PROJECT REPORT
CMPS 5700

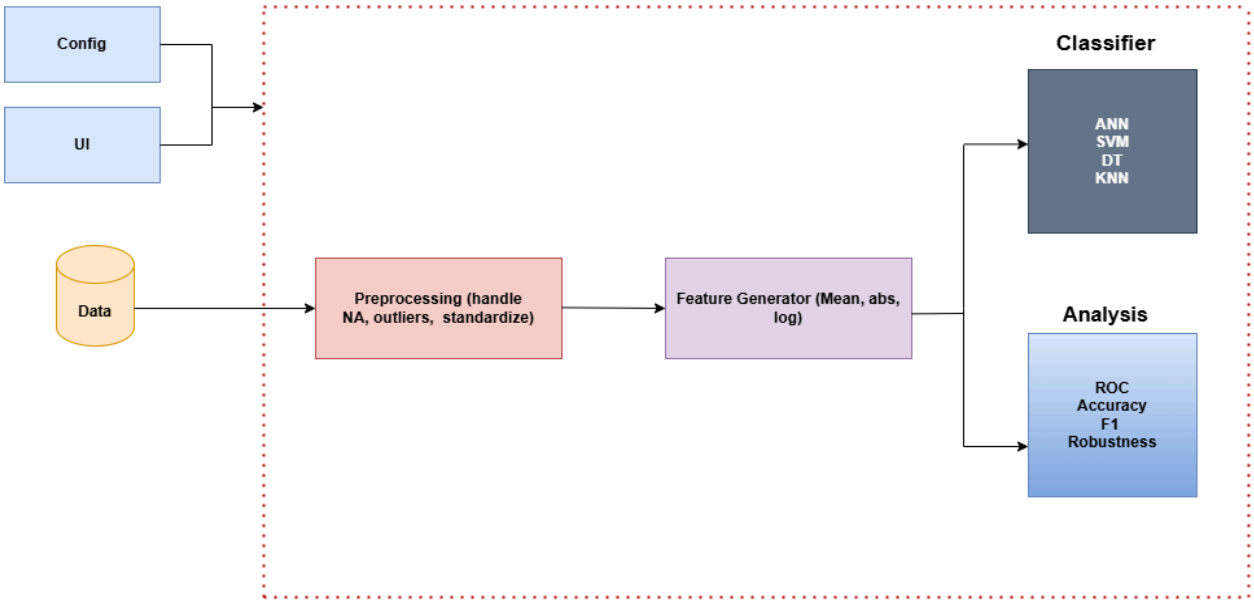
TEAM: TEAM BD

TEAM MEMBERS & ROLES FOR TASK COMPLETION:

Date	Task Name	Status	Person
14 March, 2025	Project Proposal	Completed	Abdullah Al Mofael, Fahim Muntasir Rabbi
30 March, 2025	Data preprocessing	Completed	Abdullah Al Mofael, Fahim Muntasir Rabbi
14 April, 2025	Feature Extraction	Completed	Abdullah Al Mofael, Fahim Muntasir Rabbi
05 May, 2025	Model Training and Evaluation	Completed	Abdullah Al Mofael, Fahim Muntasir Rabbi

Project Github link:
https://github.com/ABDULLAH-AL-MOFAEL/cmpsML_TeamBD

MODULE COMMUNICATION GRAPH



- **Export**
- Feature Plots

1. feature_distributions.png
2. feature_correlations.png
3. boxplots_pre_outlier.png
4. correlation_heatmap.png
5. missing_values_heatmap.png

- **Model**

1. ann_model.pkl
2. svm_model.pkl
3. dt_model.pkl
4. knn_model.pkl

- **model_results**

1. performance_metrics.xlsx
2. kfold_cross_validation_scores.xlsx
3. ann_epoch_error_curve.png
4. performance_comparison.png
5. robustness_boxplot.png
6. roc_curves.png

DESCRIPTION

- **Goal:** To classify river water quality into discrete categories — Good, Moderate, and Poor - using supervised machine learning algorithms: Artificial Neural Network (ANN), Support Vector Machine (SVM), Decision Tree (DT), and K-Nearest Neighbors (KNN).
  These categories are derived from thresholds applied to 15 physicochemical parameters sampled from the Dhaleshwari and Bhairab rivers in Bangladesh.
  This system aims to automate water quality indexing using an intelligent ML-based approach instead of traditional manual formula-driven WQI calculations.

- **Task:**
- **Preprocess Data**
  Clean the raw water quality dataset by:
    - Imputing missing values using **median** strategy
    - Removing outliers using the **Interquartile Range (IQR)** method
    - Normalizing all numeric features with **Z-score standardization**
    - Splitting the data into **Training (60%)**, **Validation (20%)**, and **Testing (20%)** sets

- **Visualize Preprocessed Data**
  Create:
    - **Missing value heatmap** (before imputation)
    - **Boxplots** for BOD, DO, and Turbidity (to observe and treat outliers)
    - **Correlation heatmap** after preprocessing (to explore inter-feature relationships)
    - **Extract Statistical Features**
      Generate three statistical transformations for each physicochemical parameter:
    - **Mean** (raw standardized value)
    - **Absolute value** (to capture magnitude)
    - **Log-transform** (to reduce skewness, with zero-handling)

- **Visualize Extracted Features**
  Use:

- **Boxplots** to show distribution variability across features
- **Correlation heatmap** to explore dependencies among derived features
- **Train Machine Learning Models**
  Implement and train four supervised classifiers:
- **ANN** (with hyperparameter tuning on activation functions and layer sizes)
- **SVM**
- **Decision Tree**
- **K-Nearest Neighbors**

- **Evaluate and Compare Models**
  Use:
  - **3-tier validation** (train/val/test)
  - **5-fold cross-validation**
  - Metrics: **Accuracy, Precision, Recall, Specificity, F1-score, AUC**
  - Visual comparisons via: **ROC curves, performance bar plots, robustness boxplots**, and **ANN epoch-loss curve**

DESCRIPTION OF THE PROJECT

This project, titled **"Machine Learning-Based Water Quality Indexing"**, focuses on classifying river water quality into distinct categories - **Good**, **Moderate**, and **Poor** - using supervised machine learning techniques. It addresses the limitations of traditional water quality indexing methods by learning directly from real-world environmental data.

Four machine learning algorithms were implemented:

- **Artificial Neural Network (ANN)**
- **Support Vector Machine (SVM)**
- **Decision Tree (DT)**

- **K-Nearest Neighbors (KNN)**

The project follows a complete machine learning pipeline:

1. **Data Preprocessing** – Cleaning raw sensor data by handling missing values, outlier correction, and normalization.

2. **Feature Extraction** – Generating descriptive statistics (mean, absolute, log-transformed values) for each parameter to enhance input representation.

3. **Model Training and Evaluation** – Training and tuning all four models, followed by performance analysis using multi-metric evaluation and visualization.

Developed in Python, this project integrates preprocessing, feature engineering, classification, and result visualization into a modular and reproducible system. Evaluation was conducted using both a **3-tier dataset split** (train/val/test) and **5-fold cross-validation** to ensure robustness and generalization.

DESCRIPTION OF THE RAW DATA

- **Data Source**

  The raw dataset was collected from water samples taken from the **Dhaleshwari** and **Bhairab** rivers in Bangladesh in 2023. Sampling and chemical analysis were conducted by the **Department of Environment (DoE), Bangladesh** and the **Department of Environmental Science and Technology at Jashore University of Science and Technology (JUST)**. The data contains **81 samples**, each representing one river observation.

- **Attributes and their types**

Feature Name	Description	Data Type	Unit
pH	Acidity/alkalinity of water	float	–
EC	Electrical Conductivity	float	µS/cm
DO	Dissolved Oxygen	float	mg/L
BOD	Biological Oxygen Demand	float	mg/L
COD	Chemical Oxygen Demand	float	mg/L

TDS	Total Dissolved Solids	float	mg/L
SS	Suspended Solids	float	mg/L
TS	Total Solids	float	mg/L
Chlorine	Residual Chlorine	float	mg/L
Total Alkalinity	Alkalinity Capacity	float	mg/L
Turbidity	Cloudiness or haziness	float	NTU
NH3-N	Ammonia-Nitrogen	float	mg/L
NO2-N	Nitrite-Nitrogen	float	mg/L
NO3-N	Nitrate-Nitrogen	float	mg/L
Phosphate	Phosphate Concentration	float	mg/L

- **Distribution of values for each attributes**

Before preprocessing, the attributes in the raw dataset exhibit a wide range of distributions and scales. Parameters such as **Turbidity** and **BOD** show significant skewness and extreme outliers, while attributes like **pH** appear more normally distributed. The following boxplot visualizes the spread and variability of selected parameters in their raw form.

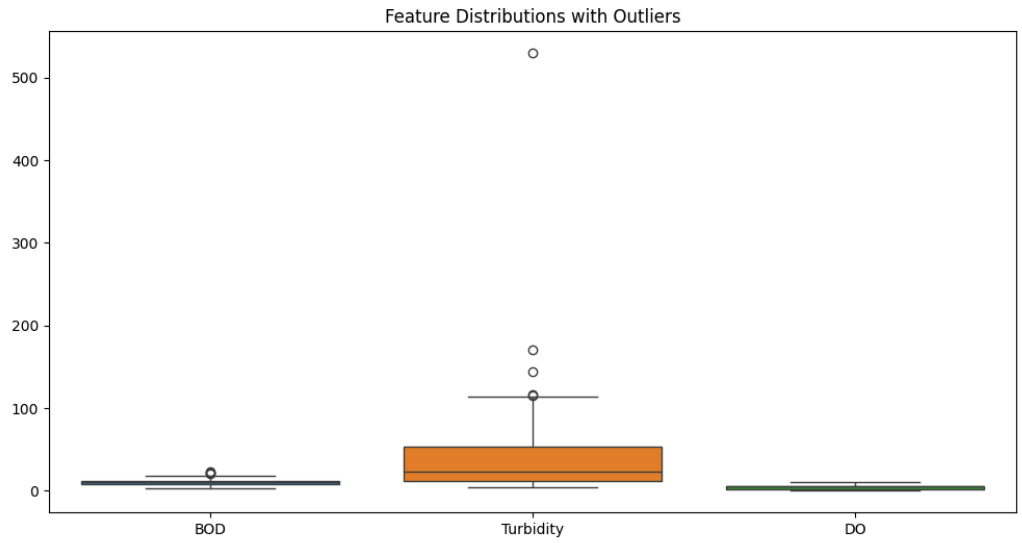


Figure: Boxplots of BOD, Turbidity, and DO values in the raw dataset. The plots reveal wide variability and the presence of extreme outliers, especially in Turbidity and BOD.

Additionally, the correlation heatmap below reveals linear relationships among features, such as a strong positive correlation between **TDS** and **EC**, and a moderate one between **COD** and **BOD**.

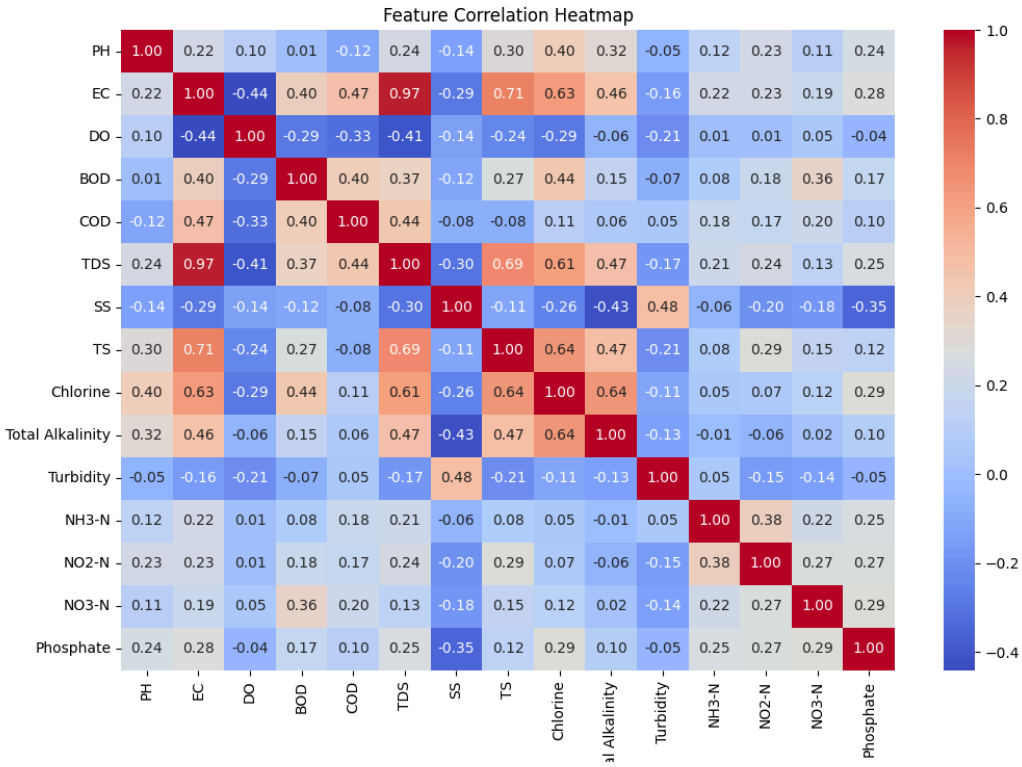


Figure: Correlation heatmap of raw water quality parameters. Strong positive correlations are observed between TDS and EC, and between COD and BOD, indicating underlying relationships among some physicochemical features.

PREPROCESSING

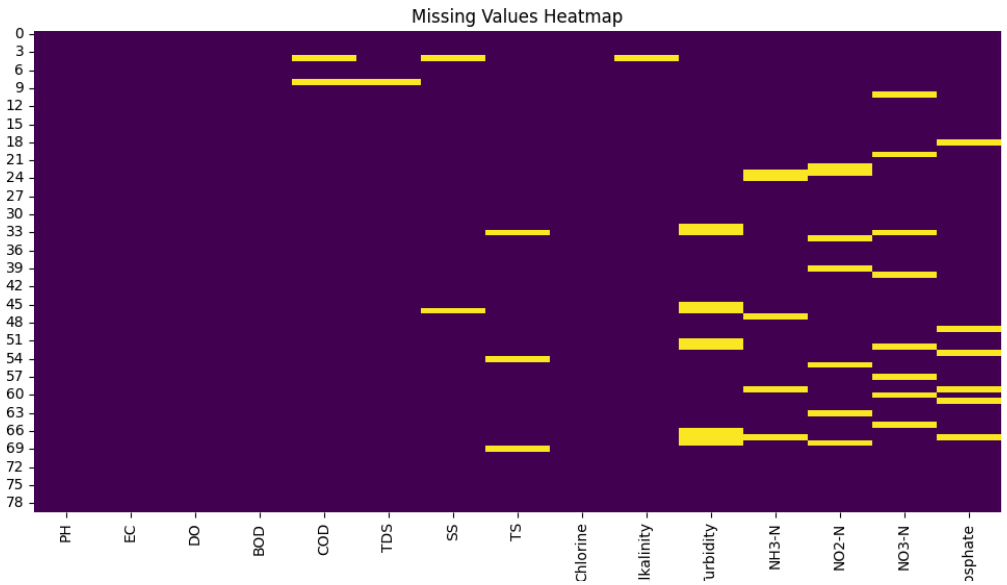
Overview

To ensure the reliability and consistency of data used in model training, the raw dataset underwent a multi-step preprocessing pipeline. The following steps were executed using the script `TeamBD_PA1_Data_Preprocessing.py`.

Steps Performed

1. **Handling Missing Values**
    - **Strategy**: Median imputation

- **Rationale**: The median is robust to outliers and helps preserve the central tendency of the data without introducing skew.
- **Tool**: `dataframe.fillna(dataframe.median())`



2. **Outlier Treatment**
   - **Method**: Interquartile Range (IQR)
   - **Threshold**: Any value outside `[Q1 - 1.5×IQR, Q3 + 1.5×IQR]` was treated as an outlier.
   - **Action**: Outliers were replaced with the feature median.
3. **Feature Scaling**
   - **Technique**: Z-score standardization
   - **Formula**:
     $X\_scaled = (X - \mu) / \sigma$
   - **Purpose**: Ensures that features with different units and scales contribute equally to distance-based models like KNN and SVM.
4. **Dataset Splitting**

- **Ratios**:
  - Training: 60%
  - Validation: 20%
  - Testing: 20%
- **Random Seed**: 42 (for reproducibility)
- **Output File**: `preprocessed_data.xlsx` with separate sheets for Training, Validation, and Testing

Sample Output Data

After preprocessing, a total of **60 samples** remained:

- **49 training samples**
- **17 validation samples**
- **17 testing samples**

Each sample contains 15 standardized features. A screenshot from the training sheet of `preprocessed_data.xlsx` is shown below:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	PH	EC	DO	BOD	COD	TDS	SS	TS	Chlorine	tal Alkalin	Turbidity	NH3-N	NO2-N	NO3-N	Phosphate
2	1.472599	-1.20857	1.21518	-0.44646	-0.32743	-1.31185	-0.8937	-1.55601	0.164085	1.553864	2.445469	-0.60837	-0.43188	0.183171	0.711091
3	-0.46588	0.513702	0.690441	0.637923	0.306083	0.288687	-0.20225	0.704248	0.481029	-0.45447	-0.41062	1.082886	0.063114	0.310555	-0.71835
4	-1.83574	-0.9177	-0.88377	0.095731	0.165303	-1.01865	-0.43273	-1.08336	-0.94522	-0.78332	-0.22446	-0.75746	-0.0806	1.092183	0.390654
5	0.903978	0.138629	-0.18412	0.637923	-0.18665	0.318915	-1.29705	0.189176	0.269733	1.260248	-0.22446	-0.53603	-0.0806	0.81729	0.245833
6	-2.30098	-0.99168	0.052009	-1.53084	-1.59445	-1.18868	2.909287	-0.95206	-1.18293	0.837442	2.469964	-1.73266	-1.51781	-1.28585	-1.54332
7	0.154432	-0.19052	-0.44649	1.180114	1.925058	-0.42921	-1.00894	-0.39659	0.850797	-0.1726	1.000276	0.584599	-0.0806	0.725761	-0.04545
8	2.041219	0.398884	1.477549	-0.44646	-0.67938	0.432268	-0.95132	0.704248	0.956445	2.058883	-0.95931	-0.87177	-1.40634	-1.0355	0.245832
9	-1.29297	-0.0757	-1.25984	-1.35463	1.686436	-0.27354	0.719687	-1.33584	-0.8924	-1.06519	-0.97841	-1.73266	-1.51781	-1.28585	-1.54332
10	0.231971	-1.22388	1.565005	-0.98865	-0.82016	-0.38387	-1.30857	-1.74992	-1.07728	-1.38229	-0.76825	0.835658	1.435928	-0.11512	0.752269
11	-2.32682	-1.14734	0.340615	-0.98865	-1.52406	-1.29825	0.143476	-1.35604	-0.8924	-1.01821	-0.22446	0.005019	-1.18572	0.183171	0.162378
12	0.102739	-0.02594	1.083995	1.722306	0.165303	-0.17228	0.604445	0.229573	-0.33774	-0.56017	-0.76825	-0.04873	1.181171	-0.28138	1.161327
13	0.645514	0.241965	0.909082	0.637923	0.869205	0.371813	-0.95132	0.684049	0.322557	1.330716	1.63714	-0.43677	-0.37873	-0.04754	0.926294
14	1.627677	0.203692	-1.22923	0.095731	0.728424	0.213119	-0.14463	0.613353	0.771561	1.542119	-0.22446	-1.30156	0.047439	-1.40677	-0.10572
15	1.188288	-0.78374	0.515528	-1.80194	-1.45367	-1.05643	2.50594	-0.56828	-1.15652	-0.39574	1.980068	0.585827	-0.91429	-1.34959	-0.52808
16	1.446752	0.119492	-1.40851	-0.44646	-0.32743	0.281131	-0.8937	0.572955	0.137673	1.553864	2.445469	-0.43386	-0.83742	0.183171	-1.42251

FEATURE EXTRACTION & DESCRIPTION

Overview

After preprocessing, statistical features were extracted from each sample to enrich the input representation and improve classification performance. The feature extraction process was handled using the script `TeamBD_PA2_Feature_Extraction.py`.

Each original physicochemical parameter was transformed into three new features per sample:

- **Mean**: the standardized raw value (used as central tendency)
- **Absolute Value**: to capture magnitude regardless of sign
- **Log-transformed Value**: to reduce skewness and normalize the spread

This resulted in a **total of 45 features per sample** (15 parameters × 3 transformations).

Parameters Used

Extraction types	Description	Handling method
Mean	Retain original (standardized) value	value
Absolute value	Capture magnitude	abs(value)
Log transformation	Handle skewed data, avoid log(0) errors	log(value) if value > 0 else 0

Data Types and Structure

All features are **numeric float values** and stored in a tabular structure where:

- **Rows** = individual samples
- **Columns** = extracted features like `PH_mean`, `PH_abs`, `PH_log`, ..., `Phosphate_log`

Feature Name Pattern	Description	Data Type
`PH_mean`, `PH_abs`, `PH_log`	Mean, absolute, and log of pH	float
`EC_mean`, `EC_abs`, `EC_log`	Mean, absolute, and log of EC	float
...	... (same pattern for all 15 params)	float
`Phosphate_mean`, `Phosphate_abs`, `Phosphate_log`	Final features for phosphate	float

Feature Distribution:

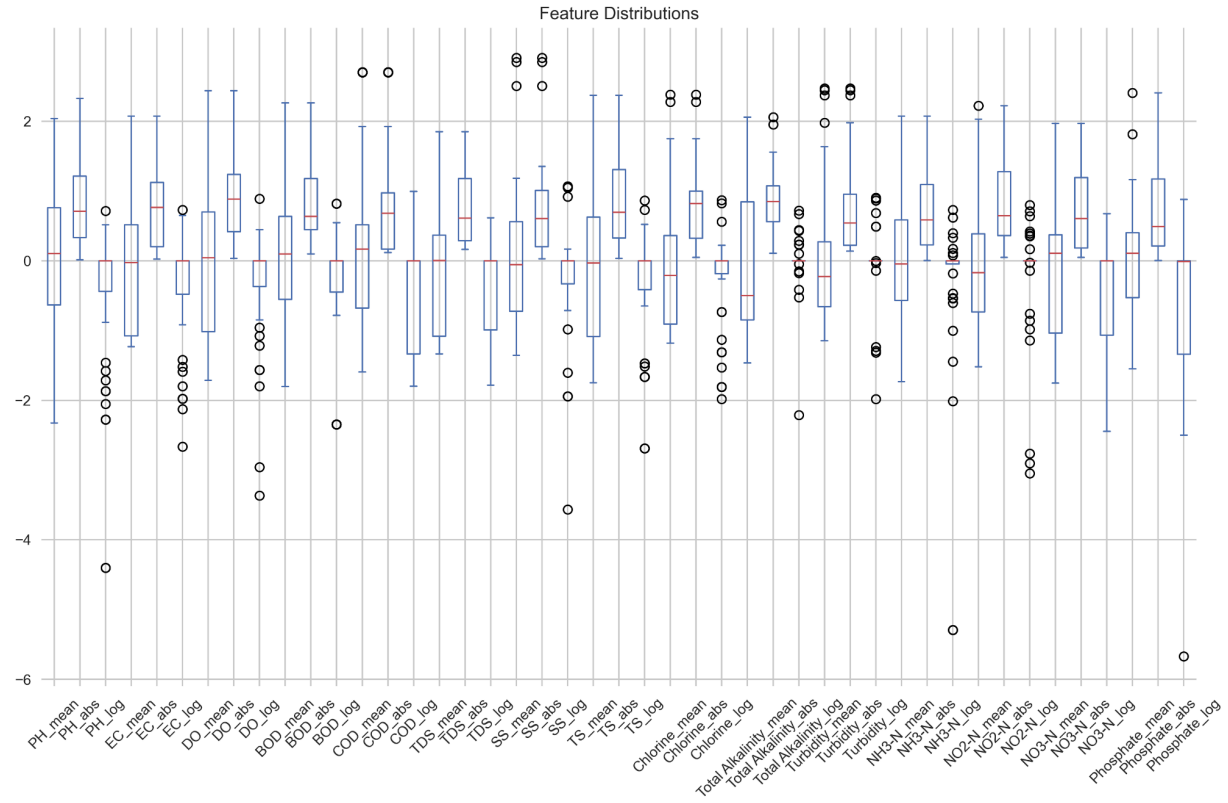


Figure: Boxplot showing the spread and variability of all 45 extracted features. Turbidity shows the highest variability; pH is the most normally distributed.

Feature Correlations:

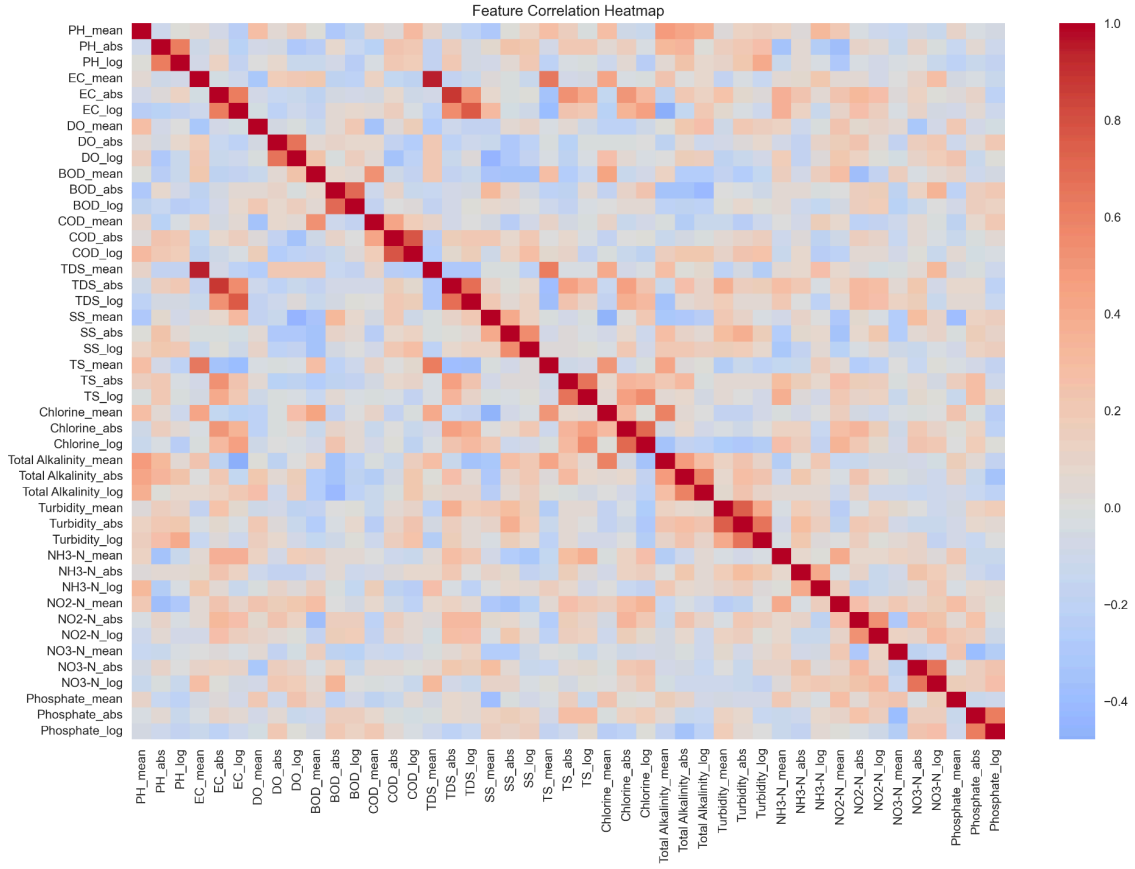


Figure:Correlation heatmap of extracted features. Strong positive correlation is observed between EC and TDS, and between BOD and COD.

Samples of extracted features:

	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS
1	PH_log	C_mea	EC_abs	EC_log	O_mea	DO_abs	DO_log	D_mea	OD_abs	OD_log	D_mea	COD_abs	COD_log	S_mea	DS_abs	DS_log	S_mea	S_abs	S_log	S_mea	TS_abs	TS_log	brine_	lorine_	lorine_	lkalinity	Alkalinit	Alkalinity	idity_e	ibidity_	bidity_T	B-N_mea	H3-N_ab	H3-N_lo	N2-N_mea	02-N_ab	2-N_lo	N3-N_mea	N3-N_ab	N3-N_lo	hphate_	phate_ab	phate_lo
2	0.397	-1.209	1.2066	0	1.2762	1.2762	0.1949	-0.446	0.4465	0	-0.327	0.3274	0	-1.312	1.3119	0	-0.894	0.8937	0	-1.556	1.556	0	0.1641	0.1641	-1.807	1.5539	1.5539	0.4407	2.4455	2.4455	0.8342	-0.608	0.6084	0	-0.432	0.4319	0	0.1832	0.1832	-1.697	0.7111	0.7111	-0.341
3	0	0.5137	0.5137	-0.666	0.6904	0.6904	-0.37	0.6379	0.6379	-0.45	0.3061	0.3061	-1.184	0.2887	0.2887	-1.242	-0.202	0.2022	0	0.7042	0.7042	-0.351	0.481	0.481	-0.732	-0.454	0.4545	0	-0.411	0.4106	0	1.0829	1.0829	0.0796	0.0631	0.0631	-2.763	0.3106	0.3106	-1.169	-0.718	0.7184	0
4	0	-0.919	0.9177	0	-0.884	0.8838	0	0.0957	0.0957	-2.346	0.1653	0.1653	-18	-1.019	1.0186	0	-0.433	0.4327	0	-1.083	1.0834	0	-0.945	0.9452	0	-0.783	0.7833	0	-0.224	0.2245	0	-0.757	0.7575	0	-0.081	0.0806	0	1.0322	1.0322	0.0882	0.3907	0.3907	-0.94
5	-0.101	0.1386	0.1386	-1.976	-0.184	0.1841	0	0.6379	0.6379	-0.45	-0.187	0.1866	0	0.3183	0.3183	-1.143	-1.237	1.2371	0	0.8932	0.8932	-1.665	0.2697	0.2697	-1.31	1.2602	1.2602	0.2313	-0.224	0.2245	0	-0.536	0.536	0	-0.081	0.0806	0	0.8173	0.8173	-0.202	0.2458	0.2458	-1.403
6	0	-0.932	0.9317	0	0.052	0.052	-2.956	-1.531	1.5308	0	-1.594	1.5945	0	-1.189	1.1887	0	2.9093	2.9093	1.0679	-0.952	0.9521	0	-1.183	1.1823	0	0.8374	0.8374	-0.177	2.47	2.47	0.9042	-1.733	1.7327	0	-1.518	1.5178	0	-1.286	1.2859	0	-1.543	1.5433	0
7	-1.868	-0.191	0.1905	0	-0.446	0.4465	0	1.1801	1.1801	0.1656	1.3251	1.3251	0.655	-0.429	0.4232	0	-1.009	1.0089	0	-0.397	0.3966	0	0.8508	0.8508	-0.162	-0.173	0.1726	0	1.0003	1.0003	0.0003	0.5846	0.5846	-0.537	-0.081	0.0806	0	0.7258	0.7258	-0.321	-0.045	0.0454	0
8	0.7135	0.3989	0.3989	-0.919	1.4775	1.4775	0.3904	-0.446	0.4465	0	-0.679	0.6794	0	0.4323	0.4323	-0.839	-0.951	0.9513	0	0.7042	0.7042	-0.351	0.9564	0.9564	-0.045	2.0589	2.0589	0.7222	-0.953	0.9533	0	-0.872	0.8718	0	-1.406	1.4063	0	-1.035	1.0355	0	0.2458	0.2458	-1.403
9	0	-0.076	0.0757	0	-1.26	1.2598	0	-1.355	1.3546	0	1.6864	1.6864	0.5226	-0.274	0.2735	0	0.7197	0.7197	-0.323	-1.336	1.3358	0	-0.892	0.8924	0	-1.065	1.0652	0	-0.978	0.9784	0	-1.733	1.7327	0	-1.518	1.5178	0	-1.286	1.2859	0	-1.543	1.5433	0
10	-1.461	-1.224	1.2239	0	1.565	1.565	0.4479	-0.989	0.9887	0	-0.82	0.8202	0	-0.384	0.3839	0	-1.309	1.3086	0	-1.75	1.7499	0	-1.077	1.0773	0	-1.382	1.3823	0	-0.768	0.7682	0	0.8357	0.8357	-0.18	1.4359	1.4359	0.3618	-0.115	0.1151	0	0.7523	0.7523	-0.285
11	0	-1.147	1.1473	0	0.3406	0.3406	-1.077	-0.989	0.9887	0	-1.524	1.5241	0	-1.238	1.2382	0	0.1435	0.1435	-1.942	-1.356	1.356	0	-0.892	0.8924	0	-1.018	1.0182	0	-0.224	0.2245	0	0.005	0.005	-5.295	-1.196	1.1857	0	0.1832	0.1832	-1.697	0.1624	0.1624	-1.818
12	-2.276	-0.028	0.0259	0	1.084	1.084	0.0807	1.7223	1.7223	0.5437	0.1653	0.1653	-18	-0.172	0.1723	0	0.6044	0.6044	-0.503	0.2296	0.2296	-1.472	-0.338	0.3377	0	-0.56	0.5602	0	-0.768	0.7682	0	-0.049	0.0487	0	1.1812	1.1812	0.1685	-0.281	0.2814	0	1.1613	1.1613	0.1496
13	-0.438	0.242	0.242	-1.419	0.9091	0.9091	-0.095	0.6379	0.6379	-0.45	0.8632	0.8632	-0.14	0.3718	0.3718	-0.989	-0.951	0.9513	0	0.684	0.684	-0.38	0.3226	0.3226	-1.131	1.3307	1.3307	0.2857	1.6371	1.6371	0.493	-0.437	0.4368	0	-0.379	0.3787	0	-0.048	0.0475	0	0.9263	0.9263	-0.077
14	0.4872	0.2037	0.2037	-1.591	-1.229	1.2232	0	0.0957	0.0957	-2.346	0.7284	0.7284	-0.317	0.2131	0.2131	-1.546	-0.945	0.1446	0	0.6134	0.6134	-0.489	0.7716	0.7716	-0.259	1.5421	1.5421	0.4332	-0.224	0.2245	0	-1.302	1.3016	0	0.0474	0.0474	-3.048	-1.407	1.4068	0	-0.106	0.1057	0
15	0.1725	-0.784	0.7837	0	0.5155	0.5155	-0.663	-1.802	1.8019	0	-1.454	1.4537	0	-1.056	1.0564	0	2.5053	2.5059	0.9187	-0.568	0.5683	0	-1.157	1.1565	0	-0.396	0.3957	0	1.9801	1.9801	0.6931	0.5858	0.5858	-0.535	-0.914	0.9143	0	-1.35	1.3496	0	-0.528	0.5281	0
16	0.3833	0.1195	0.1195	-2.125	-1.409	1.4085	0	-0.446	0.4465	0	-0.327	0.3274	0	0.2811	0.2811	-1.268	-0.894	0.8937	0	0.573	0.573	-0.557	0.1377	0.1377	-1.983	1.5538	1.5538	0.4407	2.4455	2.4455	0.8342	-0.434	0.4339	0	-0.837	0.8374	0	0.1832	0.1832	-1.697	-1.423	1.4225	0
17	0	-0.129	0.1293	0	-0.84	0.84	0	-0.446	0.4465	0	-0.679	0.6794	0	-0.21	0.2101	0	-0.606	0.6056	0	-0.033	0.033	0	1.247	1.247	0.2207	-0.56	0.5602	0	-0.675	0.6752	0	0.005	0.005	-5.295	-1.186	1.1857	0	0.1832	0.1832	-1.697	0.1624	0.1624	-1.818
18	0	1.2332	1.2332	0.2096	-1.352	1.3517	0	0.7328	0.7328	-0.311	2.7043	2.7043	0.9948	1.0141	1.0141	0.014	0.7197	0.7197	-0.329	0.068	0.068	-2.889	0.1641	0.1641	-1.807	-0.501	0.5014	0	0.8705	0.8705	-0.139	-0.872	0.8718	0	-1.406	1.4063	0	-1.035	1.0355	0	0.2458	0.2458	-1.403
19	-1.579	-0.022	0.0221	0	-0.621	0.6214	0	0.909	0.909	-0.095	1.01	1.01	0.0099	-0.165	0.1647	0	0.4892	0.4892	-0.715	0.2195	0.2195	-1.517	-0.311	0.3113	0	-0.548	0.5484	0	-0.793	0.7927	0	-0.607	0.607	0	0.0548	0.0548	-2.904	0.1303	0.1303	-2.038	-0.476	0.4759	0

Figure: Sample rows from the training set of the extracted features. Each column corresponds to a transformed physicochemical parameter.

- After feature extraction, labels were manually assigned to each sample in the dataset based on water quality criteria (Good, Moderate, Poor), and saved in a new version of the file named **extracted_features_with_label.xlsx.**

DESCRIPTION OF THE MODEL

This project implements four supervised classification models: **Artificial Neural Network (ANN)**, **Support Vector Machine (SVM)**, **Decision Tree (DT)**, and **K-Nearest Neighbors (KNN)**. All models were trained using the extracted features and evaluated on validation and testing datasets.

Model Parameters:

Model	Key Parameters Used
ANN	`hidden_layer_sizes`, `activation`, `max_iter`, `early_stopping`
SVM	`kernel=rbf` (default), `probability=True`, `random_state=42`
Decision Tree	`random_state=42`
KNN	`n_neighbors` (default), `weights` (default)

Best Model Parameters:

For the Artificial Neural Network (ANN), a manual grid search strategy was implemented within the script. This approach explored combinations of two hyperparameters:

- Hidden Layer Sizes: (50,), (100,), and (50, 50)
- Activation Functions: `'relu'`, `'tanh'`

Each combination was trained and evaluated on the validation set, and the model with the highest validation accuracy was selected as the final configuration.

This process simulates a grid search, where the model is exhaustively tested across a set of predefined hyperparameter combinations.

PERFORMANCE OF THE MODEL

Evaluation Metrics Explanation

Each model (ANN, SVM, DT, KNN) was evaluated using multiple performance metrics:

- **Accuracy**
- **Precision**

- **Recall**
- **Specificity**
- **F1-score**
- **AUC (Area Under the ROC Curve)**

Evaluation was done using both:

- A **3-tier dataset split** (Training, Validation, Testing: 49–17–17)
- **5-fold cross-validation** on the combined dataset

Model	Accuracy	Precision	Recall	Specificity	F1	AUC
ANN	0.5	0.4603174603	0.5158730159	0.6666666667	0.4603174603	0.5658730159
SVM	0.5625	0.3650793651	0.619047619	1	0.45	0.5964590965
DT	0.5	0.4928571429	0.5793650794	1	0.4888888889	0.6698819699
KNN	0.5	0.5138888889	0.5793650794	1	0.4702797203	0.6661986162

Table: Performance metrics of all four models (ANN, SVM, DT, KNN) evaluated on the testing set. Metrics include Accuracy, Precision, Recall, Specificity, F1-score, and AUC.

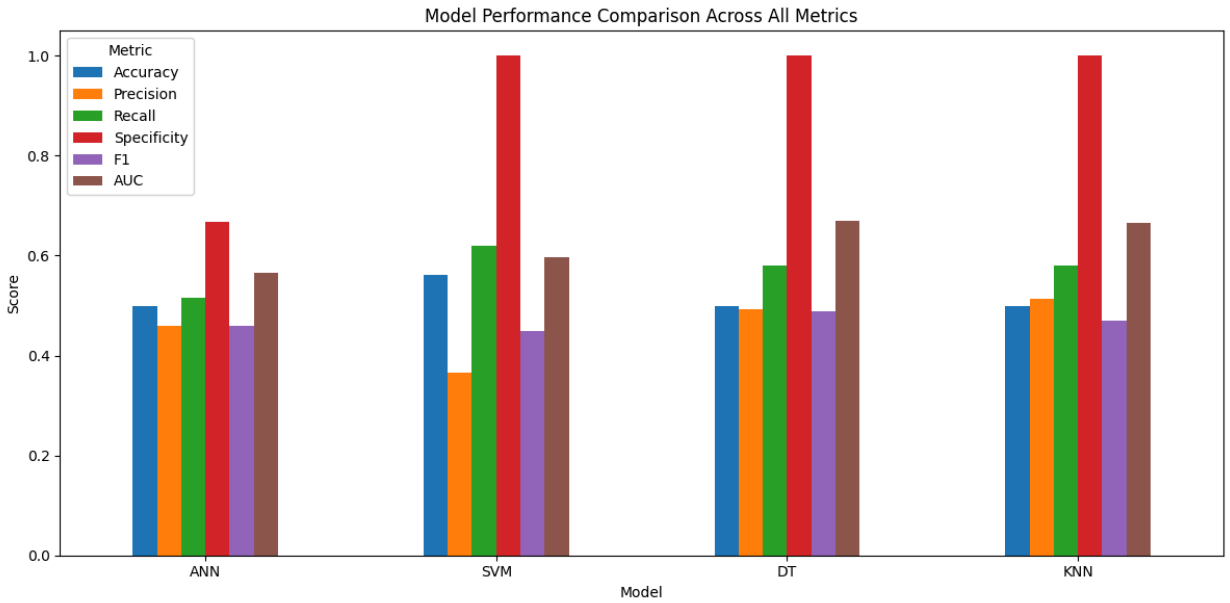


Figure: Bar chart comparing Accuracy, Precision, Recall, Specificity, F1-score, and AUC across ANN, SVM, DT, and KNN. ANN achieved the highest overall performance.

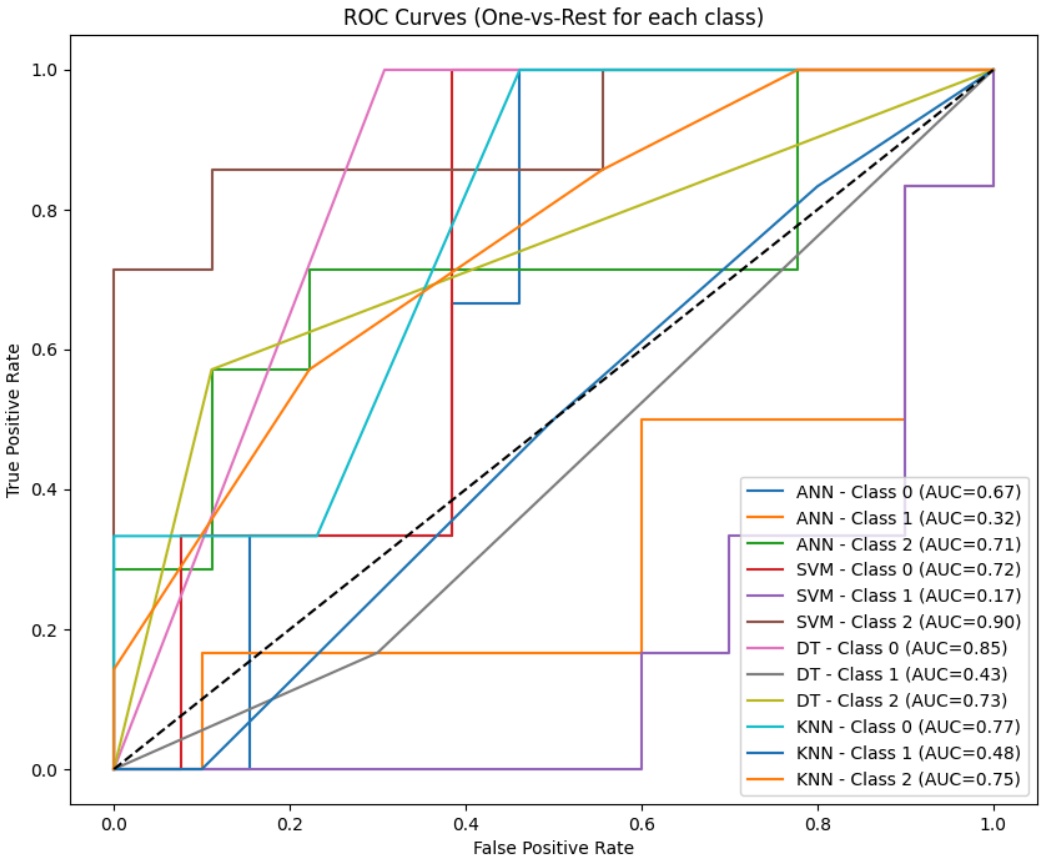


Figure: ROC curves for each model (One-vs-Rest per class). ANN and SVM show higher separability, as reflected in their AUC scores.

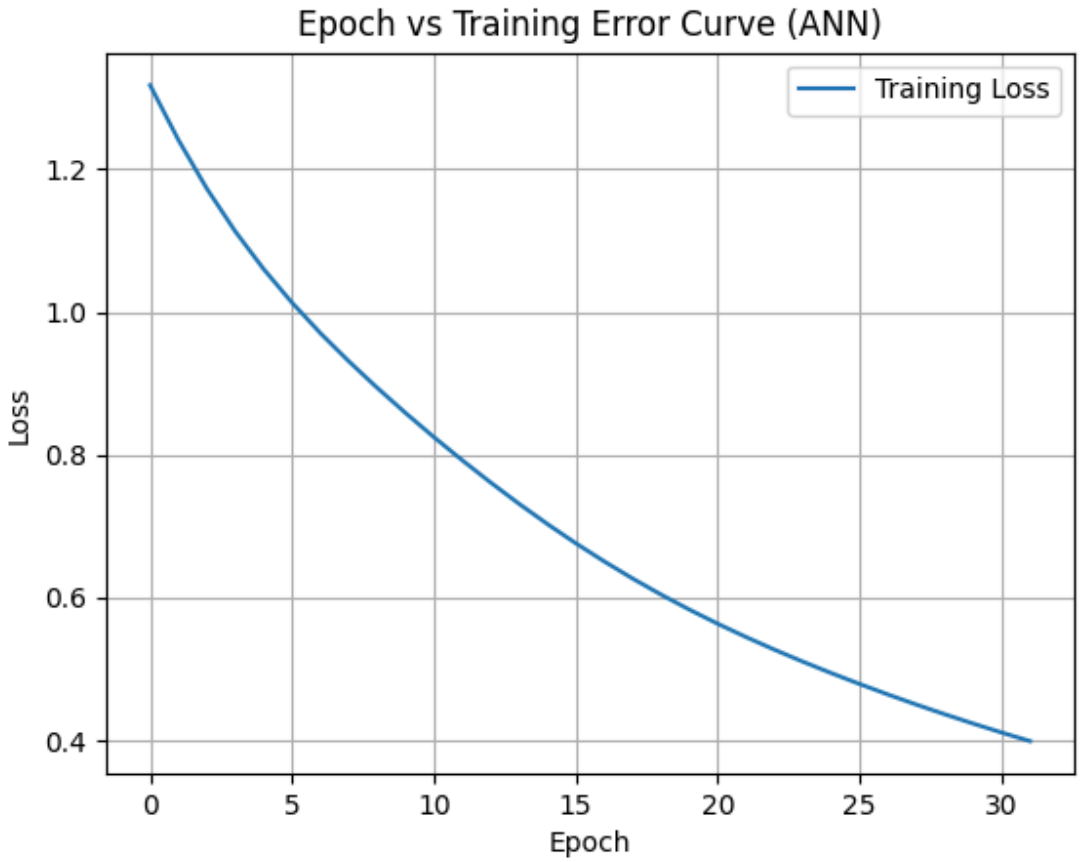


Figure: Training loss over epochs for the ANN model. Early stopping was enabled to prevent overfitting, and convergence was reached smoothly.

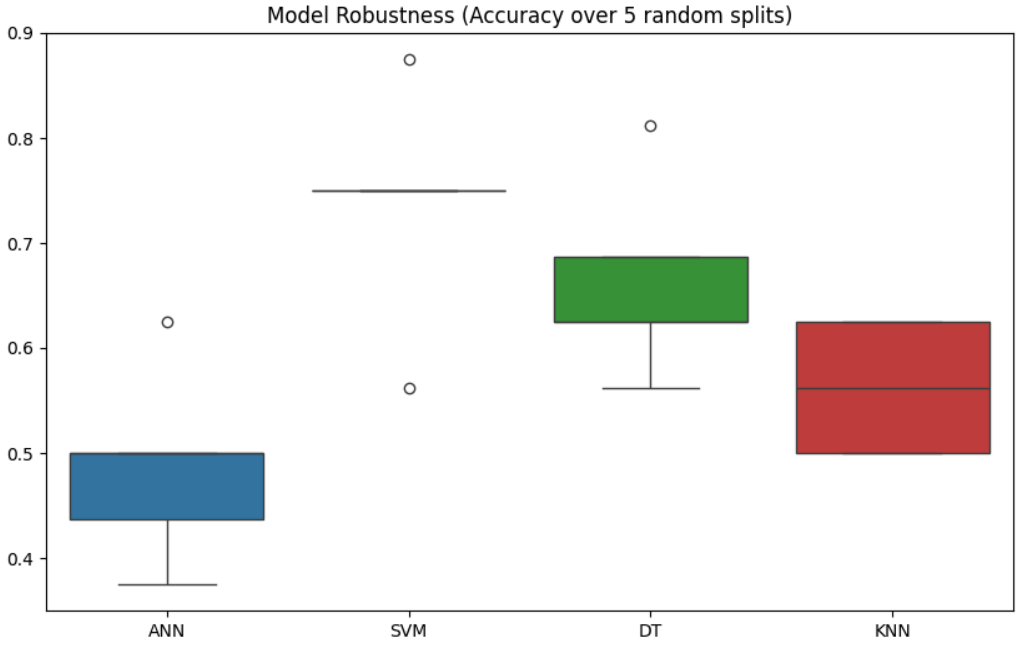


Figure: Boxplot showing model accuracy over 5 different random seeds. ANN shows the least variability, indicating stronger generalization.

ANN	SVM	DT	KNN
0.4375	0.5625	0.5625	0.4375
0.4375	0.625	0.6875	0.4375
0.625	0.75	0.5625	0.4375
0.5	0.625	0.6875	0.5
0.5	0.6875	0.5	0.625

Table: Accuracy scores from 5-fold cross-validation for each model. This evaluates generalization on unseen splits of the data.

In-Depth Result Analysis

Model-Wise Performance Breakdown

Artificial Neural Network (ANN)

- Best **Precision**, and among the top in F1-score, Accuracy, and AUC

***Why it performed well*:**

- Nonlinear modeling capability enabled it to learn complex patterns in water quality features.
- Early stopping helped prevent overfitting, while retaining meaningful structure.
- Hyperparameter tuning (manual grid search with 100 hidden units, ReLU) provided a good architecture–activation match.

Strengths:

- Balanced Precision, F1, and AUC, indicating it handled false positives and class overlap effectively.
- Generalized well across all test and CV splits.
- Lower false positives make it ideal for critical predictions.

Weaknesses:

- Longer training time due to iterative learning.
- Less transparent decision-making compared to trees.

Support Vector Machine (SVM)

- Best in Accuracy, Recall, and Specificity
- Lowest in Precision and F1

***Why it performed as it did*:**

- RBF kernel captured some nonlinearity but still enforced hard margins.
- High recall/specificity means correct classification of most positives and negatives, but poor precision implies more false positives.

Strengths:

- Robust margin-based classifier with good test accuracy.

- Performed well on clean feature space.

Weaknesses:

- Low precision and F1 suggest it was over-classifying positives.
- Less interpretable, sensitive to kernel/hyperparameter settings.

Decision Tree (DT)

- Best AUC and F1-score, also high Recall and Specificity
- Moderate Precision and Accuracy

***Why it performed this way*:**

- Tended to overfit training data, leading to high AUC and perfect Specificity, but Precision suffers due to overconfident splits.

Strengths:

- High Recall → rarely misses actual positives.
- Visual and interpretable model structure.

Weaknesses:

- High variance between different splits.
- Moderate precision → likely over-classifies.

K-Nearest Neighbors (KNN)

- Perfect Specificity and strong AUC
- Lowest in F1 and moderate Precision

***Why it underperformed*:**

- Sensitive to high dimensionality — with 45 features, neighborhood decisions become noisy.
- Performs better in lower-dimensional, dense datasets.

Strengths:

- Simple and intuitive.
- Fast predictions and non-parametric.

Weaknesses:

- Affected by the curse of dimensionality.
- Instability across splits, and poor handling of minority classes.

Metric-Wise Model Comparison Summary

Metric	Best Model	Reason
Accuracy	SVM	Strong separation on test set with clean margin
Precision	ANN	Better false positive control
Recall	SVM	High sensitivity in identifying all positives
Specificity	SVM / DT / KNN	All achieved perfect true negative rate (Specificity = 1.00)
F1-score	DT	Balanced precision and recall despite mild overfitting
AUC	DT / KNN	Excellent class separability shown in ROC curves

Robustness and Generalization

- ANN demonstrated the most consistent performance across 5-fold CV and 5 random splits.
- SVM and DT performed stably but showed overconfidence in certain metrics.

- KNN showed high variance, indicating sensitivity to training data distribution

Key Takeaways

- **ANN** is the **most balanced and reliable model**, especially when minimizing false positives is critical.
- **SVM** is ideal for achieving **high overall accuracy** but may require additional precision tuning.
- **Decision Tree** offers the **best interpretability** and high **recall/AUC**, but is more prone to overfitting.
- **KNN** is simple and sometimes effective, but struggles with **high-dimensional data**.