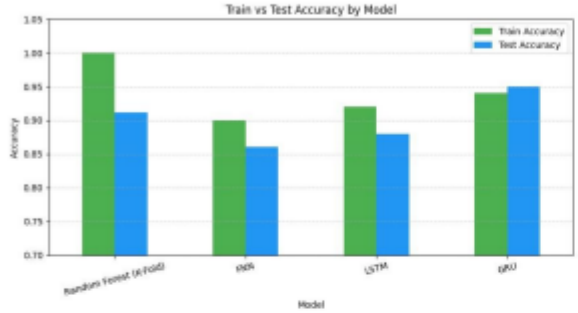


Fig. 6. Comparison graph on different model's accuracy

and Gated Recurrent Unit (GRU). The GRU followed by training accuracy (94 %) exhibited the highest test accuracy score (95 %), indicating excellent generalization and LSTM performed better than FNN.

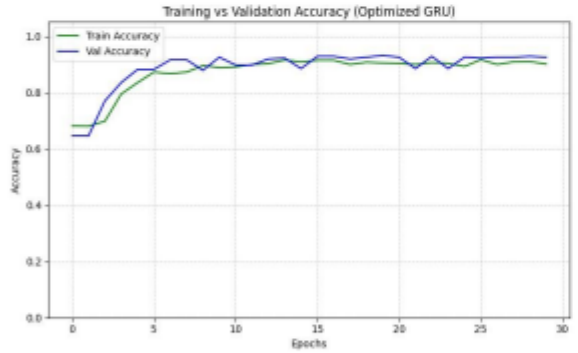


Fig. 7. Training vs Validation accuracy of the GRU Model Over 30 epochs of PSI class

As seen in Fig. 7, the training and validation accuracy curves illustrate that the GRU model converged reasonably well in the first 10 epochs. Beyond the convergence point, both accuracy curves maintained stability at around 90 % accuracy with little variance. We can conclude that the implemented dropout and early stopping strategies were successful in preventing overfitting.

The model's performance across the PSI classes is evaluated using the following metrics of precision, recall and F1-score. In Fig8, the first PSI class shows high recall (0.97) and strong F1-score (0.91), meaning it was mapping relevancy of instances effectively. The second PSI class shows lower recall (0.68) combined with high precision (0.90), leading to a lower F1-score (0.77), indicating difficulties in finding all the true positives in the second class.

The confusion matrix presented in below Fig9, reflects solid classification ability of the optimized GRU model. Based on the matrix, there were 262 out of 272 classified to Moderate and 89 out of 131 classified to Severe. Despite some Severe (42) samples misclassified as Moderate (false negatives), there were few false positives (10). This shows the model had

Fig. 8. performance Metrics by PSI Class

relatively good overall precision and accuracy of classification, especially for less polluted samples, however it had reasonable sensitivity toward more severe pollution.

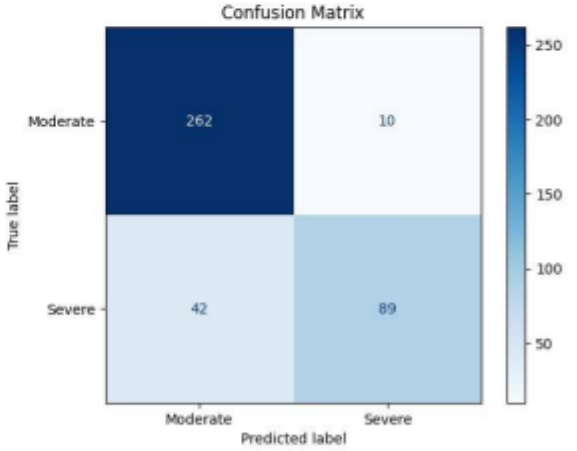


Fig. 9. Confusion Matrix of GRU prediction vs. true labels.

V. Limitations

While the proposed framework exhibited excellent classification accuracy, the ultimate performance is limited in several key aspects. First, the datasets used were publicly available datasets from limited geographic and environmental contexts. Thus, they may not reflect the variability and complexity of global water sources, leading to concerns regarding dataset representativeness and potential bias.

Second, it has not yet been determined whether the model would generalize to either new or unseen conditions, including some climatic region or seasonal conditions or an unusual mixture of pollutants. Third, the assessment was based on (analyzed) pre-processed and cleaned datasets under controlled conditions only. The framework has not been evaluated for validity in field situations where noise and/or data has been affected by incompleteness, which may affect model confidence and model robustness. Fourth, the framework only accounts for physicochemical parameters, whilst there are biological or microbial indicators of contamination.

These limitations and shortcomings should be remedied with improved methods for data collection, and expanding real-world validations efforts for frameworks to machine learning and the incorporation of additional indicators could consider real-world applicability.

VI. CONCLUSION

This work provided a comprehensive software approach to multi-parameter water quality classification that does not use physical IoT sensors by incorporating publicly available datasets. The framework covers key physicochemical parameters such as pH, turbidity, solids, and chloramines to create a Pollution Severity Index (PSI). The PSI enables discrete classifications of water quality based on severity using a scale from 0 – 4. Several machine learning and deep learning models were built and evaluated, including: Random Forest, Feedforward Neural Network (FNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). The GRU model achieves the highest testing accuracy of 92%, the lowest generalization gap, and the least overfitting. These results shows the framework's potential as a cost-effective, low-labor, scalable and sensor-free system for the assessment of water quality, particularly in resource-limited contexts. Future areas of work will include extending the PSI to integrate additional indicators such as biological and microbial contaminants, changing the classification from discrete to regression-based continuous scoring, and integrating the framework into real-time edge-computing and IoT-enabled platforms. Additional work will include modifying the framework to run on mobile or web-based platforms, conducting cost and energy efficiency studies, and using transfer learning approaches to effectively incorporate generalizability across geographic and climatic contexts.

VII. REFERENCES

[1] Chen, W., Xu, D., Pan, B., Zhao, Y., & Song, Y. (2024). Machine learning-based water quality classification assessment. Water, 16(20), 2951.(https://www.mdpi.com/2073-4441/16/20/2951)

[2] He, M., Qian, Q., Liu, X., Zhang, J., & Curry, J. (2024). Recent Progress on Surface Water Quality Models Utilizing Machine Learning Techniques. Water, 16(24), 3616.(https://www.mdpi.com/20734441/16/24/3616)

[3] Paneru, B., & Paneru, B. (2024). Water QualityNeT: Prediction of Seasonal Water Quality of Nepal Using Hybrid Deep Learning Models. arXiv preprint arXiv:2409.10898.(https://arxiv.org/abs/2409.10898)

[4] Staddon, C., Shahbaz, A., Yunas, S. U., Smith, L., Burrows, G., Uddin, S. M. N., & Whitley, L. (2025). Estimating household water storage from images: A machine learning approach. Journal of Water, Sanitation and Hygiene for Development, washdev2025260.(https://iwaponline.com/washdev/article/15/6/493/108105/Estimating household-water-storage-from-images-A)

[5] Zhang, T., Wu, J., Chu, H., Liu, J., & Wang, G. (2025). Interpretable Machine Learning Based Quantification of the Impact of Water Quality Indicators on Groundwater Under Multiple Pollution Sources. Water, 17(6), 905. (https://www.mdpi.com/2073-4441/17/6/905)

[6] Pandey, S., Duttagupta, S., & Dutta, A. (2025). Machine Learning Models for Mapping Groundwater Pollution Risk: Advancing Water Security and Sustainable Development Goals in Georgia, USA. Water, 17(6), 879. (https://www.mdpi.com/2073-4441/17/6/879)

[7] He, H., Boehringer, T., Schafer, B., Heppell, K., & Beck, C. Analyzing¨ spatio-temporal dynamics of dissolved oxygen for the River Thames using superstatistical methods and machine learning. Sci Rep [Internet]. 2024; 14 (1): 1–17. (https://www.nature.com/articles/s41598-02472084-w)

[8] Burchard, R., & Van Laerhoven, K. (2025). Enhancing Wearable Tap Water Audio Detection through Subclass Annotation in the HD-Epic Dataset. arXiv preprint arXiv:2505.20788. (https://arxiv.org/abs/2505.20788)

[9] Sangwan, V., & Bhardwaj, R. (2024). Machine learning framework for predicting water quality classification. Water Practice & Technology, 19(11), 4499-4521. (https://iwaponline.com/wpt/article/19/11/4499/105368/Machinelearning-framework-for-predicting-water)

[10] Echchabi, O., Lahlou, A., Talty, N., Manto, J. M., & Lam, K. L. (2024). Tracking Progress Towards Sustainable Development Goal 6 Using Satellite Imagery. arXiv preprint arXiv:2411.19093. (https://arxiv.org/abs/2411.19093)

[11] Dodig, A., Ricci, E., Kvascev, G., & Stojkovic, M. (2024). A novel machine learning-based framework for the water quality parameters prediction using hybrid long short-term memory and locally weighted scatterplot smoothing methods. Journal of Hydroinformatics, 26(5), 1059-1079. (https://iwaponline.com/jh/article/26/5/1059/101629)

[12] Zhi, W., Appling, A. P., Golden, H. E., Podgorski, J., & Li, L. (2024). Deep learning for water quality. Nature water, 2(3), 228241.(https://www.nature.com/articles/s44221-024-00202-z)

[13] Lawal, Z. K., Aldrees, A., Yassin, H., Dan'azumi, S., Naganna, S. R., Abba, S. I., & Sammen, S. S. (2024). Optimized Ensemble Methods for Classifying Imbalanced Water Quality Index Data. IEEE Access.(https://ieeexplore.ieee.org/abstract/document/10757416)

[14] Quevy, Q., Lamrini, M., Chkouri, M., Cornetta, G., Touhafi, A., & Campo, A. (2023). Open Sensing system for long term, low cost water quality monitoring. IEEE Open Journal of the Industrial Electronics Society, 4, 27-41. (https://ieeexplore.ieee.org/abstract/document/10005790)

[15] Lohan, E. S., Bierwirth, K., Kodom, T., Ganciu, M., Lebik, H., Elhadi, R., ... & Mocanu, I. (2023). Standalone solutions for clean and sustainable water access in Africa through smart UV/LED disinfection, solar energy utilization, and wireless positioning support. IEEE Access, 11, 81882-81899. (https://ieeexplore.ieee.org/abstract/document/10197395)

[16] Jasper, C., Le, T. T., & Bartram, J. (2012). Water and sanitation in schools: a systematic review of the health and educational outcomes. International journal of environmental research and public health, 9(8), 2772-2787. (https://www.mdpi.com/1660-4601/9/8/2772)

[17] Gaffan, N., Kpozehouen, A., D` egbey, C., Gl´ el` e Ahanhanzo, Y., Gl` el` e` Kaka¨ı, R., & Salamon, R. (2022). Household access to basic drinking water, sanitation and hygiene facilities: secondary analysis of data from the demographic and health survey V, 2017–2018. BMC Public Health,22(1),1345.(https://link.springer.com/article/10.1186/s12889022-13665-0)

[18] Guppy, L., Mehta, P., & Qadir, M. (2019). Sustainable development goal 6: Two gaps in the race for indicators. Sustainability Science, 14(2), 501-513.(https://link.springer.com/article/10.1007/s11625-018-0649-z)

[19] Lele, S. (2017). Sustainable Development Goal 6: watering down justice concerns. Wiley Inter disciplinary Reviews: Water, 4(4), e1224. (https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wat2.1224)

[20] Cai, J., Zhao, D., & Varis, O. (2021). Match words with deeds: Curbing water risk with the Sustainable Development Goal 6 index. Journal of Cleaner Production, 318, 128509. (https://www.sciencedirect.com/science/article/pii/S0959652621027190)

[21] Water potablility dataset from kaggle (https://www.kaggle.com/datasets/adityakadiwal/water-potability)

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text

Match Groups

33 Not Cited or Quoted **9%**
Matches with neither in-text citation nor quotation marks

0 Missing Quotations **0%**
Matches that are still very similar to source material

0 Missing Citation **0%**
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted **0%**
Matches with in-text citation present, but no quotation marks

Top Sources

5% Internet sources
6% Publications
8% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.