Université de BLIDA 1

Faculté des Sciences

Département d'Informatique

FINAL REPORT

Option : Systèmes InformatiQues

WATER QUALITY MONITORING SYSTEM

By :

BLAILI ZAKARIA & SEBIHA MASSINISSA NAIL

In front of a jury composed of:

Dr. YKHLEF Hadjer **Supervisor**

Année universitaire 2023/2024

Abstract

Water quality monitoring is crucial for environmental preservation and public health. In Algeria, like many regions globally, ensuring water quality remains a significant concern. This report presents a Water Quality Monitoring System leveraging Internet of Things (IoT) technology, predictive analytics, and web-based visualization tools . This system aims to continuously collect, analyze, and provide real-time data on the water quality of Algerian rivers and reservoirs. The core components of our system include Node.js for backend development, Streamlit for frontend visualization, a monitoring device equipped with microcontroller and water quality parameters sensors for collecting real-time data, and most importantly an LSTM (Long Short-Term Memory) model for forecasting analysis.

KEYWORDS

Water quality monitoring , IoT technology, Node.js, LSTM models, monitoring devices.

Résumé

La surveillance de la qualité de l'eau est cruciale pour la préservation de l'environnement et la santé publique. En Algérie, comme dans de nombreuses régions à travers le monde, garantir la qualité de l'eau reste une préoccupation majeure. Ce rapport présente un système de surveillance de la qualité de l'eau exploitant la technologie de l'Internet des objets (IoT), l'analyse prédictive et des outils de visualisation basés sur le web. Ce système vise à collecter, analyser et fournir en continu des données en temps réel sur la qualité de l'eau des rivières et des réservoirs algériens. Les composants clés de notre système comprennent Node.js pour le développement backend, Streamlit pour la visualisation frontend, un dispositif de capteur équipé d'un microcontrôleur et de capteurs de paramètres de qualité de l'eau pour la collecte de données en temps réel, et un modèle LSTM (Long Short-Term Memory) pour l'analyse prédictive.

Mots-clés

Surveillance de la qualité de l'eau, Technologie IoT, Node.js, Modèles LSTM, Dispositifs de capteurs.

ملخص

مراقبة جودة المياه أمر بالغ الأهمية لحفظ البيئة وصحة العامة. في الجزائر، مثل العديد من المناطق على الصعيد العالمي، يظل ضمان جودة المياه مصدر قلق كبير. يقدم هذا التقرير نظام مراقبة جودة المياه الذي يستفيد من تقنية الإنترنت للأشياء (IoT) والتحليل التنبؤي وأدوات التصور القائمة على الويب. يهدف هذا النظام إلى جمع وتحليل وتوفير البيانات في الوقت الحقيقي بشكل مستمر عن جودة المياه في الأنهار والسدود الجزائرية. تتضمن المكونات الأساسية لنظامنا Node.js لتطوير الجزء الخلفي ، و Streamlit للتصور الجبهي، وجهاز مستشعر مجهز بوحدة تحكم مصغرة ومستشعرات لمعلمات جودة المياه لجمع البيانات في الوقت الحقيقي، ونموذج LSTM (الذاكرة القصيرة الطويلة الأجل) للتحليل التنبؤي.

الكلمات المفتاحية

مراقبة جودة المياه، تكنولوجيا الإنترنت للأشياء، Node.js ، نماذج LSTM، أجهزة الاستشعار.

Acknowledgement

We would like to extend our heartfelt appreciation to all individuals who have made valuable contributions to the successful culmination of this report. Initially, we express our gratitude to our supervisor, Dr. Ykhlef Hadjer, for her leadership, assistance, and valuable perspectives during the entire process. Her exceptional knowledge and unwavering dedication have played a crucial role in molding this project and guaranteeing its triumphant implementation.

We would like to express our gratitude to the professors and faculty members of the computer science department for their expertise and support throughout the years, which have enabled us to successfully undertake this project. Their profound knowledge and skills in the domains of Physics, Mathematics, and Computer Science have been immensely useful in molding our comprehension of essential topics and elevating the overall caliber of this study.

We like to express our gratitude for the valuable contributions made by the scientific community and the writers of the pertinent literature in the domain of seed sorting and machine learning. Their innovative research and enlightening articles have established a strong basis for this project and have played a crucial role in creating the theoretical framework and methods utilized in this project.

Moreover, we would want to convey our appreciation to our friends and family for their steadfast support, encouragement, and comprehension during this endeavor. Their conviction in our talents and their ongoing motivation have been crucial in conquering hurdles and keeping focused.

Lastly, we would like to thank the University of Blida 1 for providing the necessary resources, facilities, and opportunity for carrying out this study. The project's successful completion was greatly facilitated by the critical institutional assistance it received.

To all those who have helped in various ways, directly or indirectly, we send our heartfelt appreciation. We really appreciate your support and contributions, since they have been essential in the successful preparation of this study.

Thank you.

sebiha massinissa nail & blaili zakaria

Contents

List of Figures

List of Tables

1 INTRODUCTION

Water is the basis of life, as it is considered an essential factor to ensure survival. Water is used in several fields, including medicine, agriculture, and even the building, which makes preserving it and ensuring its cleanliness a global responsibility.

Since Algeria is the biggest country in Africa and lining the Mediterranean Sea makes it depend on various types of water's resources from seawater desalination, groundwater and deep sedimentary to satisfy its needs from drinking, agriculture and industry [1]. However, aquifer pollution is a serious issue and has occurred in some areas and it menaces the water resources mentioned before; for example saline intrusion along the coast relating to over-abstraction, and nitrate pollution from agriculture. Our country is expected to face issues like having limited rainfall and a rising population. This puts strain on water resources, impacting agriculture, industry, and everyday life.

To address this issue, we propose a system that monitors developments occurring in various bodies of water and sources using Internet of Things (IoT). This project intends to create a thorough Water Quality Monitoring System using IoT technologies, a user-friendly dashboard. In addition, our system uses an LSTM (Long Short-Term Memory) model to forecast water quality parameters based on historical data. This system sends a notification to an authorized device when the detected water quality falls short of the predetermined standards, allowing appropriate action to be taken. In short, this system is designed to create a real-time, cost-effective, wireless sensor network and Internet of Things integrated water quality monitoring, raising caution regarding the danger of drinking contaminated water and to encourage individuals to quit polluting it [2].

The rest of this report is structured as follows: The background section explains important concepts for a better comprehension of our water quality monitoring system. The design and implementation part provide details about our water quality monitoring system architecture, deployment, and experimentation. The report concludes with key findings, future research directions, and a list of references.

2 BACKGROUND

This section provides a literature review on the subject matter and some background concepts, exploring topics like water quality parameters, current approaches, anomaly detection, machine learning and Long Short-Term Memory (LSTM). The aforementioned topics should provide a comprehensive understanding of the implemented system.

2.1 Water Quality Monitoring Task

Monitoring water quality plays a crucial role in safeguarding human health and the environment by ensuring the safety of drinking water for communities because of the large number of chemicals used in our everyday lives and in commerce that can make their way into our waters [3]. The primary objectives of water quality monitoring encompass assessing the suitability of water for human consumption, detecting contaminants, and ensuring compliance with regulatory standards [4]. By monitoring the water quality, we can identify specific pollutants, a certain chemical, and the source of the pollution, identify trends, short and long-term and emergencies. This process is essential for protecting public health, maintaining environmental sustainability, and meeting water quality goals set by governments and regulatory agencies [4].

2.2 Water quality parameters

The World Health Organization (WHO) has established water quality standards through its Guidelines for Drinking-water Quality (GDWQ). This latter provides health-based targets and recommendations for a wide range of chemical and microbial contaminants including heavy metals like arsenic, lead, and chromium, as well as microbial contaminants like Escherichia Coli (a bacteria that can cause serious food poisoning) [5].

Water has many properties and when it comes to its quality there are important parameters that should be measured. These parameters include physical, chemical, and biological properties:

The **chemical parameters** include **pH** which is one of the first measurements that should be taken. The pH of water is measured with a simple pH sensor or test kit, which indicates how acidic or basic the water is affecting its taste and ability to corrode pipes. It ranges from 0 and 14 [6]. If we receive a reading equal to 7, this means that the water is *neutral*. Any readings below 7.0 are acidic, while any readings above 7.0 are *alkaline*. Pure water has a neutral pH. However, rainfall is somewhat more acidic and typically has a 5.6 pH. Water is considered to be safe to drink if it has a pH of 6.5-8.5 [6]. Another metric that should be measured is **dissolved oxygen**, which is a critical water quality parameter that can help determine how polluted rivers, lakes, and streams are [6]. When water has a high concentration of dissolved oxygen, the water quality is high. Dissolved oxygen occurs because of the solubility of oxygen. The amount of Dissolved Oxygen in water depends on numerous factors. The primary of which include the water's salinity, pressure, and temperature. It is possible to measure dissolved oxygen levels with a colorimeter or with the electrometric method [6].

The **physical parameters** include **temperature** and it should be taken into consideration since some of the water quality parameters are influenced by it [6]. The ideal water temperatures range from 10-15° C [6]. **Turbidity** refers to how cloudy the water is. Turbidity sensors are designed to measure the ability that light has to pass through water. High levels of turbidity can occur as a result of higher concentrations of silt, clay, and organic materials [6]. The main issue with turbidity in water is that the water will look bad. No one wants to drink cloudy water [6]. **Electrical conductivity measures** how well a sample of water or similar solution can carry or conduct electrical currents. Conductivity levels increase as the amount of ions in the water increases. This is one of the main parameters when measuring water quality because of how easy it is to detect water contamination levels when measuring the conductivity of water. High conductivity means that the water contains a high amount of contaminants. On the other hand, potable water and ultra-pure water are practically unable to conduct an electrical current. The main units of measurement for electrical conductivity include micromhos/cm and milliSiemens/m, the latter of which is abbreviated into mS/m [6].

The **biological parameters** include bacteria, algae, nutrients, and viruses. If the water contains a high number of bacteria, the water soon becomes unsafe, harboring waterborne diseases such as typhoid and cholera [7].

2.3 Current approaches

Traditionally, water quality monitoring relies on manual systems of data collection sampling. Sensors are placed at stations to record data, with a sample taken from each and sent to a laboratory for analysis. It is time consuming, costly and inefficient, creating difficult conditions to compare or assess the various water quality parameters [4].

2.4 Anomaly detection

Anomaly detection is important in water quality monitoring because anomalies that go undetected can lead to incorrect assumptions, insufficient risk assessments, and delayed reactions to possible threats [8].

Anomaly detection is the identification of observations, events or data points that deviate from what is usual, standard or expected [9]. It has a long history in the field of statistics, where analysts and scientists would study charts looking for any elements that appeared abnormal. Today, anomaly detection leverages Artificial Intelligence (AI) and Machine Learning (ML) to automatically identify unexpected changes in a data set's normal behavior [9]. Various anomaly detection methods can be used in building an anomaly detection system such as: Statistical tests can be used by data scientists to detect data anomalies by comparing the observed data with the expected distribution or pattern [9]. These tests include the following methods:

- Z-score: Measures how far a data point is from the mean as a multiple of the standard deviation.
- Interquartile Range (IQR): Based on the range between the first and third quartiles.
- Boxplot: Visual representation of the distribution of data points.
- Histogram: A graphical representation of the distribution of data.

Machine learning algorithms can be used to detect data anomalies by learning the underlying pattern in the data and then identifying any deviations from that pattern [9]. These algorithms can analyze large datasets, recognize complex patterns within data, minimize misidentifying normal instances as anomalies, and adapt to changing conditions [10]. Some of the most common **ML anomaly detection algorithms** include: Decision trees, One-Class Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), Naive Bayesian, Autoencoders, Local Outlier Factor (LOF) and k-means clustering [9].

2.5 Supervised machine learning

Supervised learning is about training an algorithm on a labeled dataset, where the algorithm learns to predict outcomes from input data [11]. And neural network models involve several concepts like model architecture, loss function, gradient, and test stage. A neural network model is a mathematical function that maps input features x which are the **water quality parameters** to an output y which is **the forecasted water quality parameters** denoted as $f{:}x \to y$, it also includes additional model parameters. In the context of neural networks, the loss function measures the disparity between the model's predictions and the true target values (original data), and for this task there are common loss functions used like Mean Absolute Error and Mean Square Error. In the training stage, the gradient descent, which is an optimization algorithm used to update the model parameters in a way that minimizes the loss function. The step size taken during this optimization process is defined by the learning rate. The test stage involves evaluating the performance of a model on a separate dataset that was not used during training [12].

Forecasting involves predicting future numeric values or trends based on historical data, such as using an LSTM model to predict future water quality parameters. In contrast, classification assigns categorical labels to data based on learned patterns, like determining whether an email is 'spam' or 'not spam'. Since the corresponding time series may contain some significant events with lengthy delays and intervals,the design and operating principles of the LSTM are presented in detail in the next section.

2.6 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a specialized type of a sequence model that addresses the issue of learning long-term dependencies [13]. LSTMs are equipped with memory cells controlled by gates that regulate the flow of information, enabling them to selectively retain or discard information over time. This mechanism allows LSTMs to capture intricate patterns in sequential data, making them particularly effective for tasks requiring understanding of order depende

nce, such as speech recognition, and time series forecasting. In the context of water quality monitoring LSTM is used to predict water quality parameters based on historical data. LSTMs surpass at learning from sequences with long-range dependencies [13].

A Architecture

The architecture of LSTM networks is made of LSTMs units (LSTM cell) and each unit is composed of gates: a forget gate, a learn gate and an output gate, dealing with both Long Term Memory (LTM) and Short Term Memory (STM). The following figure illustrate the architecture of an LSTM unit :

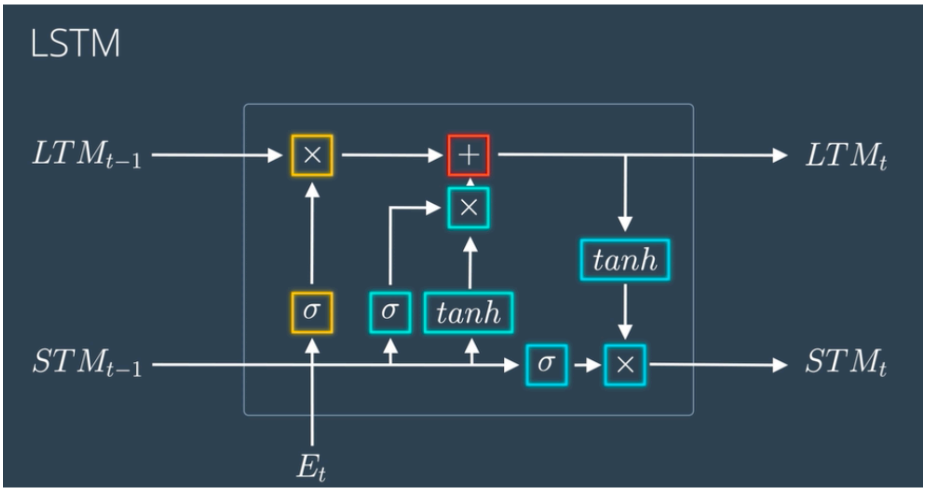


Figure 2.1 : Architecture of an LSTM unit [14].

The Forget Gate: Takes Previous Long Term Memory (LTMt-1) as input and decides on which information should be kept and which to forget [14].

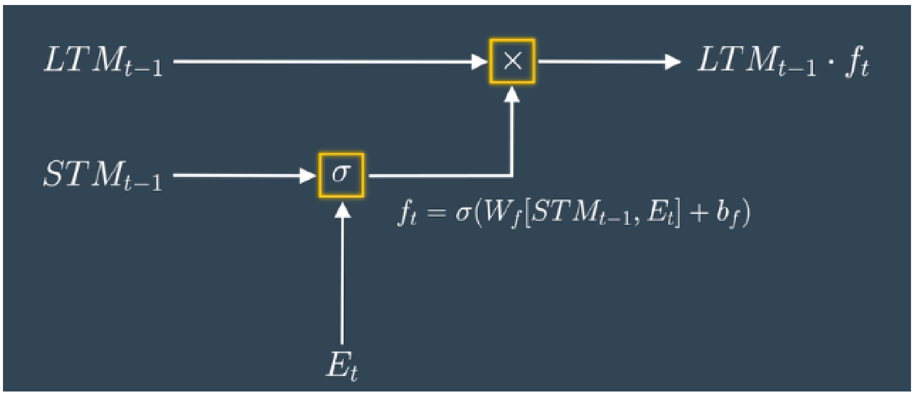


Figure 2.2 : Calculation of the Forget gate of an LSTM unit [14].

Previous Short Term Memory **STMt-1** and Current Event vector **Et** are joined together **[STMt-1, Et]** and multiplied with the weight matrix **Wf** and passed through the Sigmoid activation function with some bias **bf** to form Forget Factor **ft**. Forget Factor ft is then multiplied with the Previous Long Term Memory (**LTMt-1**) to produce forget gate output [14].

Learn gate: Takes Event (Et) and Previous Short Term Memory (STMt-1) as input and keeps only relevant information for prediction [14].

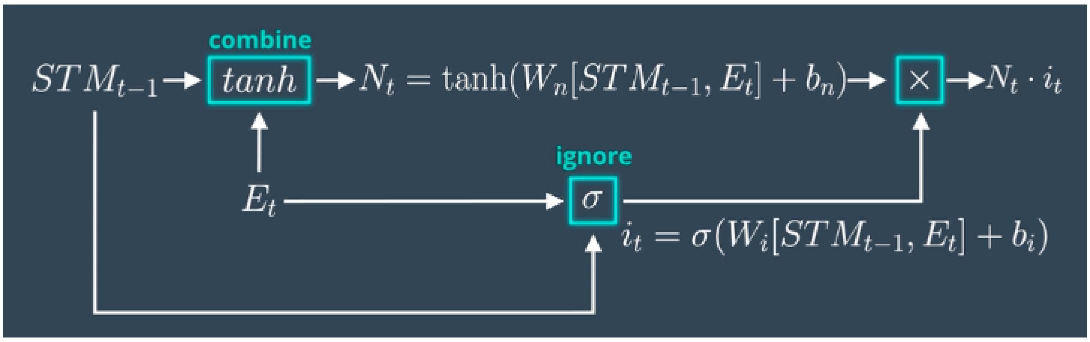


Figure 2.3 : Calculation of the Learn gate of an LSTM unit [14].

Previous Short Term Memory **STMt-1** and Current Event vector Et are joined together **[STMt-1, Et]** and multiplied with the weight matrix **Wn** having some bias **bn** which is then passed to **tanh (hyperbolic Tangent)** function to introduce non-linearity to it, and finally creates a matrix **Nt**. For ignoring insignificant information we calculate one Ignore Factor it, for which we join Short Term Memory **STMt-1** and Current Event vector **Et** and multiply with weight matrix **Wi** and passed through **Sigmoid** activation function with some bias **bi**. Learn Matrix Nt and Ignore Factor it is multiplied together to produce learn gate result [14].

The output of **Forget Gate** and **Learn Gate** are added together to produce an output of Remember Gate which would be **LTM** for the next cell:

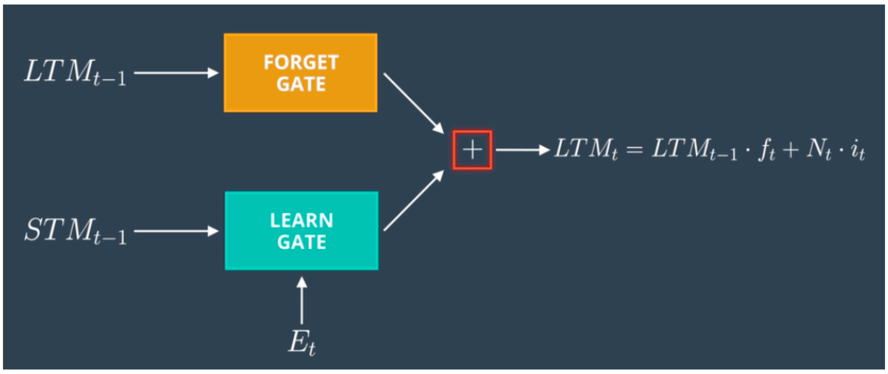


Figure 2.4 : Calculation of the new LTM of an LSTM unit [14]

The output gate: Combine important information from Previous Long Term Memory and Previous Short Term Memory to create STM for next and cell and produce output for the current event [14].

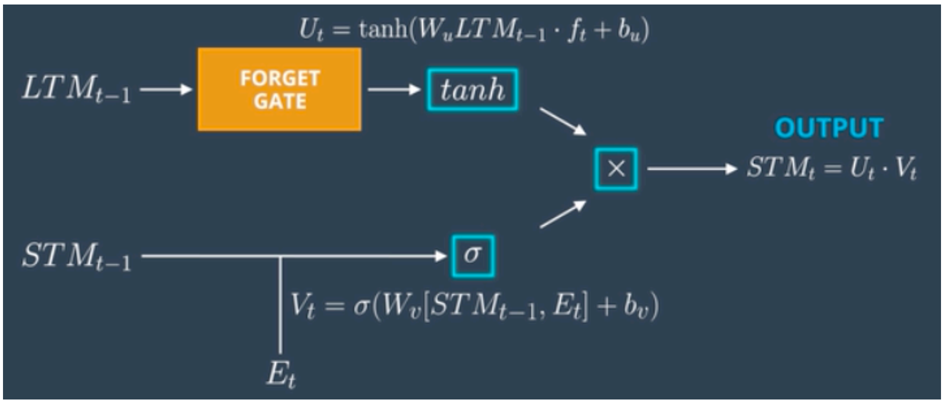


Figure 2.5 : Calculation of the output gate of an LSTM unit [14]

Previous Long Term Memory (**LTM-1**) is passed through **Tangent activation function** with some bias **bu** to produce **Ut**. Previous Short Term Memory (**STMt-1**) and Current Event (**Et**)are joined together and passed through **Sigmoid activation function** with some bias **bv** to produce **Vt**. Output **Ut** and **Vt** are then multiplied together to produce the output which also works as **STM** for the next cell.

B Training LSTM

Training an LSTM involves two main phases: the forward pass and backpropagation

In **the forward pass**, we start by propagating the data inputs to the input layer, go through the hidden layer(s), measure the network's predictions from the output layer, and finally calculate the network error based on the predictions the network made [15]. This network error measures how far the network is from making the correct prediction. For example, if the correct output is 4 and the network's prediction is 1.3, then the absolute error of the network is 4-1.3=2.7. Once the network error is calculated, then the forward propagation phase has ended, the backward pass starts [15].

In the **backward pass**, the flow is reversed so that we start by propagating the error to the output layer until reaching the input layer passing through the hidden layer(s) [15].

The weights and biases of a neural network are the unknowns in the LSTM model. The goal is to determine the values of the weights and biases that achieve the best fit for a given dataset. The best fit is achieved when the losses (i.e., errors) are minimized [16]. At the start of the minimization process, the neural network is seeded with random weights and biases, i.e., we start at a random point on the loss surface. To reach the lowest point on the surface we start taking steps along the direction of the steepest downward slope. This is what the gradient descent algorithm achieves during each training epoch or iteration [16]. At any **nth** iteration the weights and biases are updated as follows:

$$w_i^{n+1} = w_i^{n} - \alpha \nabla_w L \, , \; i = 1 \ldots m \, ,$$

where:

w : the weight.

L: Loss function.

α: Learning rate.

m: the total number of weights and biases in the network.

3 DESIGN AND IMPLEMENTATION

In this section we present the design and implementation of the water quality monitoring system, after learning about the water quality and the existing systems, describing all major components and their roles.

3.1 Design

A System Architecture

The water quality monitoring system consists of four main components: monitoring devices, a central server, forecasting model, and a User-Friendly Dashboard, as shown in **Figure 3.1** . The workflow begins with the monitoring devices collecting real-time data of the water quality parameters. The collected data is transmitted via a communication module to a cloud-based central server. Then it is stored in a time-series database. The forecasting model forecasts the water quality parameters based on the real-time data, it uses a LSTM model. The dashboard visualizes the water quality and its metrics.

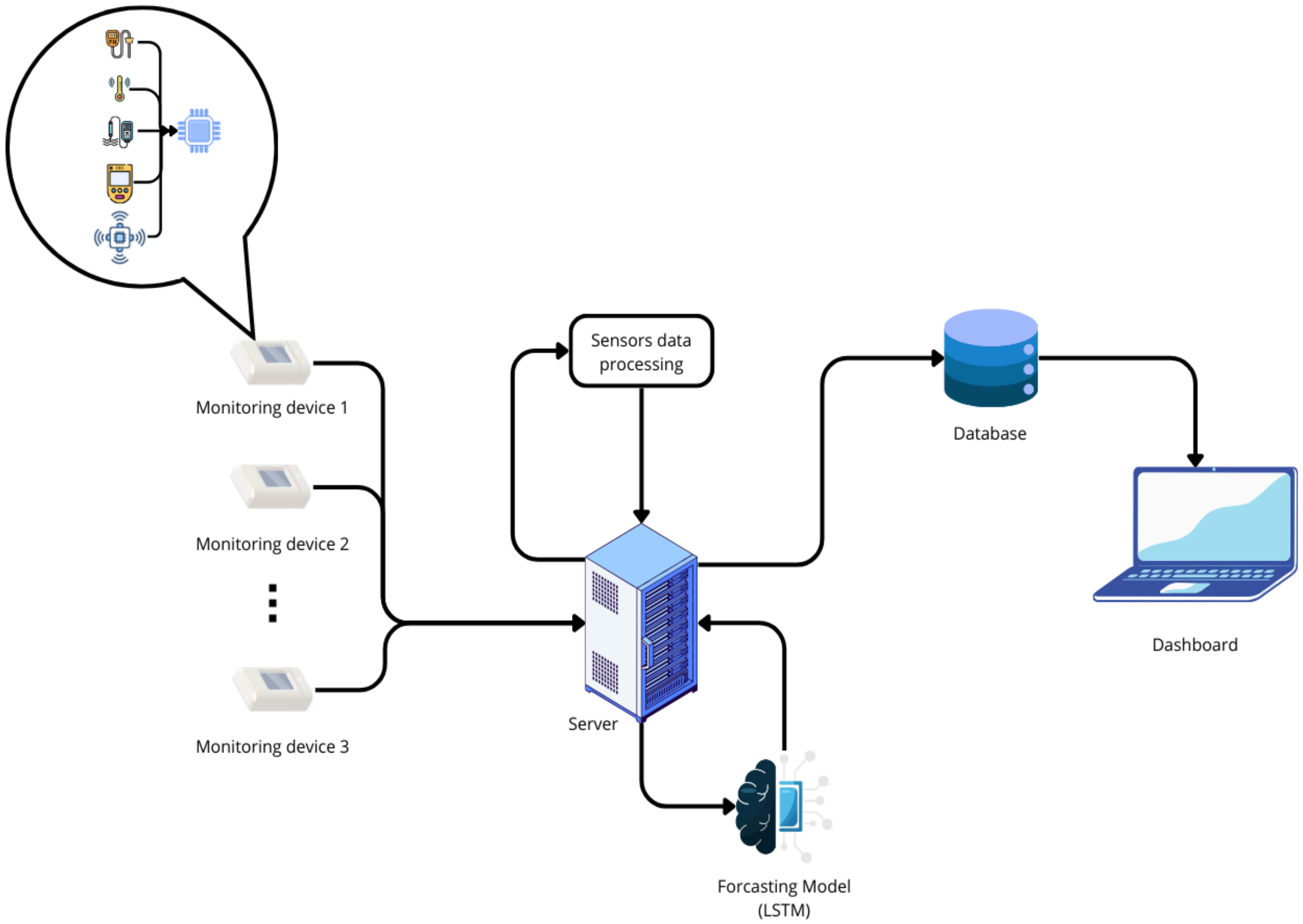


Figure 3.1 : System Architecture of the water quality monitoring system.

B Monitoring device

This is the physical part where each device consists of various water quality sensors and a microcontroller for data processing and communication. The sensors are responsible for measuring vital parameters such as pH, temperature, turbidity, dissolved oxygen, and conductivity.
The devices are deployed in rivers and reservoirs collecting real-time water quality data.

C Server

The server is a computer program that is responsible for receiving and storing incoming data from monitoring devices in a database. It also performs initial data processing tasks such as validation, aggregation, and time-stamping. This component acts as the central hub for managing and analyzing water quality data from multiple reservoirs.

D Database

Database management is crucial for storing and organizing the vast amount of time-series data collected from monitoring devices. Historical water quality data for each reservoir is stored in the database, enabling trend analysis, anomaly detection, and historical comparisons.

E Sensor data processing

The following algorithm processes sensor data for pH, turbidity, conductivity, and temperature to calculate Water Quality Index (WQI). Each parameter is normalized based on expected ranges and weighted according to its significance. The normalized values are then used to compute individual indices, which are used to calculate the final WQI.

```
Algorithm CalculateWQI
Input:pH, Temperature, Turbidity
Output:WQI
// Weights for each parameter
 const ph_weight = 0.33
 const turbidity_weight = 0.33
 const temperature_weight = 0.33
// Step 1: Normalize each parameter
 ph_norm = ((pH - 6) / (8.5 - 6)) * 100
 turbidity_norm = (Turbidity / 5) * 100
 temperature_norm = (Temperature / 30) * 100

// Step 2: Calculate each index
 ph_index = ph_norm * ph_weight
 turbidity_index = turbidity_norm * turbidity_weight
 temperature_index = temperature_norm * temperature_weight

// Step 3: Compute the Water Quality Index (WQI)
 WQI = ph_index + turbidity_index + temperature_index

// Output the Water Quality Index
 return WQI
```

Figure 3.3 : Water quality index interpretation.

F Forecasting model

The forecasting LSTM model plays a crucial role in water quality monitoring. The LSTM model is used to forecast water quality parameters for the next day based on historical data (Today's value). The data for the water quality monitoring system is collected from various devices and stored in the database. For the learning phase, the input data consists of historical water quality measurements. Since we do not have access to real world water bodies we used an external training dataset(that has been collected from USGS NWIS online repository :). This latter gets preprocessed to ensure compatibility with the LSTM model, which includes converting data types, handling missing values, and scaling features. The data is then splitted into training and testing sets based on a specified train ratio(60% for training and 40% for testing). The LSTM model architecture consists of an input layer, LSTM layer, and a dense output layer. The figure below illustrates the process of using an LSTM model to predict a water quality parameter:

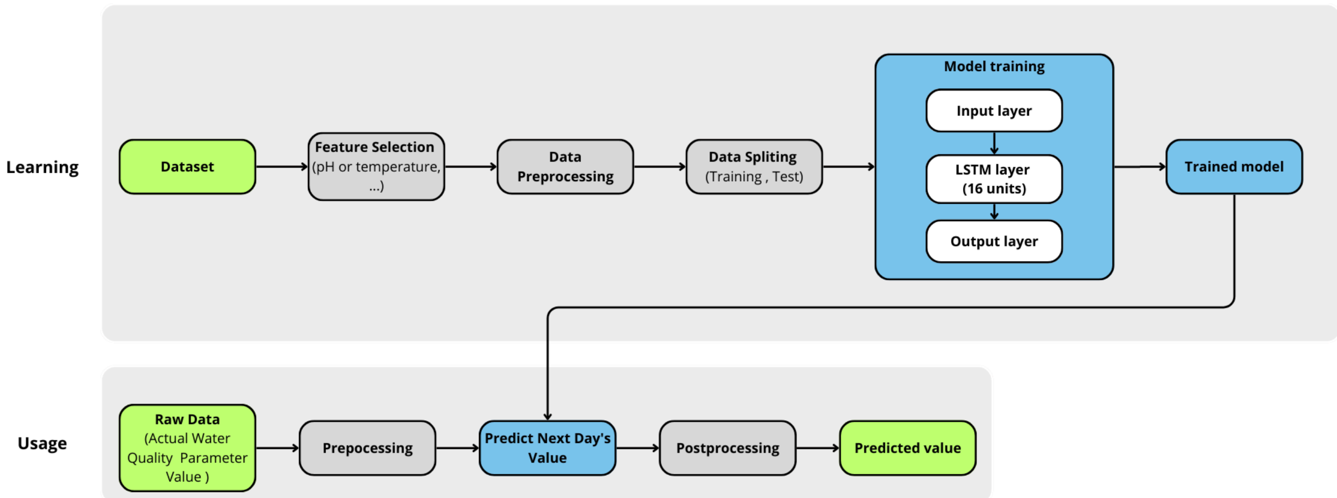


Figure 3.4 : LSTM Learning and Usage for Water Quality Prediction.

- The input layer accepts sequences of historical water quality data recorded at 30-minute intervals.
- The LSTM layer comprises memory cells with gating mechanisms to capture long-term dependencies in sequential data.
- The output layer predicts the next value in the sequence.

Table 3.1 LSTM Model architecture.

Layer (Type)	Output Shape	Parameters	Activation	Initialization
Input Layer	(None, 1, feature_dim)	0	-	-
LSTM	(None, 16)	2,112	ReLU	LeCun Uniform
Dense	(None, 1)	17	-	-

G Dashboard

The dashboard consists of a user-friendly web application for visualizing water quality data in real-time. The web-based dashboard displays interactive charts, graphs, and maps, providing intuitive insights into water quality trends across different reservoirs. Users can customize the data updates rate, create new devices, receive notifications for abnormal water quality conditions, enabling proactive monitoring and management, and use a chatbot to ask any questions .

3.2 Implementation

The figure below shows the water quality monitoring system.

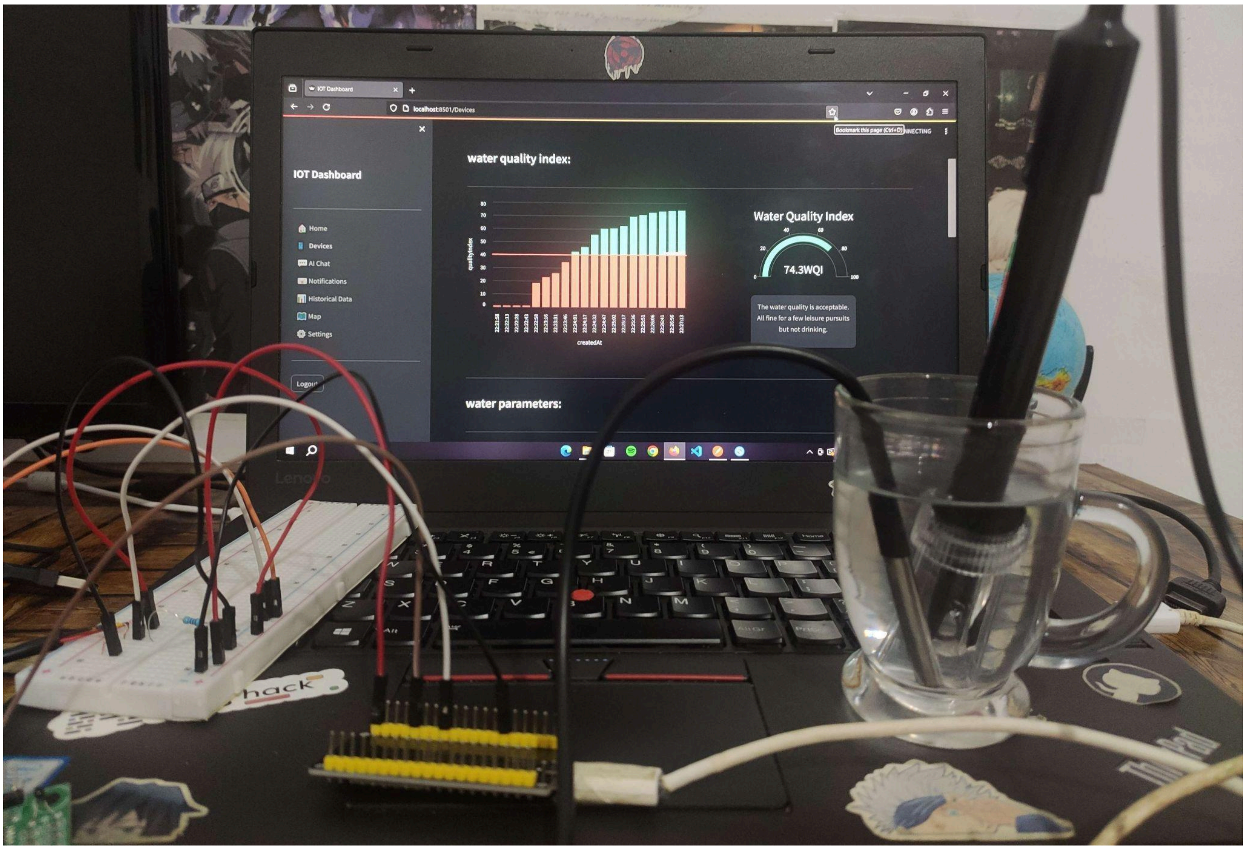


Figure 3.5 : The water quality monitoring system.

A Monitoring Device

The Monitoring device is composed of a microcontroller ESP32 and various water quality sensors: pH, Temperature, and Turbidity. Figure 3.3 illustrates the hardware design of our Monitoring device made with the online diagramming tool draw.io.

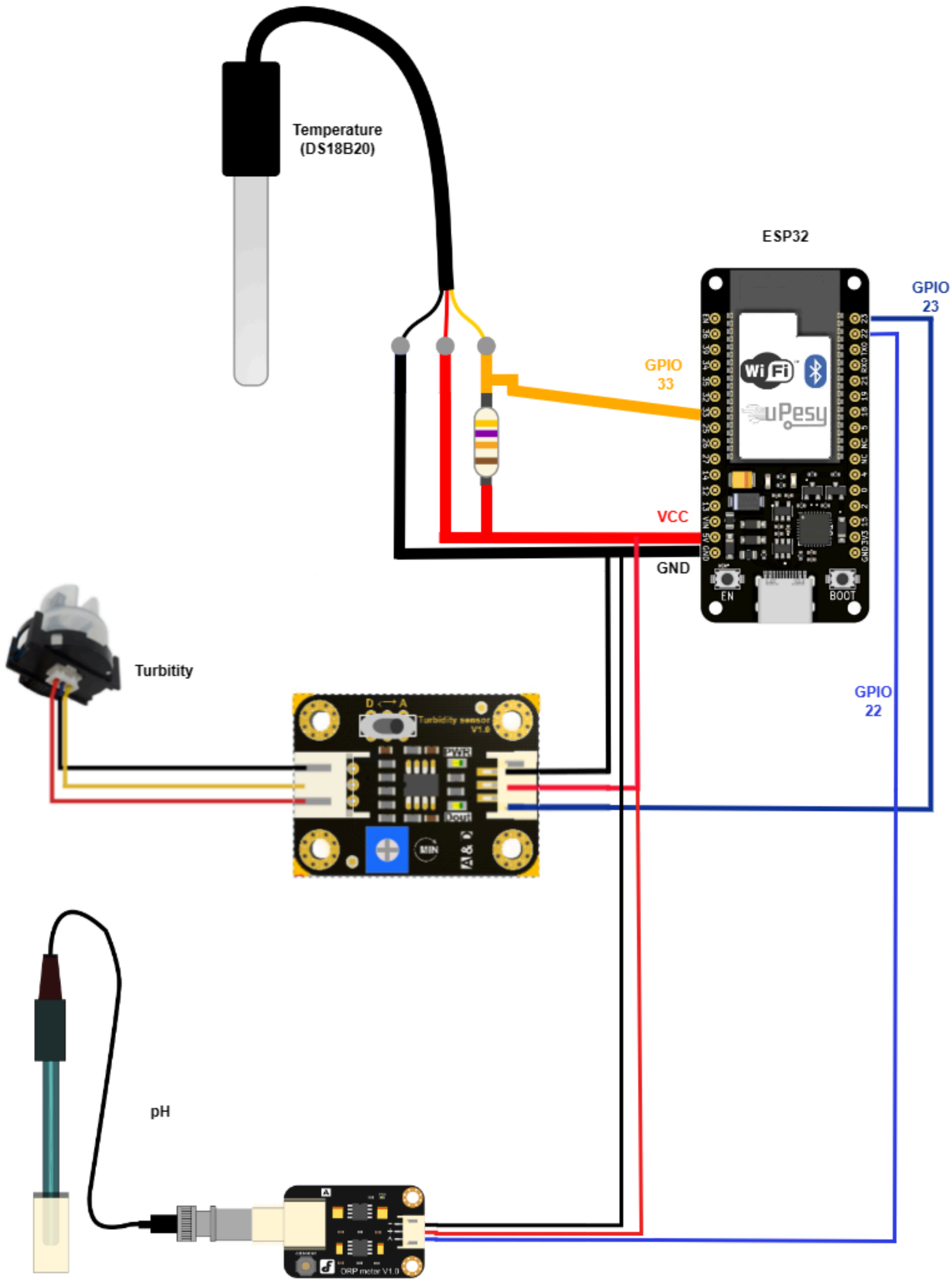


Figure 3.6 : The monitoring device.

The microcontroller (ESP32)

The microcontroller used is an ESP32 onto which a sketch is loaded that gets the data collected by the sensors and sends it to the server.

Why ESP32 and not Arduino Uno? Table 3.2 illustrates a Comparison between ESP32 and Arduino UNO.

Table 3.2 Comparison between ESP32 and Arduino UNO.

Feature	Arduino UNO	ESP32
Uses the c/c++ language :	✓	✓
Has an open-source community :	✓	✓
CPU :	Single core CPU	Dual core CPU
Support wifi :	✗	✓
Support bluetooth :	✗	✓

ESP32 has the same features as Arduino Uno and some developed ones, and for the communication modules that it offers (in our case it is WI-FI) and using it is cheaper than using Arduino Uno plus a communication modules chip (ESP32 or ESP8266).

B Database

We chose MongoDB as the database management system (SGBD) for our Water Quality Monitoring System for several reasons. MongoDB's document-based architecture makes it easier to handle the diverse data collected from the monitoring devices [17]. This includes integrating sensor readings, geographical information, and metadata seamlessly. Its scalability and performance capabilities are crucial for managing the large volumes of data across multiple water bodies. Moreover, MongoDB's features such as change streams and compatibility with various programming languages simplify real-time data accessibility and dashboard development.

The figure below represents the database schema:

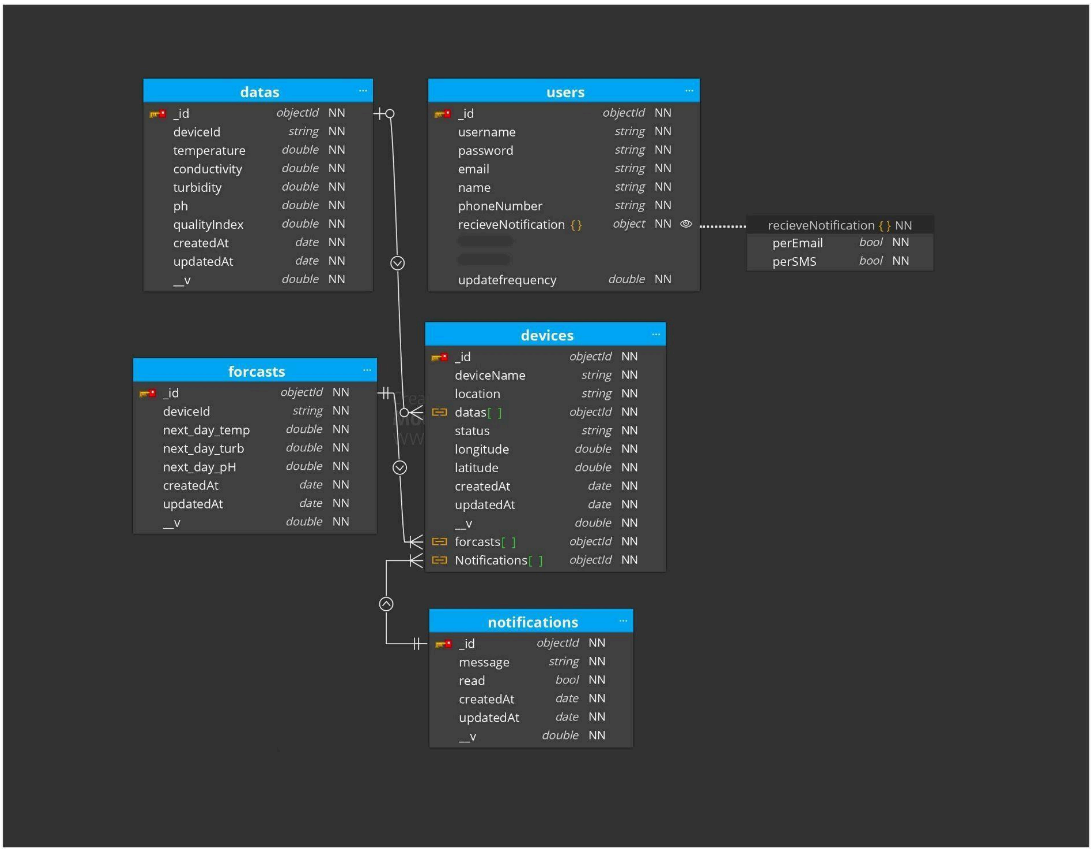


Figure 3.7 : Database schema

This schema categorizes data into many entities, which are identified in MongoDB documents by distinct properties and defines the relationships between these as documents. The documents are categorized into various collections based on their kind and function. There are a total of **five collections** :

- **Data collection :** It saves the data that it gets from each device along with the deviceId so that it can be sorted and sent to the right place.

- **Users collection :** since our system is based on single user management this collection has no relations with the rest and it is only designed to hold the user preferences such as receiving notifications or changing the email/phone number, password.
- **Forecasts collection :** it is made to save the forecasted data for each day by our LSTM model for each device.
- **Notifications collection :** just like the forecasts collection, this was made to store the messages that will be sent by email, phone, or the dashboard.
- **Devices collection :** It is the most important collection because it stores references to other collections in an array of their IDs and links the data together. It also has information about each device, like its name,location, ID and so on.

The relationships assist the arrangement of data entries, forecasts, and notifications with their respective devices. This form of relationship is called **Many-to-One Reference**, and there are three of them. :

- Multiple **datas** records can reference a single device record.
- Multiple **forcasts** records can reference a single device record.
- Multiple **notifications** records can reference a single device record.

C Forecasting model

The forecasting LSTM model's implementation involves translating the design specifications into executable code using Python and the Keras deep learning library where the LSTM layer contains 16 LSTM units, and 100 epochs . The data is loaded from the database. Missing values are handled by dropping rows with NaN values. The features are scaled using MinMaxScaler to normalize the values between -1 and 1. The date column is converted to datetime format and set as the index of the DataFrame. The data is split into training and testing sets with a train ratio equals to 0.6 .

Training and evaluation

The scaled training data is then used to create input sequences (X_train) and output sequences (Y_train) for the LSTM model. Specifically,(X_train) has the shape (n, m, 1), where (n) is the number of training samples, (m) is the number of features (in this case, (m = 1), and the additional dimension is for the LSTM's expected input format. Similarly,(y_train) has the shape (n, 1), matching the number of samples. The model is trained using the Adam optimizer with mean squared error (MSE) as the loss function, with the number of epochs and batch size as hyperparameters (epochs = 100 and batch =1). Early stopping is implemented to prevent overfitting.

We trained three separate LSTM models, each dedicated to predicting a single feature (pH, Temperature, Turbidity). Each trained LSTM model is evaluated on the testing set. The scaled testing data is used to create input (X_test) and output (Y_test) sequences. The model's predictions on the testing set are obtained using the predict method. Model performance is evaluated using metrics such as mean squared error (MSE) and R-squared (R2) score. Table 3.3 shows the results

Table 3.3 Evaluation model performance.

Feature	**MSE** (Train set)	**MSE** (Test set)	**R2** (Train set)	**R2** (Test set)
pH	0.032	0.039	0.739	0.720
Temperature	0.007	0.006	0.973	0.982
Turbidity	0.050	0.066	0.198	0.299

Discussion

The LSTM model performs well for predicting pH and temperature, evidenced by low MSE and high R^2 values, indicating good accuracy and generalization. For pH, MSE slightly increases from the training to the test set, with R^2 values around 0.72, showing stable performance. Temperature predictions are highly accurate with extremely low MSE and R^2 near 0.98, demonstrating excellent fit. However, turbidity predictions are less accurate, with higher MSE and lower R^2 values, suggesting the model struggles to capture its variance,or it is due to higher noise in the turbidity data, more complex underlying patterns that the model fails to capture . Improving turbidity predictions may require more data, additional features, or different model architectures.

Figure below represents a comparison between predicted values and real values :

The X-axis represents the sequence of observations or time steps in the testing set. Each point along this axis corresponds to a specific observation in the test dataset,which can help to visualize how the predictions and true values change over time. while the Y-axis represents the scaled values of the target variable(for both true and predicted values).

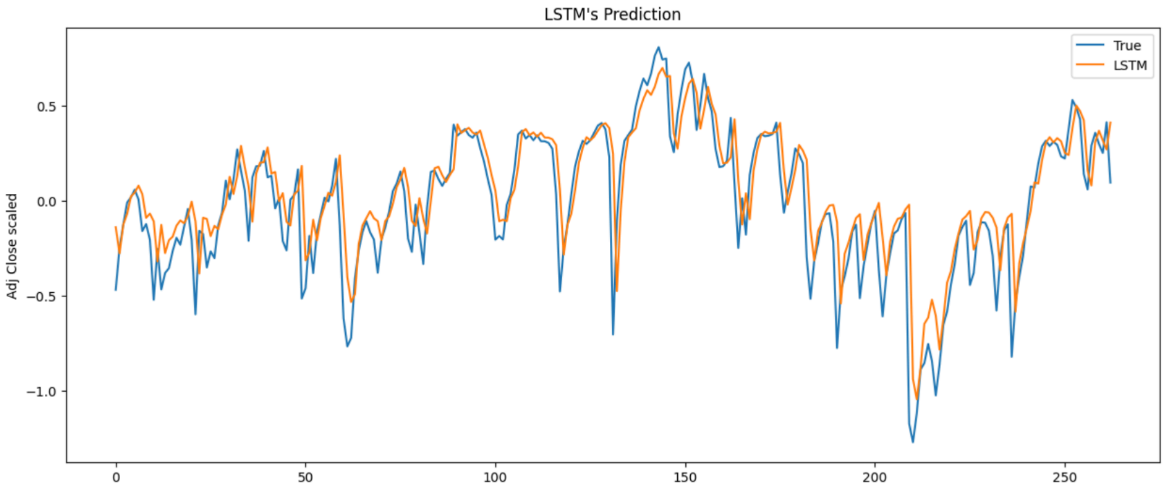


Figure 3.8 : Comparison between predicted values and real pH values.

Discussion

The model's predictions (orange line) closely follow the true pH values (blue line), demonstrating the model's ability to capture the overall trend and fluctuations in pH. Despite minor deviations, particularly at some peaks and troughs, the predicted values align well with the actual values, indicating good performance. This visual representation validates the numerical evaluation metrics, where the model achieved reasonably low MSE and high R^2 values, confirming its effectiveness in predicting pH levels with an adequate degree of precision.

D Dashboard

to develop our dashboard, we have used the following tech :

Streamlit

Streamlit is an open-source Python framework for data scientists and AI/ML engineers to deliver dynamic data apps with only a few lines of code, without needing extensive knowledge of web development, focusing on the functionality and data visualization aspects rather than the layout and design [18]. Streamlit aligns with our goals since our primary focus is on the functionality and visualization of the data rather than complicated web design. Streamlit simplifies the process by abstracting away the complexities of web development, allowing us to concentrate on what matters most: making our data actionable and insightful.

Node.js

Node.js is an open-source, cross-platform JavaScript runtime environment that executes JavaScript code outside of a web browser. It allows developers to use JavaScript to write server-side scripts, enabling the development of scalable and high-performance network applications.

Why Node.js?

- Node.js serves as the backend for our application, handling server-side logic, data processing, and integration with external services and APIs.
- With Node.js, we can implement custom server-side functionalities and business logic required for our dashboard, such as user authentication, data aggregation, and real-time updates.
- Node.js provides excellent support for MongoDB through libraries like Mongoose. This allows seamless integration between our backend (Node.js) and database (MongoDB), simplifying data retrieval, manipulation, and storage operations within our dashboard.

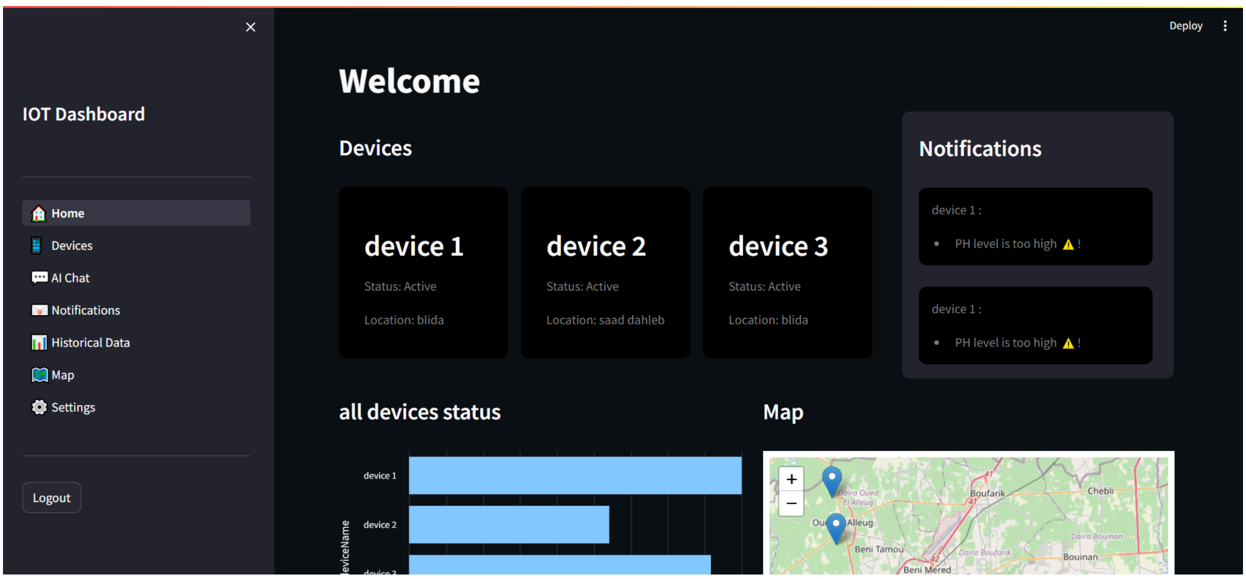
Some interfaces

a. Home Page

The arrangement of the IoT Dashboard's **home page** is simple and effective:

Section of Welcome, Contains a message of welcome at the top.The Devices Section lists all of the linked devices along with their locations and statuses. Notifications Panel (right side), Displays a list of the most recent device-related alerts or notifications.Every Device Status: Contains a bar graph or chart that displays the overall state of every observed water. Map Section, Shows a map with the locations of the devices marked with pins.

This design makes sure that users can easily access navigation options and important information, which makes it easier for them to monitor and control their IoT devices.



<u>Figure 3.9 :</u> The Home page.

b. Device Page

The **device dashboard page** of the IoT Dashboard provides detailed water quality information for all linked devices. This page includes :

Data Analytics, Collected data is shown in a table format, including temperature, turbidity, pH levels, analyte index, and timestamps.Predictions for the next day, Displays predicted pH, turbidity, and temperature values, highlighting variations from earlier readings using color-coded signs for rises and decreases.

More Details Section, it gives the latest collected data of each parameter with corresponding water quality index and graphs .

This part is critical for monitoring real-time water quality measures and identifying patterns based on gathered data and future projections.

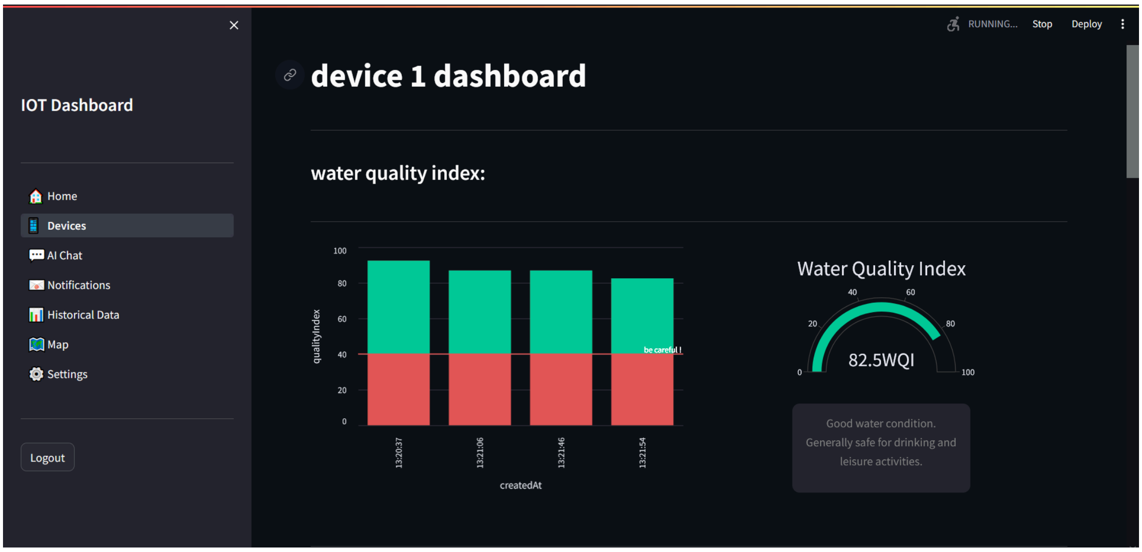


Figure 3.10 : The Device page section 1.

Figure 3.11 : The Device page section 2.

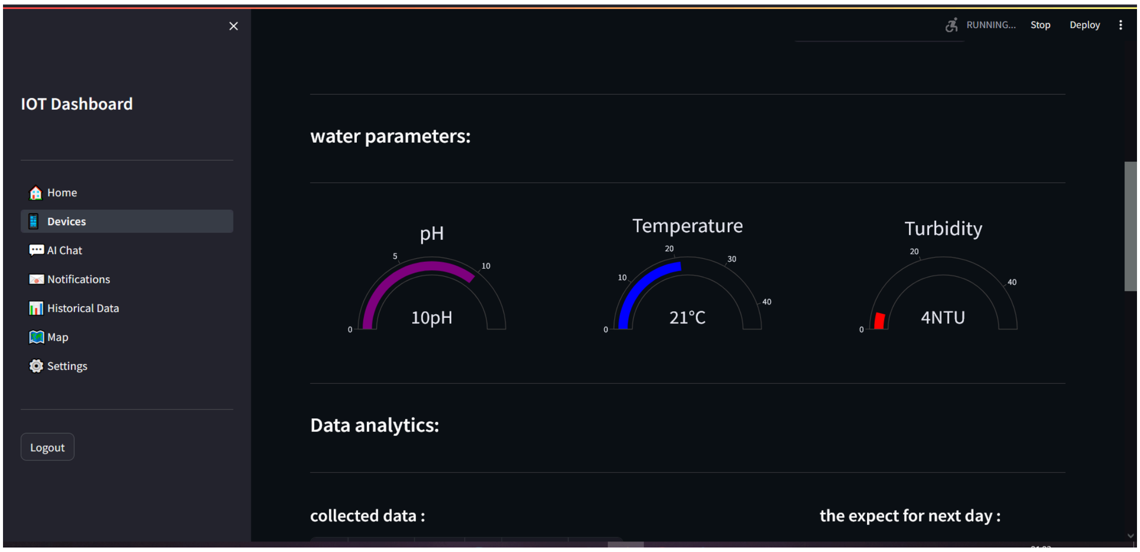
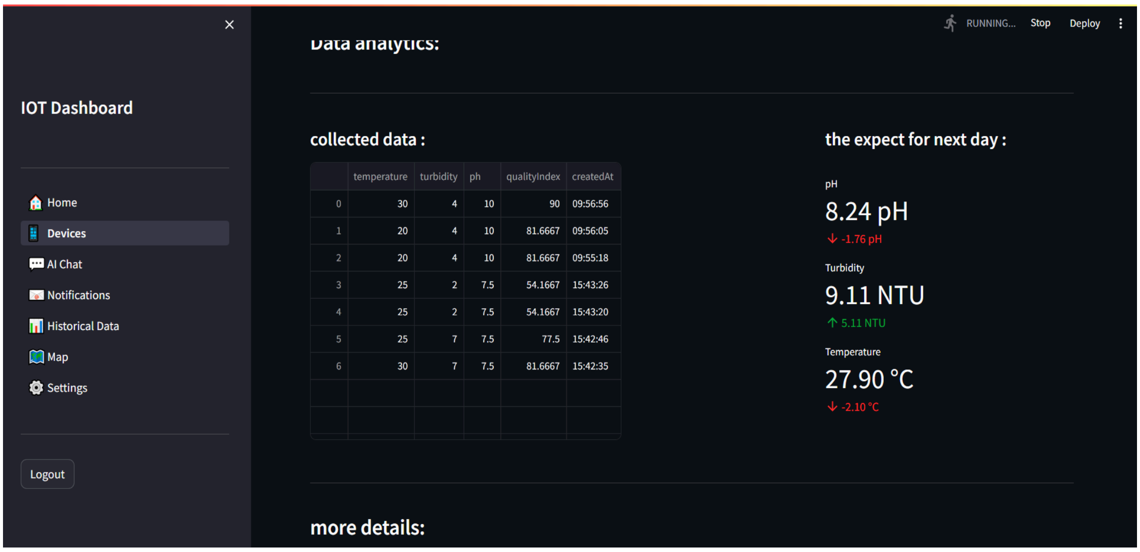


Figure 3.12 : The Device page section 3.



<u>Figure 3.13 :</u> The Device page section 4.

c. AI Chat Page

This **AI Chat tab** of the IoT Dashboard is intended to enable interactive data analysis and visualization. The important components are:

Chat Interface, Users can use messaging to connect with the AI for data-related questions.File Upload, Users can upload CSV files for analysis by dragging and dropping or browsing, with a size restriction set.Dataset selection, A dropdown menu allows users to select a specific device dataset.

This page seeks to improve user interaction with data by integrating chat-based commands and graphical representations, making it easier to examine and comprehend device-generated data.

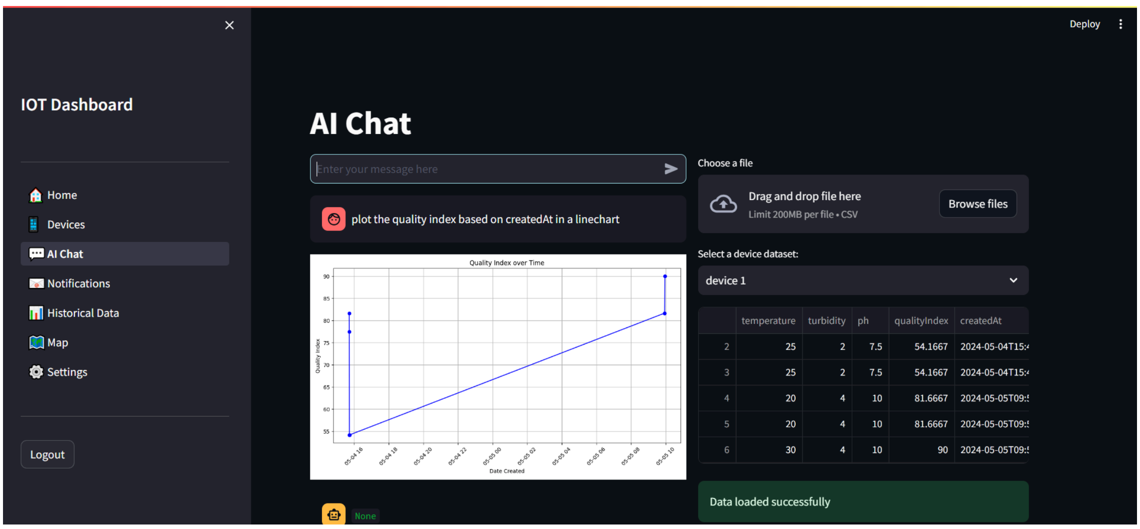


Figure 3.14 : The Chatbot page.

d. Map Page

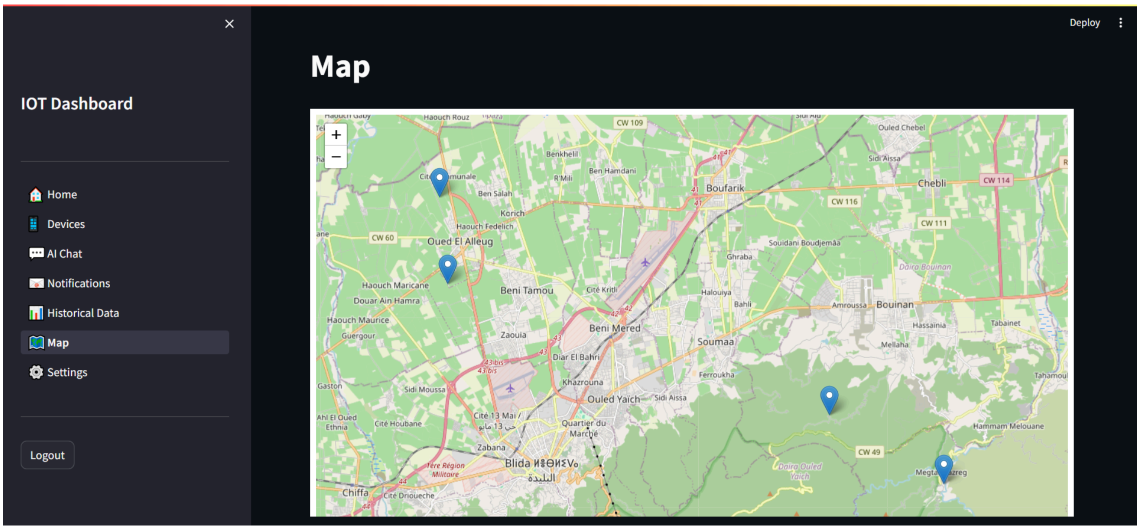


Figure 3.15 : The Map page.

e. Device Configuration Page

In order to give consumers a simple, adaptable, and effective means of connecting their devices to the internet and so improve their overall experience and happiness, the WiFi Configuration tool is essential.

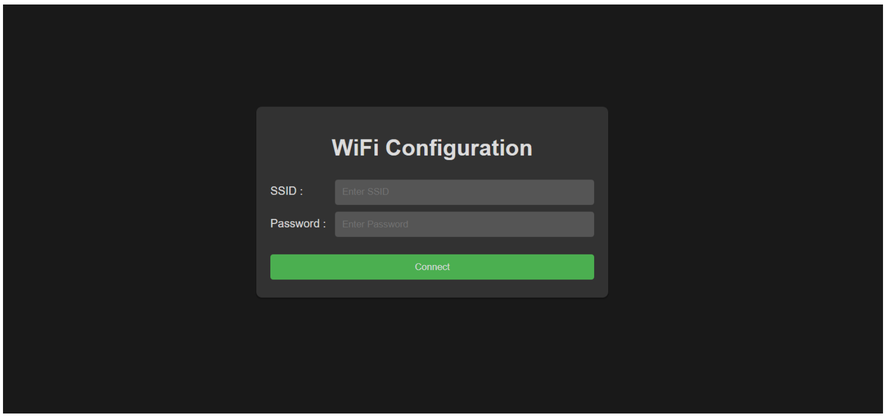


Figure 3.16 : Device configuration page.

E Experiment with final device

This section shows results of different experiments where various substances were added to water samples to observe the changes in water quality parameters. To evaluate the effectiveness of our Water Quality Monitoring System.

Baseline Measurement:

A sample of clean water was taken, and baseline readings of water quality parameters were recorded using the system's sensors as shown in the figure below:

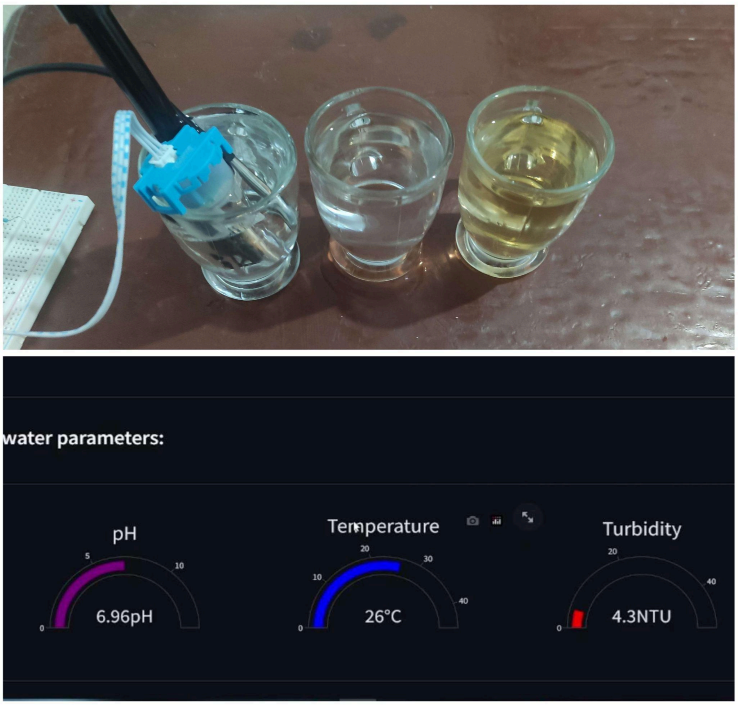


Figure 3.17 : Water quality parameters values of clean water.

1. **Experiment: Adding Vinegar to Water**

   To assess the impact of adding vinegar (acetic acid) on water quality parameters.

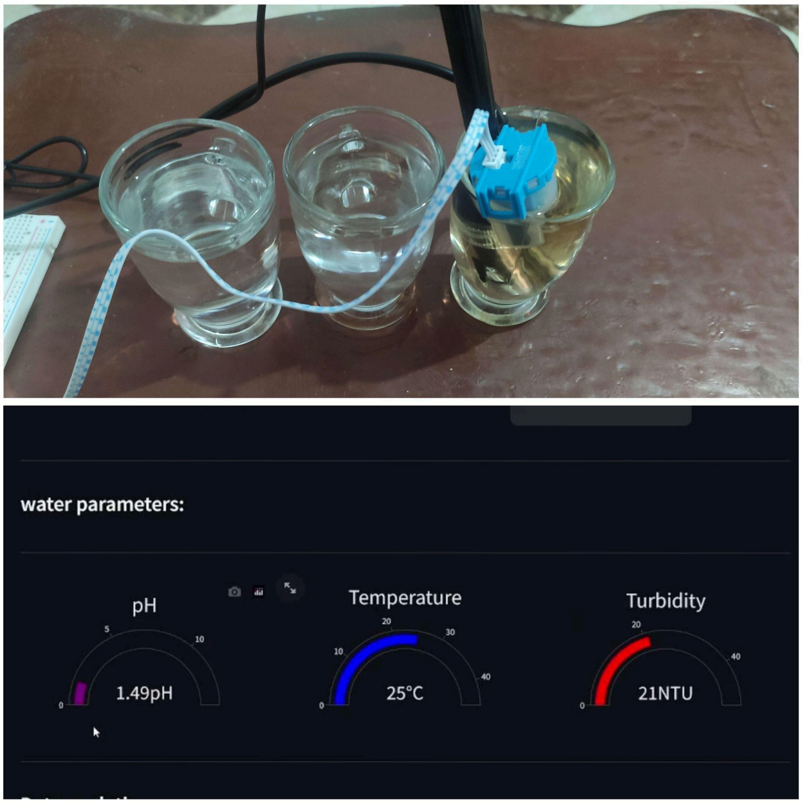
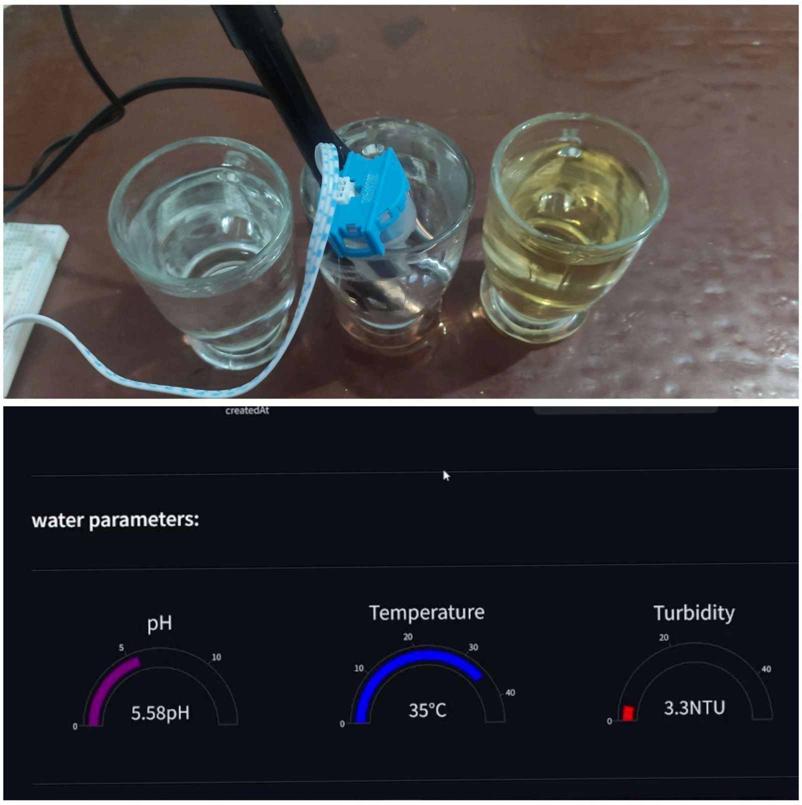


Figure 3.18 : Water quality parameters values after adding vinegar.

2. Experiment: Heating up the Water

To assess the impact of heating up the water on the water quality parameters.



<u>Figure 3.19</u> :Water quality parameters values after heating up the Water.

4 CONCLUSION AND FUTURE WORKS

In this report, we presented a real-time, cost-effective Water Quality Monitoring System that utilizes Internet of Things (IoT) technologies and a user-friendly dashboard to address water pollution issues in Algeria. By integrating a wireless sensor network and real-time monitoring capabilities, the system ensures that water quality standards are maintained, thereby protecting vital water resources from contamination. Additionally, we employed an LSTM model to forecast water quality parameters, enhancing the system's predictive capabilities and enabling proactive measures to be taken. Our findings demonstrate the effectiveness of the proposed system in accurately monitoring water quality, offering a cost-effective and accessible solution for water resource management. This system has the potential to significantly improve water quality monitoring practices, contributing to the sustainability of water resources in regions facing similar challenges.

Based on our findings, we propose several recommendations for future research and actions:

- **Expansion to Additional Water Parameters:** While our system focused on specific water quality indicators, future research can explore the extension to monitor a broader range of parameters, such as heavy metals and organic pollutants. This would require additional sensors and data integration techniques.
- **Robustness and Scalability:** Future research can investigate improving the system's performance under various environmental conditions and expanding its scalability for wider applications. This includes adapting the system for different geographical areas and larger-scale implementations.
- **Integration of Advanced Analytical Models:** To further improve the system's predictive capabilities, integrating advanced machine learning models and combining data from additional sensors, such as satellite imagery and weather data, could provide a more comprehensive understanding of water quality trends and potential threats.

5 REFERENCES

[1] *Water Resources in Algeria*. (2019, July 30). Fanack Water. Retrieved May 29, 2024, from https://water.fanack.com/algeria/water-resources/

[2] *Magtaa Reverse Osmosis (RO) Desalination Plant, Algeria*. (n.d.). Water Technology. Retrieved May 29, 2024, from https://www.water-technology.net/projects/magtaa-desalination/

[3] (2016, August 10). Why monitor water quality? Retrieved May 29, 2024, from https://water.usgs.gov/owq/WhyMonitorWaterQuality.pdf

[4] *Why monitoring water quality is important*. (2021, May 11). Sinay. Retrieved May 29, 2024, from https://sinay.ai/en/why-monitoring-water-quality-is-important/

[5] Fewtrell, L., & Bartram, J. (Eds.). (2001). *Water quality: guidelines, standards & health*. IWA publishing.

[6] O'Donnell, D. (2021, September 20). *Three Main Types of Water Quality Parameters Explained*. Sensorex. Retrieved May 29, 2024, from https://sensorex.com/three-main-types-of-water-quality-parameters-explained/

[7] *3 Main Water Quality Parameters Explained*. (2023, March 13). Atlas Scientific. Retrieved May 29, 2024, from https://atlas-scientific.com/blog/water-quality-parameters/

[8] El-Shafeiy, E., Alsabaan, M., Ibrahem, M. I., & Elwahsh, H. (2023). Real-Time Anomaly Detection for Water Quality Sensor Monitoring Based on Multivariate Deep Learning Technique. *Sensors*, *23*(20), 8613.

[9] *What Is Anomaly Detection?* (2023, December 12). IBM. Retrieved May 29, 2024, from https://www.ibm.com/topics/anomaly-detection

[10] Hope E. (2023) 4 Machine Learning Algorithms For Anomaly Detection. Retrieved May 29, 2024, from (5) 4 Machine Learning Algorithms For Anomaly Detection | LinkedIn

[11] Johnson, D. (2024, March 16). Supervised Machine Learning: What is, Algorithms with Examples. Guru99. Retrieved May 29, 2024, from https://www.guru99.com/supervised-machine-learning.html

[12] Kaur, G., & Bassi, E. (2023, September 23). Training vs. testing data in machine learning. Cointelegraph. Retrieved May 29, 2024, from https://cointelegraph.com/learn/training-vs-testing-data-in-machine-learning

[13] Jain, S. (2023, December 8). Deep Learning | Introduction to Long Short Term Memory. GeeksforGeeks. Retrieved May 29, 2024, from https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/

[14] CHATTERJEE, S. (n.d.). Understanding the LSTM Architecture. Analytics Vidhya. Retrieved May 29, 2024, from https://www.analyticsvidhya.com/blog/2021/01/understanding-architecture-of-lstm/

[15] A Comprehensive Guide to the Backpropagation Algorithm in Neural Networks. (2023, August 22). neptune.ai. Retrieved May 29, 2024, from https://neptune.ai/blog/backpropagation-algorithm-in-neural-networks-guide

[16] Roy, R. (2022, June 14). Neural Networks: Forward pass and Backpropagation | by Ritwick Roy. Towards Data Science. Retrieved May 29, 2024, from https://towardsdatascience.com/neural-networks-forward-pass-and-backpropagation-be3b75a1cfcc

[17] MongoDB Architecture Guide. (n.d.). MongoDB. Retrieved May 29, 2024, from https://www.mongodb.com/resources/products/fundamentals/mongodb-architecture-guide

[18] Singh, T. (2021, October 25). Streamlit: A must learn tool for data Scientist. Medium. Retrieved May 29, 2024, from https://medium.com/crossml/streamlit-2256000541ad