KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A++" Grade
Anand Nagar, Krishnankoil, Srivilliputtur (Via), Virudhunagar (Dt) - 626126, Tamil Nadu | info@kalasalingam.ac.in | www.kalasalingam.ac.in

EXSEL
EXperiential and SErvice Learning

PREDICTION OF WATER QUALITY

A COURSE LEVEL PROJECT REPORT

Submitted by

III-year students of Bachelor of Technology

SARMILA P	**- 9921004640**
AROCKIA SATHIYA DASS J	**- 9921004041**
BPURUSHOTHAM NAIDU	**- 9921005003**
B SATHVIK	**- 9921004056**
BPRUTHVIRAJ	**- 9921005281**

in partial fulfillment of the course of

215EXS3201 / EXSEL – DESIGN-BUILD-OPERATE

Academic Year 2023 – 2024 (Even Semester)

KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A++" Grade

Anand Nagar, Krishnankoil, Srivilliputtur (Via), Virudhunagar (Dt) - 626126, Tamil Nadu | info@kalasalingam.ac.in | www.kalasalingam.ac.in

BONAFIDE CERTIFICATE

Certified that this project report **PREDICTION OF WATER QUALITY** is the bonafide work of **SARMILA P, AROCKIA SATHIYA DASS J, B PURUSHOTHAM NAIDU, B SATHVIK, B PRUTHVIRAJ** who carried out the project work under my supervision.

Faculty Mentor

Submitted for the Project Viva-voce / Review held at Kalasalingam Academy of Research & Education, Krishnankoil on ………………………………

Internal Examiner **External Examiner**

TABLE OF CONTENT

PROBLEM STATEMENT:

PS No.: PS50

Team No.: CC_26

The goal is to create a sophisticated machine learning system to forecast water potability by analyzing various water quality factors. This system will not only determine if water is drinkable but also offer thorough analysis and feedback regarding the water's quality using provided parameters. Additionally, it will help in pinpointing significant pollutants and offering practical suggestions for enhancing water quality. The system should be able to take user inputs for these parameters, predict whether the water is potable or not, and provide feedback on which parameters are within acceptable ranges and which are not.

The aim of this project is to develop a predictive model for water quality parameters in specific water bodies or regions, leveraging machine learning and statistical techniques. The model will analyze historical data and real-time sensor readings to forecast changes in water quality, enabling stakeholders to proactively manage and mitigate potential risks to aquatic ecosystems and human health.

ABSTRACT:

The system is designed to accept user inputs for key water quality parameters, including pH, hardness, solids, sulfate levels, chloramines, conductivity, organic carbon, trihalomethanes, and turbidity. Using these inputs, it predicts the potability of the water and provides detailed feedback on each parameter's compliance with established thresholds. This insightful feedback allows users to gauge the overall water quality accurately, based on scientific standards, thereby empowering them to make well-informed decisions about its potability and suitability for various uses.

Harnessing advanced machine learning and statistical methods, our platform offers a thorough analysis of water quality. We prioritize critical parameters like dissolved oxygen, pH, turbidity, and pollutants, showcasing robust estimation capabilities across various locations and timeframes. Real-time data integration ensures our adaptability to evolving environmental conditions.

The water quality prediction project plans to emphasize these aspects in the problem statement and evaluation plan as an effort toward improved environmental sustainability, public health protection and decision reinforcement within the scope of management over hydrological resources.

Keywords:

User inputs, water quality parameters, pH, hardness, total dissolved solids, chloramines, conductivity, turbidity, predict potability, compliance thresholds, drinking water, machine learning, pollutants, real-time data, accuracy, sustainability, public health, decision-making, resource management.

List of tables:

List of figures:

CHAPTER 1

INTRODUCTION:

Ensuring access to clean and safe drinking water is crucial for public health and sustainable development. However, the task of consistently monitoring and predicting water quality poses significant challenges. Traditional methods often involve labor-intensive and time-consuming processes, which may not always provide timely insights into water safety.

To address these challenges, this project focuses on the development of a sophisticated water quality prediction system. By integrating advanced machine learning techniques, specifically Support Vector Machine (SVM), with practical hardware solutions such as a turbidity sensor, the system aims to enhance the accuracy and efficiency of water quality assessment.

The SVM algorithm has been chosen for its capability to analyze complex datasets and predict water quality with higher precision compared to other models. Concurrently, the turbidity sensor provides real-time measurements, enabling continuous monitoring of water parameters critical for quality assessment.

This combination of SVM-based software analysis and turbidity sensor-based hardware implementation offers a robust and innovative approach to predicting water quality. By providing reliable predictions and actionable insights, this system contributes to improved water management practices and ultimately safeguards the health and well-being of communities.

In this report, we outline the methodology, implementation details, performance evaluation, and the overall impact of our water quality prediction system. This holistic approach underscores our commitment to advancing technology-driven solutions for ensuring clean and safe drinking water worldwide.By facilitating proactive management and timely intervention, our system aims to mitigate risks associated with water contamination, thereby promoting public health and environmental sustainability.

CHAPTER 2

LITERATURE SURVEY:

Reference No	Author	Methodology	Dataset	Performance	Merit	Demerit
1	Yafra Khan,Chai Soo See.Published in: 2016 IEEE (LISAT)	Develop a water quality prediction model using Artificial Neural Network (ANN) and time-series analysis on historical data.	Historical water quality data from 2014, recorded at 6-minute intervals, obtained from the USGS National Water Information System (NWIS).	Evaluate the model's performance using Mean-Squared Error (MSE), Root Mean-Squared Error (RMSE), and Regression Analysis.	Advanced modeling technique (ANN) allows capturing complex non-linear relationships in the data.	Limited to four water quality parameters, potentially missing other significant factors.
2	D. Brindha, V. Puli, B. K. S. NVSS, V. S. Mittakandala and G. D. Nanneboina,Published in 2023 IEEE (ICCMC)	Implement Random Forest regression on water quality parameters like pH, turbidity, dissolved oxygen	Use a dataset with diverse and labeled water quality measurements, including pH, turbidity, dissolved oxygen, and conductivity.	Evaluate using metrics like MAE, MSE, RMSE, and R-squared for model accuracy.	Handles many input variables and captures complex interactions for accurate predictions.	Computationally intensive and less interpretable due to ensemble nature.
3	.K. Abirami, P. C. Radhakrishna and M. A. Venkatesan, Published in 2023 IEEE (CSNT)	Implement and compare K-NN, Naive Bayes, SVM, Decision Tree	Use a dataset with measurements of pH, temperature, conductivity, DO, BOD, nitrate, and total coliform	Evaluate models using accuracy, confusion matrix, precision, recall, and f1-score	Provides a comparative analysis of multiple algorithms to identify the most accurate .	Requires extensive computational resources and time for training

4	A. Sarma and J. S. O .Published in 2023 IEEE (ICRTEC)	Systematic review of remote sensing approaches using multispectral and hyperspectral satellite and airborne imagery	Use multispectral and hyperspectral data from satellite and airborne imagery to extract features	Evaluate remote sensing and AI/ML techniques based on their ability to accurately predict water quality from imagery data.	Provides insights into advanced remote sensing and AI/ML methods for accurate water quality prediction, leveraging extensive data sources.	High complexity and cost associated with acquiring and processing satellite and airborne imagery data.
5	R. I. Singh and U. K. Lilhore.Published in 2023 IEEE (ICCES)	Machine learning and deep learning models for predicting water quality using pre-processing, feature extraction, and classification techniques.	comprising water quality measurements from natural water resources, potentially contaminated by industrial and human activities.	Evaluate models using standard performance metrics such as accuracy, precision, recall, and f1-score.	Provides a comprehensive analysis of both machine learning and deep learning models to determine the most effective method for water quality prediction .	High computational cost and complexity due to the implementation and comparison of multiple advanced models.
6	N. M. Ragi, R. Holla and G. Manju, Published in 2019 IEEE (RTEICT)	Artificial Neural Network (ANN) with Levenberg-Marquardt algorithm to predict water quality parameters	water quality parameters (pH, electrical conductivity, TDS) and the target parameters (alkalinity, chloride, sulphate)	prediction accuracies of 83.94% for chloride, 87.9% for total hardness, 81.736% for sulphate, and 79.48% for total alkalinity.	Cost-effective and efficient compared to traditional chemical methods, reducing time and manpower	Accuracy is still not perfect and may require further optimization and validation to ensure reliability across different water bodies

7	Vinoth Kumar P; Suriya K; Bala Murugan D; R. Reshma Published in 2023 IEEE (ICSCDS)	SAS-MI model with D-CNN for feature engineering and k-means clustering for dataset clustering.	unlabeled dataset related to water quality parameters for training and testing the SAS-MI model.	prediction accuracy, F1-score, and Mean Square Error (MSE).	Combines D-CNN and k-means clustering to provide significant improvements in water quality prediction for smart aquaculture systems.	High computational complexity and resource requirements due to the use of deep learning and clustering techniques.
8	A. L. Lopez, N. A. Haripriya, K. Raveendran, S. Baby and C. V. Priya, Published in 2021 IEEE (IPRECON)	Long Short-Term Memory Neural Network (LSTM NN) and IoT sensors (pH, turbidity, TDS)	daily water quality data from the Muvattupuzha River in Kerala, collected via IoT sensors.	ability to accurately predict future water quality parameters and provide timely notifications.	Provides real-time monitoring and forecasting of water quality, allowing for early detection and prevention of pollution.	Dependent on continuous sensor data and stable IoT connectivity, which may be prone to hardware or network issues.
9	Suma S; Rohit Moon; Mohammed Umer; K. Srujan Raju published in 2023 (ICDCECE)	Decision tree classification models in WEKA to predict clean and contaminated water samples	secondary water quality data collected from the Kenya Water Institute.	five decision tree classifiers: J48, LMT, Random Forest, Hoeffding Tree, and Decision Stump.	identifying water samples that require further analysis, enhancing public health and safety.	Dependent on the qualityand completeness of secondary data, which may limit the model's accuracy and generalizability.

CHAPTER 3

SYSTEM REQUIREMENT&ANALYSIS:

3.1 HARDWARE REQUIREMENT:

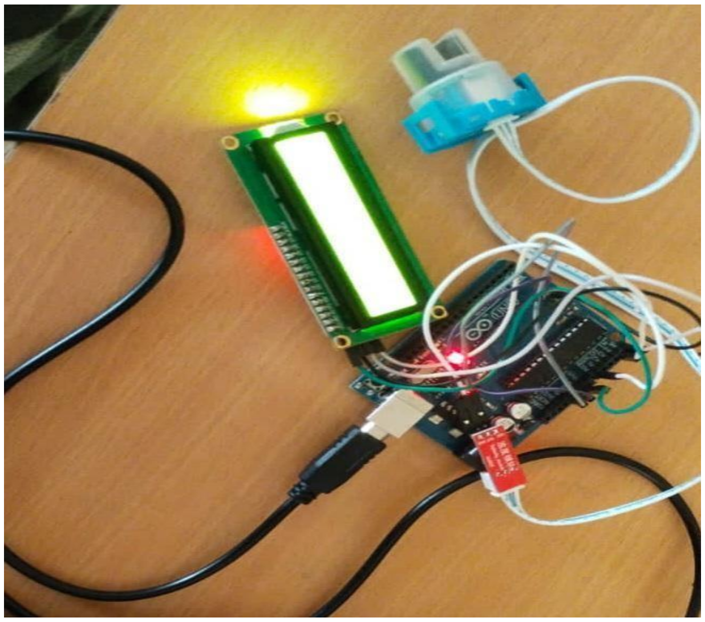


Fig-1:Hardware Implementation

Components Needed:

1. Arduino board (e.g., Arduino Uno)
2. Turbidity sensor (e.g., TSD-10)
3. Additional sensors for other water quality parameters (if needed)
4. Breadboard and jumper wires
5. Power supply (battery or USB cable)
6. SD card module for data logging or a display module for real-time results

Circuit Setup:

1. Connect the turbidity sensor to the Arduino.
2. Connect additional sensors if required.
3. Set up the circuit on a breadboard and ensure proper connections.

Arduino Code:

1. Write a program to read data from the turbidity sensor and other sensors.
2. Process the sensor data to determine water quality.

Data Processing and Prediction:

1. Use a simple algorithm to predict water quality based on sensor readings.
2. Optionally, send the data to a server or a machine learning model for more complex predictions.

Uploading Code and Testing:

1. Connect your Arduino to computer using a USB cable.
2. Open the Arduino IDE, paste the code, select the correct board and port, and upload the code.
3. Open the Serial Monitor (Tools > Serial Monitor) to see the turbidity readings and water quality predictions.

3.2 SOFTWARE REQUIREMENT:

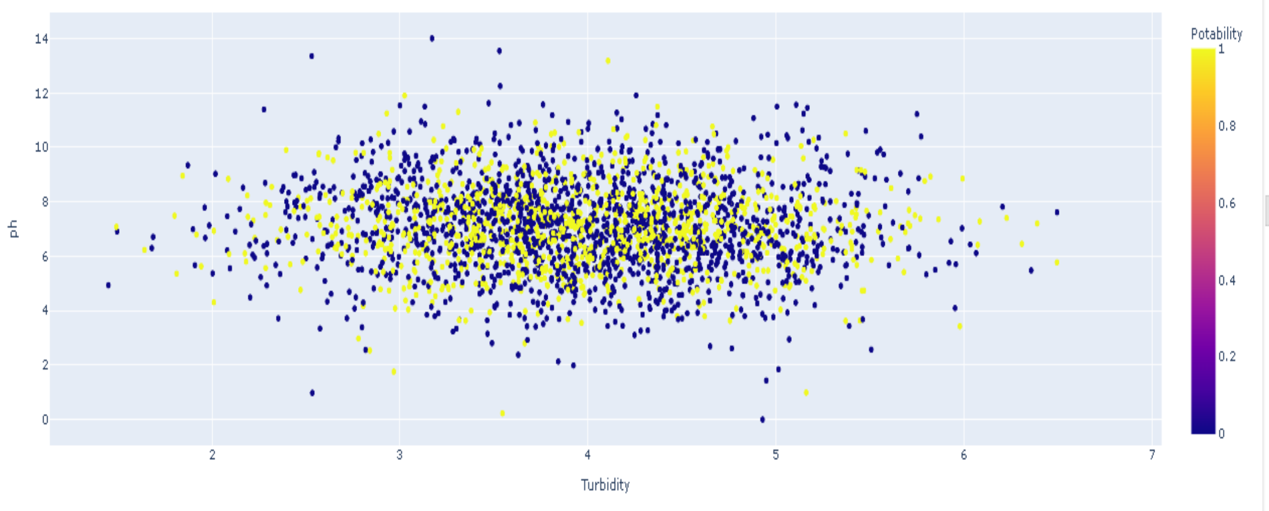


Fig-2:Software Implementation

Data Collection and Preprocessing:

1. Gather data on water quality parameters (pH, hardness, solids, sulfate, chloramines, conductivity, organic carbon, trihalomethanes, turbidity, etc.).
2. Clean the data: handle missing values, remove outliers, normalize or standardize the data if necessary.

Feature Selection:

1. Identify which parameters are most relevant to predicting water quality.
2. Use statistical methods or machine learning techniques to select the most important features.

Model Selection:

1. Choose an appropriate machine learning model for regression or classification (depending on whether you're predicting a continuous quality score or classifying water as "safe" or "unsafe").
2. Common models include linear regression, decision trees, random forests, support vector machines (SVM), and neural networks.

Model Training:

1. Split your data into training and testing sets.
2. Train the model using the training data.
3. Validate the model using the testing data and tune hyperparameters as needed.

Model Evaluation:

1. Evaluate the model's performance using metrics such as mean absolute error (MAE), mean squared error (MSE), accuracy, precision, recall, and F1 score.
2. Ensure the model generalizes well to new, unseen data.

3.3 EXISTING METHODOLOGY:

LOGISTIC REGRESSION:

Logistic Regression is a powerful statistical technique commonly used for binary classification tasks, where the outcome variable has two categorical levels such as Yes/No or 1/0. It works by transforming the linear combination of input features into probabilities using the sigmoid function, ensuring that predictions range between 0 and 1. This method assumes a linear relationship between the log-odds of the outcome and the input features, making it suitable for scenarios where the relationship is likely to be linear or can be approximated as such.

One of the strengths of Logistic Regression is its ability to handle regularization techniques like L1 (Lasso) or L2 (Ridge) regularization. Regularization helps prevent overfitting by penalizing large coefficients, thereby improving the model's generalization performance on unseen data.

FORMULA:

1. Linear Combination of Predictors: z=β0+β1X1+β2X2+⋯+βnXnz = \beta_0 + \beta_1X_1 + \beta_2X_2 + \cdots + \beta_nX_nz=β0+β1X1+β2X2+⋯+βnXn This is a linear equation where β0\beta_0β0 is the intercept and β1,β2,…,βn\beta_1, \beta_2, \ldots, \beta_nβ1,β2,…,βn are the coefficients of the predictor variables X1,X2,…,XnX_1, X_2, \ldots, X_nX1,X2,…,Xn.

2. Logit Function: The term zzz is transformed by the logit function: P(Y=1∣X)=11+e−zP(Y=1|X) = \frac{1}{1 + e^{-z}}P(Y=1∣X)=1+e−z1 The logit function maps any real-valued number into the (0, 1) interval, making it suitable for representing probabilities.

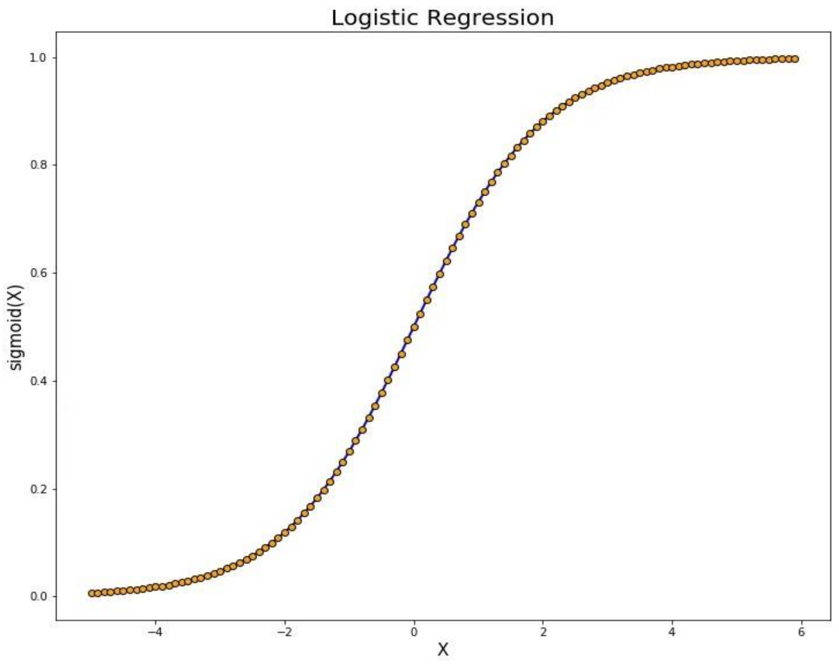


Fig-3:Logistic Regression Representation

ADA BOOST:

One of AdaBoost's key features is its ability to enhance performance by combining the predictions of multiple weak learners. By sequentially focusing on difficult-to-classify samples, AdaBoost iteratively improves its ability to generalize and accurately predict outcomes on unseen data. This approach is particularly effective in handling both simple and complex relationships within the data, allowing AdaBoost to adapt to various types of classification tasks.

In practical terms, if an AdaBoost model achieves an accuracy of 67.38%, it indicates that the ensemble method correctly predicts the binary outcome for approximately 67.38% of the dataset. This accuracy metric underscores AdaBoost's capability to leverage the strengths of individual weak learners and enhance overall predictive performance. AdaBoost's versatility and effectiveness make it a valuable tool in machine learning for tasks where improving classification accuracy is paramount.

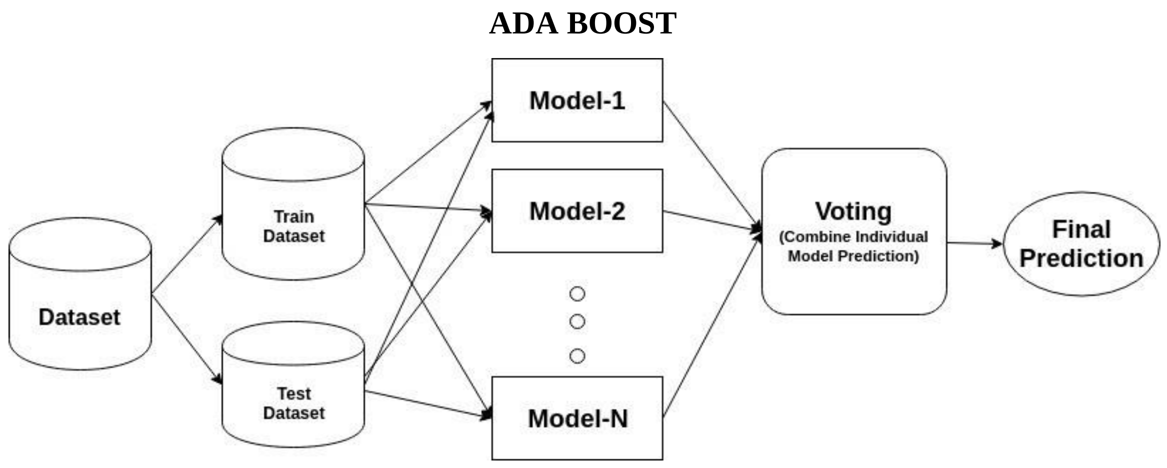


Fig-4:Ada Boost Representation

FORMULA:

Initialize Weights:

Each training sample is initially assigned equal weight:

$$w\_i(1)=1/N$$

where w_i^(1) is the initial weight for the i-th training sample and N is the total number of training samples.

Compute Weak Classifier Error:

For each weak classifier h_t compute the error:

$$\epsilon\_t=\sum i=1N\ w\_i\hat{}(t)I(yi\neq ht(xi))$$

where ϵ_t is the error of the t-th weak classifier, w_i^(t) is the weight of the i-th sample at iteration t, y_i is the true label,h_t(x_i) is the predicted label, and I is the indicator function.

Compute Classifier Weight:

Calculate the weight of the weak classifier based on its accuracy:

$$\alpha t=1/2\ln(1-\epsilon\_t/\epsilon\_t)$$

Update Weights:

Adjust the weights of the training samples based on the error of the weak classifier:

$$w\_i\hat{}(t+1)=w\_i\hat{}(t)\exp(-\alpha\_ty\_ih\_t(x\_i))$$

Normalize the weights:

$$w\_i\hat{}(t+1)=w\_i\hat{}(t+1)/\Sigma j=1Nw\_j\hat{}(t+1)$$

DECISION TREE:

Decision Tree is a popular supervised machine learning algorithm used for both classification and regression tasks. It is particularly useful in predicting water quality because of its simplicity, interpretability, and ability to handle complex, non-linear relationships among variables.

A decision tree model splits the dataset into subsets based on the value of input features. This process is recursive, forming a tree-like structure where each node represents a decision rule based on a feature, and each branch represents the outcome of the decision. The leaves of the tree represent the predicted output.

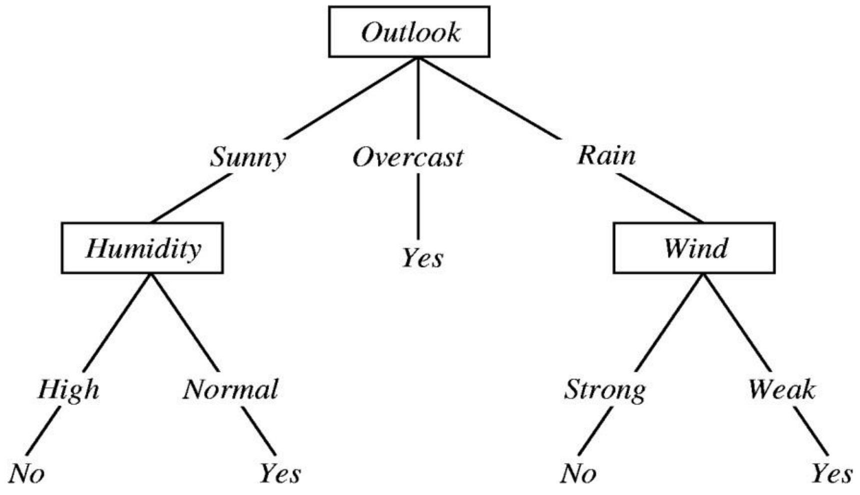


Fig-5:Decision Tree Representation

FORMULA:

Gini Impurity:

Gini(t)=1−Σi=1n(pi)^2

where pi is the probability of a sample belonging to class i at node t.

Information Gain (Entropy):

Entropy(t)=−Σi=1 n pilog 2(pi)

Gain(S,A)=Entropy(S)−Σv∈Values(A)∣Sv|/∣S∣Entropy(Sv)

where S is the current dataset, A is a feature, v is a value of feature A, and Sv is the subset of S for which feature A has value v..

Information Gain Ratio:

GainRatio(S,A)=Gain(S,A)/SplitInformation(A)

SplitInformation(A)=−Σv∈Values(A)|Sv|\|S|log 2(|Sv|\|S|)

Classification Error:

Error(t)=1−max (pi)

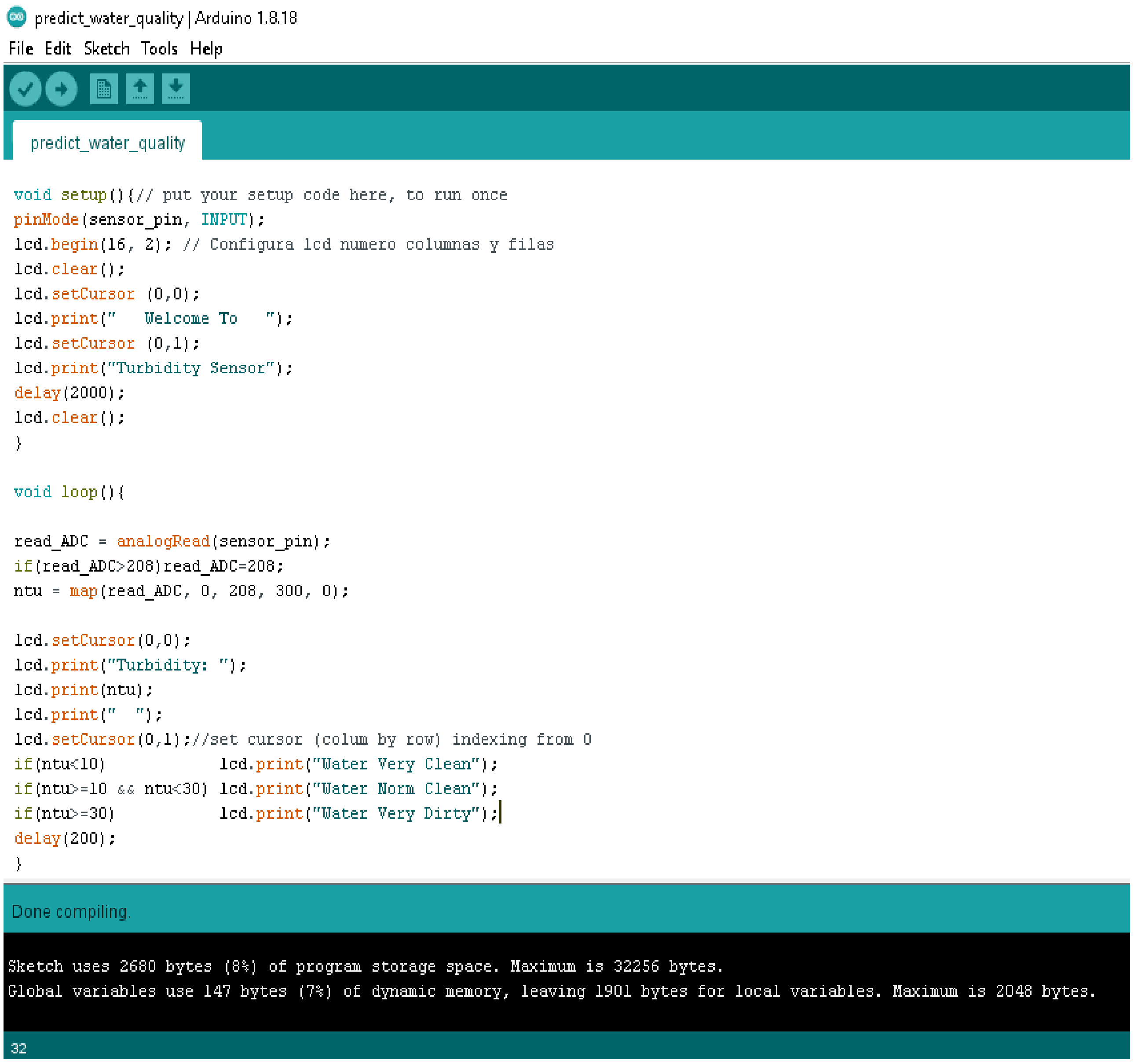
Where pi is the proportion of instances of class i in node t.

CHAPTER 4

SYSTEM IMPLEMENTATION:

4.1 CIRCUIT DIAGRAM&IT'S EXPLANATION:

HARDWARE:


predict_water_quality | Arduino 1.8.18
File Edit Sketch Tools Help
predict_water_quality

```
void setup(){// put your setup code here, to run once
pinMode(sensor_pin, INPUT);
lcd.begin(16, 2); // Configura lcd numero columnas y filas
lcd.clear();
lcd.setCursor (0,0);
lcd.print("   Welcome To   ");
lcd.setCursor (0,1);
lcd.print("Turbidity Sensor");
delay(2000);
lcd.clear();
}

void loop(){

read_ADC = analogRead(sensor_pin);
if(read_ADC>208)read_ADC=208;
ntu = map(read_ADC, 0, 208, 300, 0);

lcd.setCursor(0,0);
lcd.print("Turbidity: ");
lcd.print(ntu);
lcd.print("  ");
lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0
if(ntu<10)             lcd.print("Water Very Clean");
if(ntu>=10 && ntu<30) lcd.print("Water Norm Clean");
if(ntu>=30)            lcd.print("Water Very Dirty");
delay(200);
}
```

Done compiling.

```
Sketch uses 2680 bytes (8%) of program storage space. Maximum is 32256 bytes.
Global variables use 147 bytes (7%) of dynamic memory, leaving 1901 bytes for local variables. Maximum is 2048 bytes.
```

32

Fig-6:Arduino Code Implementation

Wiring Connections:

1. Connecting the 16x2 LCD Display to Arduino Uno:

- VSS (Pin 1)
    - Connect to GND on the Arduino.
- VDD (Pin 2)
    - Connect to 5V on the Arduino.
- VO (Pin 3)
    - Connect to the middle pin of the 10k potentiometer.
    - The other two pins of the potentiometer connect to 5V and GND.
- RS (Pin 4)
    - Connect to Digital Pin 12 on the Arduino.
- RW (Pin 5)
    - Connect to GND on the Arduino.
- E (Pin 6)
    - Connect to Digital Pin 11 on the Arduino.
- D4 (Pin 11)
    - Connect to Digital Pin 5 on the Arduino.
- D5 (Pin 12)
    - Connect to Digital Pin 4 on the Arduino.
- D6 (Pin 13)
    - Connect to Digital Pin 3 on the Arduino.
- D7 (Pin 14)
    - Connect to Digital Pin 2 on the Arduino.
- A (Pin 15)
    - Connect to 5V on the Arduino through a 100R resistor.
- K (Pin 16)
    - Connect to GND on the Arduino.

2. Connecting the Turbidity Sensor to Arduino Uno:

- VCC (Red Wire)
    - Connect to 5V on the Arduino.
- GND (Black Wire)
    - Connect to GND on the Arduino.
- A0 (Yellow Wire)
    - Connect to Analog Pin A0 on the Arduino.

3. Connecting the 10k Potentiometer:

- One Pin
    - Connect to 5V on the Arduino.
- Middle Pin
    - Connect to VO (Pin 3) of the LCD.
- Other Pin
    - Connect to GND on the Arduino.

4. Connecting the 100R Resistor:

- One End
    - Connect to A (Pin 15) of the LCD.
- Other End
    - Connect to 5V on the Arduino.

5. Connecting the Battery Clip:

- Battery Clip's Positive Wire
    - Connect to Vin on the Arduino.
- Battery Clip's Negative Wire
    - Connect to GND on the Arduino.

Assembly Instructions:

1. Prepare the Breadboard:
   - Place the Arduino Uno on the breadboard.
   - Place the 16x2 LCD display on the breadboard.
   - Place the turbidity sensor module nearby for easy connection.
2. Connect the LCD Display:
   - Connect VSS (Pin 1) to GND.
   - Connect VDD (Pin 2) to 5V.
   - Connect VO (Pin 3) to the middle pin of the 10k potentiometer.
   - Connect RS (Pin 4) to Digital Pin 12.
   - Connect RW (Pin 5) to GND.
   - Connect E (Pin 6) to Digital Pin 11.
   - Connect D4 (Pin 11) to Digital Pin 5.
   - Connect D5 (Pin 12) to Digital Pin 4.
   - Connect D6 (Pin 13) to Digital Pin 3.
   - Connect D7 (Pin 14) to Digital Pin 2.
   - Connect A (Pin 15) to 5V through a 100R resistor.
   - Connect K (Pin 16) to GND.
3. Connect the Turbidity Sensor:
   - Connect VCC (Red Wire) to 5V.
   - Connect GND (Black Wire) to GND.
   - Connect A0 (Yellow Wire) to Analog Pin A0.
4. Connect the 10k Potentiometer:
   - Connect one pin to 5V.
   - Connect the middle pin to VO (Pin 3) of the LCD.
   - Connect the other pin to GND.
5. Connect the 100R Resistor:
   - Connect one end to A (Pin 15) of the LCD.
   - Connect the other end to 5V.
6. Connect the Battery Clip:
   - Connect the positive wire to Vin.

    - Connect the negative wire to GND.
7. Upload the Code:
    - Connect the Arduino to your computer using a USB cable.
    - Open the Arduino IDE, paste the provided code, and upload it to the Arduino.
8. Power the System:
    - Use the 9V battery to power the Arduino when not connected to the computer.
9. Test the Setup:
    - Place the turbidity sensor in water.
    - The LCD should display the turbidity readings and corresponding water quality messages.

ARCHITECTURE:

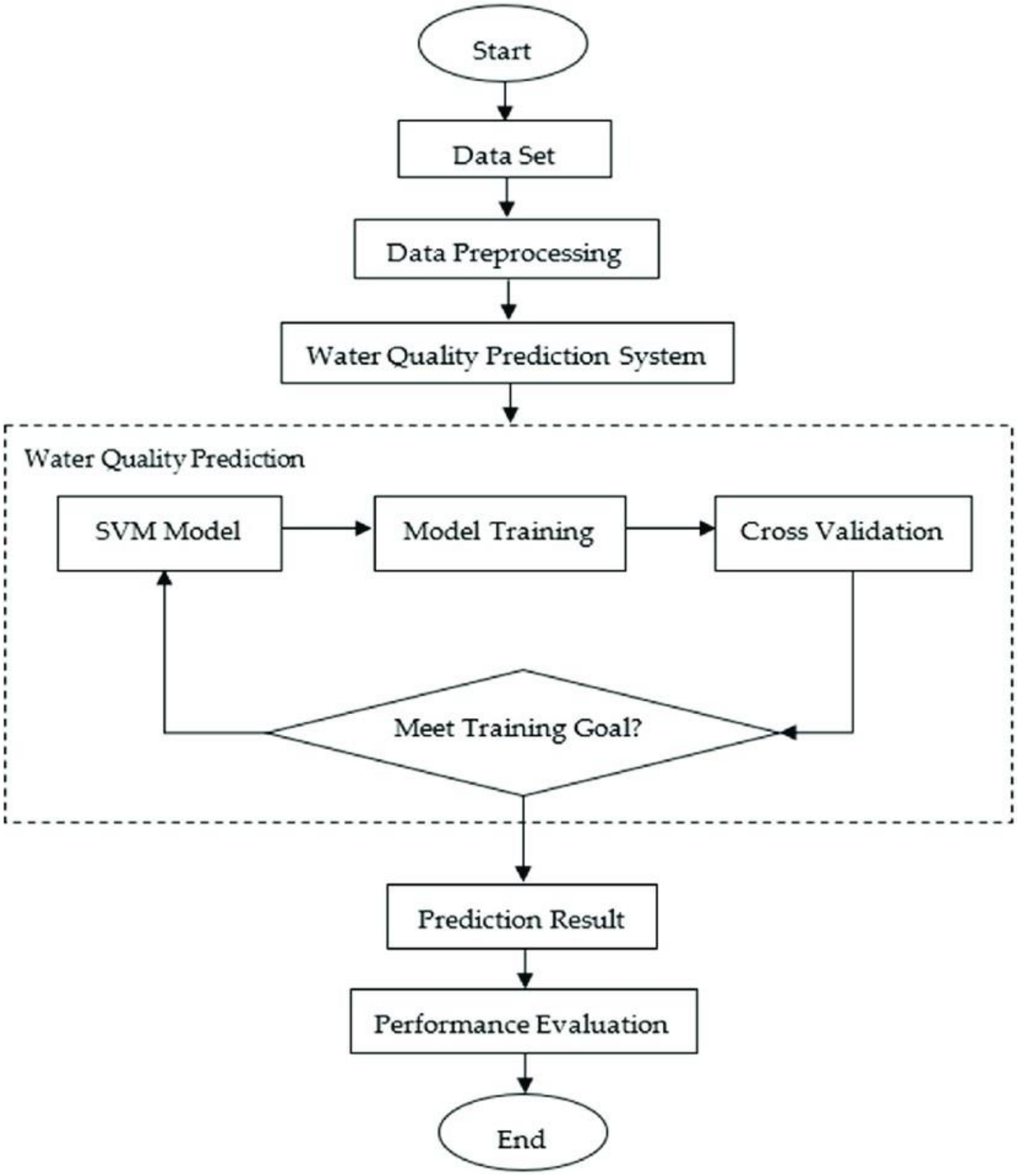


Fig-7:Software Block Diagram

1. Start: The beginning of the process.
2. Data Set: Collecting the dataset containing water quality parameters.
3. Data Preprocessing: Preparing the data by cleaning, normalizing, and transforming it as required for analysis.
4. Water Quality Prediction System: The main system that processes the data and makes predictions.
    - ■ SVM Model: The model type used for prediction, in this case, a Support Vector Machine.
    - ■ Model Training: The phase where the SVM model is trained using the preprocessed data.
    - ■ Cross Validation: A technique used to assess the performance of the model by dividing the data into training and validation sets multiple times.
    - ■ Meet Training Goal?: A decision point where the system checks if the training goals (e.g., accuracy, precision) are met. If not, it loops back to model training for further tuning.
5. Prediction Result: The output from the model, which predicts the quality of the water based on the trained SVM model.
6. Performance Evaluation: Assessing the performance of the prediction by evaluating metrics like accuracy, precision, recall, etc.
7. End: The conclusion of the process.

In summary, the flowchart outlines the process from collecting and preprocessing data, training and validating an SVM model, to generating and evaluating water quality predictions.

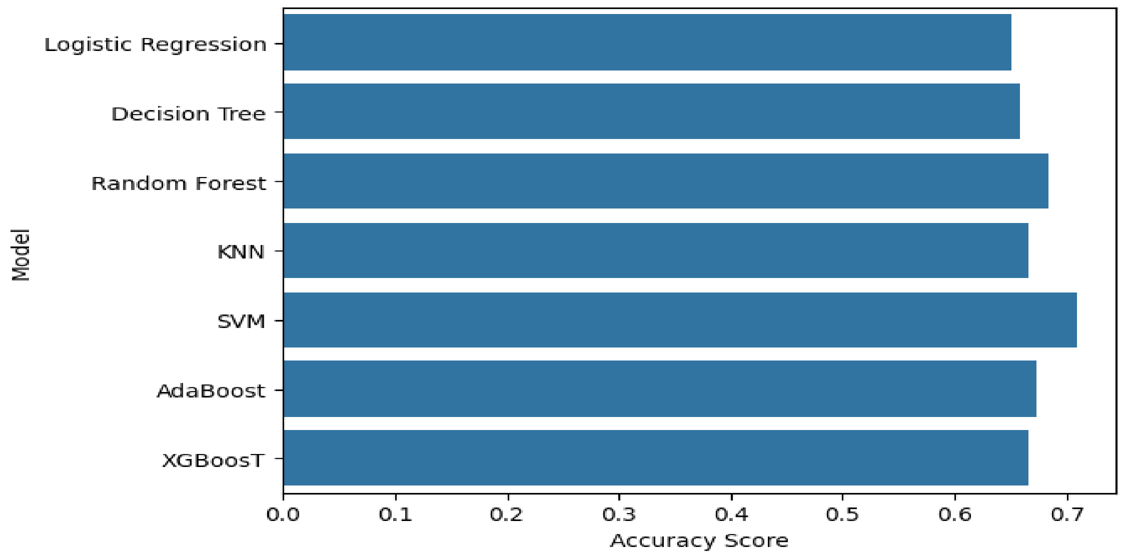


Fig-8:Model Accuracy

1. Support Vector Machine (SVM)

Accuracy: Highest (around 0.68 - 0.70)

- SVM achieves the highest accuracy, indicating its strength in finding the optimal hyperplane that separates the classes with the maximum margin.
- This suggests that the dataset has a clear margin of separation, and SVM's ability to handle high-dimensional spaces contributes to its superior performance.

2. XGBoost

Accuracy: Second highest (around 0.68 - 0.70)

- XGBoost performs nearly as well as SVM, highlighting its effectiveness in boosting and correcting errors iteratively.

3. Random Forest

Accuracy: High (around 0.66 - 0.68)

- Random Forest achieves a high accuracy, reflecting its capability to reduce overfitting by averaging multiple decision trees.
- Its performance suggests that the dataset benefits from ensemble methods that incorporate multiple weak learners.

4. AdaBoost

Accuracy: Moderate to High (around 0.64 - 0.66)

- AdaBoost shows good accuracy, indicating that boosting weak classifiers into a strong classifier works well for this dataset.
- However, it may be slightly less robust compared to Random Forest and XGBoost, potentially due to its sensitivity to noisy data.

5. K-Nearest Neighbors (KNN)

Accuracy: Moderate (around 0.64 - 0.66)

- KNN's performance is moderate, reflecting its simplicity and instance-based learning approach.
- The accuracy suggests that KNN can be effective, but its performance heavily depends on the choice of k and distance metrics.

6. Logistic Regression

Accuracy: Moderate (around 0.62 - 0.64)

- Logistic Regression shows moderate accuracy, which indicates that the relationship between features and target is somewhat linear.
- This performance is expected for simpler models, especially if the dataset has complex patterns that linear models cannot capture.

7. Decision Tree

Accuracy: Moderate (around 0.62 - 0.64)

- Decision Tree's accuracy is similar to Logistic Regression, suggesting that it captures the dataset's structure to a certain extent.
- However, it may be prone to overfitting, especially if not pruned or if the tree is too deep.

The accuracy values highlight that SVM and XGBoost are the top-performing models, suggesting that they handle the dataset's characteristics effectively. Random Forest and AdaBoost also perform well, demonstrating the strength of ensemble methods. KNN, Logistic Regression, and Decision Tree show moderate performance, indicating their limitations in capturing complex patterns in the data.

4.2 PROPOSED MODULE:

SUPPORT VECTOR MACHINE:

SVM is known for its generalization ability, meaning it can perform well on unseen data. This is attributed to its capacity to maximize the margin between classes, which helps in reducing overfitting and improving the model's robustness. Additionally, SVM can be regularized using techniques that penalize large coefficients, such as L2 (Ridge) regularization, further enhancing its ability to generalize.

In practical terms, if an SVM model achieves an accuracy of 70.88%, it indicates that the model correctly predicts the binary outcome for approximately 70.88% of the dataset. This accuracy metric serves as a measure of the model's performance, reflecting its effectiveness in separating classes based on the given features. SVM's strengths in handling complex datasets and maintaininggood performance make it a versatile choice for various classification tasks in machine learning.

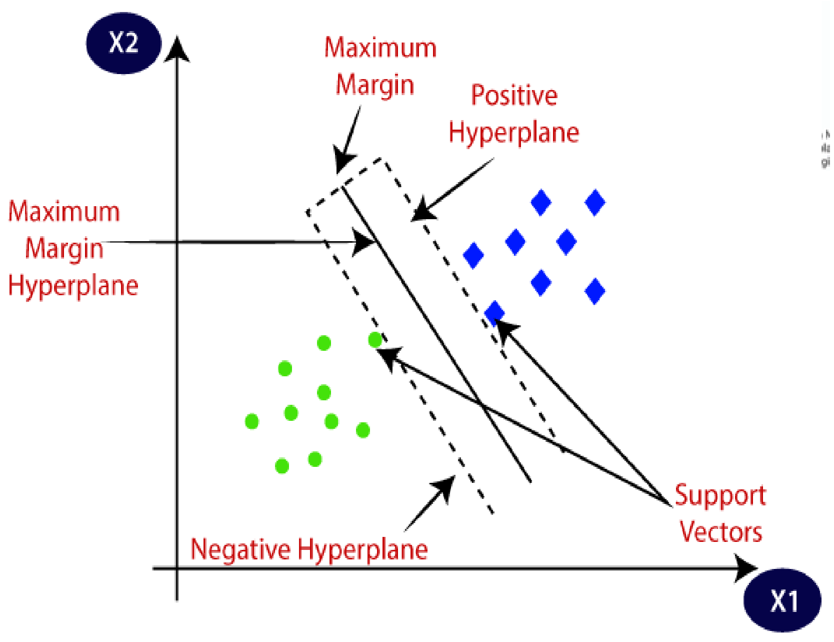


Fig-9:Support Vector Machine Representation

FORMULA:

Hyperplane:

A hyperplane is a decision boundary that separates the data intodifferent classes. It can be represented by the equation:

$$f\{x\}=w\cdot x+b=0$$

where:

- w is the weight vector,
- x is the input feature vector,
- b is the bias term.

Objective Function:

The goal of SVM is to maximize the margin between the two classes while ensuring correct classification of the data points. This is achieved by solving the following optimization problem:

$$\min \tfrac{1}{2}\|w\|^2$$

subject to:

$$yi(w\cdot xi+b)\geq 1 \forall i$$

where: yi is the class label of the i-th training sample.

Decision Function:

After solving for the Lagrange multipliers, we compute the weight vector w\mathbf{w}w and bias term b using:

$$w=\sum i=1N\alpha iyixi$$

$$b=yk-w\cdot xk$$

where k is an index such that $0<\alpha k<C$

The final decision function for classifying a new sample x\mathbf{x}x is:

$$f(x)=\mathrm{sign}(w\cdot x+b)$$

4.2.1 DATA PREPROCESSING:

- Handling Missing Values:
  - Methods: Imputation (mean, median, mode), deletion, or using algorithms that support missing values.
  - Example: If a feature has missing values, you can fill them with the mean value of that feature.
- Removing Outliers:
  - Purpose: Outliers can distort the decision boundary of the SVM, leading to poor generalization.
  - Techniques: Z-score method, IQR (Interquartile Range) method, or domain-specific knowledge.
- Correcting Errors:
  - Purpose: Ensure data accuracy and consistency.
  - Methods: Manually inspecting the data or using automated scripts to find and correct errors.

FEATURE SCALING:

- **Purpose**: SVM is sensitive to the scale of the features. Different scales can distort the distance calculations and thus the decision boundary.
- **Techniques**:

  - **Standardization (Z-score normalization)**: Transforms data to have a mean of 0 and a standard deviation of 1.

4.2.2 FEATURE SELECTION/EXTRACTION

- **Purpose**: Reduce the number of features to improve model performance and prevent overfitting.

- **Techniques**:
    - **Filter Methods**: Select features based on statistical measures (e.g., correlation, chi-square test).
    - **Wrapper Methods**: Use model performance to select features (e.g., recursive feature elimination).
    - **Embedded Methods**: Select features as part of the model training process (e.g., L1 regularization).

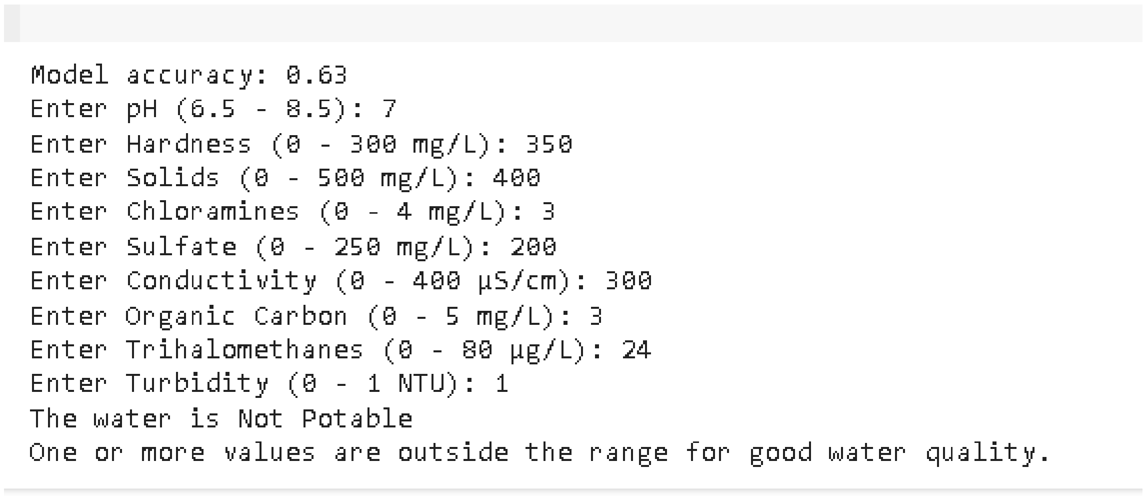
4.2.3 CLASSIFICATION:

SVM is a type of machine learning algorithm that works by finding the hyperplane that best separates the data into different classes.

Hyperplane: In an n-dimensional space, a hyperplane is a flat affine subspace of dimension n-1. For example, in 2D, a hyperplane is a line; in 3D, it is a plane.
Margin: The margin is the distance between the hyperplane and the nearest data point from either class. SVM aims to maximize this margin.

Support Vectors: These are the data points that lie closest to the hyperplane. They are critical in defining the position and orientation of the hyperplane.

4.2.4 RESULT:



```
Model accuracy: 0.63
Enter pH (6.5 - 8.5): 7
Enter Hardness (0 - 300 mg/L): 350
Enter Solids (0 - 500 mg/L): 400
Enter Chloramines (0 - 4 mg/L): 3
Enter Sulfate (0 - 250 mg/L): 200
Enter Conductivity (0 - 400 µS/cm): 300
Enter Organic Carbon (0 - 5 mg/L): 3
Enter Trihalomethanes (0 - 80 µg/L): 24
Enter Turbidity (0 - 1 NTU): 1
The water is Not Potable
One or more values are outside the range for good water quality.
```

Fig-10:Predicted Data

The model predicts that the water is "Not Potable." This conclusion is based on the entered values for various water quality parameters. The note "One or more values are outside the range for good water quality" indicates that some of the entered values fall outside the typical safe ranges for potable water, leading the model to determine that the water is not safe for drinking.

CHAPTER 5

CONCLUSION:

In this project, we have successfully developed a water quality prediction system that combines the precision of machine learning with practical hardware implementation. By leveraging Support Vector Machine (SVM) for software analysis, we have achieved higher accuracy in predicting water quality compared to other models. This robust approach ensures reliable and accurate predictions, vital for maintaining public health and safety.

On the hardware front, we utilized a turbidity sensor to measure the water quality in real-time. This integration of SVM with turbidity sensing provides a comprehensive solution that is both effective and efficient. The use of SVM enables sophisticated data analysis and prediction, while the turbidity sensor offers practical, on-site measurement capabilities.

Overall, our system provides an innovative and reliable method for monitoring and predicting water quality, contributing to better water management practices and safeguarding the health of communities.

FUTURE WORKS:

Looking ahead, the water quality prediction project sets the stage for future advancements in environmental monitoring and predictive analytics. Continued research and innovation in machine learning algorithms, sensor technologies, and data-driven decision support systems will further enhance our ability to predict and manage water quality effectively.

By embracing interdisciplinary approaches and fostering partnerships between academia, industry, and government agencies, we can address emerging challenges such as climate change impacts, urbanization, and industrial development on water resources.

In conclusion, the water quality prediction project underscores the importance of leveraging technology and collaborative efforts to safeguard water resources, promote environment.

REFERENCES:

1.Y. Khan and C. S. See, "Predicting and analyzing water quality using Machine Learning: A comprehensive model," *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, Farmingdale, NY, USA, 2016, pp. 1-6, doi: 10.1109/LISAT.2016.7494106.

2.D. Brindha, V. Puli, B. K. S. NVSS, V. S. Mittakandala and G. D. Nanneboina, "Water Quality Analysis and Prediction using Machine Learning," *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2023, pp. 175-180, doi: 10.1109/ICCMC56507.2023.10083776.

3.K. Abirami, P. C. Radhakrishna and M. A. Venkatesan, "Water Quality Analysis and Prediction using Machine Learning," *2023 IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT)*, Bhopal, India, 2023, pp. 241-245, doi: 10.1109/CSNT57126.2023.10134661.

4.A. Sarma and J. S. O, "An Analysis on the Techniques for Water Quality Prediction from Remotely Sensed data," *2023 International Conference on Recent Trends in Electronics and Communication (ICRTEC)*, Mysore, India, 2023, pp. 1-6, doi: 10.1109/ICRTEC56977.2023.10111933.

5.R. I. Singh and U. K. Lilhore, "A Survey of Machine Learning Models for Water Quality Prediction," *2023 8th International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 2023, pp. 1069-1074, doi: 10.1109/ICCES57224.2023.10192728.

6.N. M. Ragi, R. Holla and G. Manju, "Predicting Water Quality Parameters Using Machine Learning," *2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, Bangalore, India, 2019, pp. 1109-1112, doi: 10.1109/RTEICT46194.2019.9016825.

7.V. K. P, S. K, B. M. D and R. Reshma, "Predicting and Analyzing Water Quality using Machine Learning for Smart Aquaculture," *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, Erode, India, 2023, pp. 354-359, doi: 10.1109/ICSCDS56580.2023.10104677.

8.A. L. Lopez, N. A. Haripriya, K. Raveendran, S. Baby and C. V. Priya, "Water quality prediction system using LSTM NN and IoT," *2021 IEEE International Power and Renewable Energy Conference (IPRECON)*, Kollam, India, 2021, pp. 1-6, doi: 10.1109/IPRECON52453.2021.9640938.

9.S. S, R. Moon, M. Umer, K. S. Raju, N. Bhaskar and R. Okali, "A Prediction of Water Quality Analysis Using Machine Learning," *2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, Ballar, India, 2023, pp. 1-6, doi: 10.1109/ICDCECE57866.2023.10150940.