

Deep Learning-Based Predictive Modelling for Water Quality Assessment

Abstract — Ensuring safe and clean drinking water is considered as the main motto of the United Nations sustainable development goals 6 (SDG 6). However, due to the pollution and limited testing infrastructure there are many regions across the world where they are still struggling to get clean water. Our study proposes a deep learning based predictive model that aims at checking the quality of water using its properties like physicochemical characteristics, including pH, turbidity, sulphate concentration, and conductivity. This study focuses to develop an automated, cost-effective, and scalable solution that differentiates water samples as potable or non-potable. The model is trained on a real-world water quality dataset using deep neural network architecture. By using effective data preprocessing and feature selection techniques, study focuses on providing better model performance and efficiency. This proposed system contributes to SDG 6 by giving a practical support tool for water quality monitoring, particularly in areas where resources are not available. It provides timely checking and supports sustainable water management practices, ultimately aiding in the global vision of making safe water available to all.

Keywords — SDG 6, deep learning, water potability, classification, neural networks, water quality prediction, feature selection, sustainability.

I. ALIGNMENT WITH UN SUSTAINABLE DEVELOPMENT GOALS (SDGs)

This project directly aligns with Sustainable Development Goal 6 (SDG 6): Ensure availability and Sustainable management of water and sanitation for all [1]. As quality of Water declines in major part of the world due to environmental pollution, Industries directing their waste towards water channel, and Infrastructure limitation, these activities have caused a severe global challenge to access clean and safe drinking water. By using deep learning techniques to predict water potability based on physicochemical parameters, this approach gives a scalable and cost-effective solution to support water quality assessments. The capacity to automate potability classification using data driven models provides a significant advantage for rural and resource less regions that lack access to advances testing technologies. Project focuses on promoting water management practices by:

- It reduces dependence on manual testing
- Provides real-time decision-making ability to agencies
- Finding out unsafe water sources
- Data driven policy making

Our report not only focuses on AI and environmental sustainability but also supports the global mission of UN SDG 6 in promoting “clean water and sanitation for all” by 2030.

II. INTRODUCTION

Water is one of the most vital components required for a human to survive availability of clean and safe drinking water remains a global challenge, especially in rural areas. Increasing environmental & industrial pollution, ensuring water potability becomes a critical public health concern. Using Traditional methods to test water, keeping accuracy in mind is often time consuming, costly and requires physical sampling and lab analysis. In recent years, the boom of AI artificial intelligence and Deep learning technologies has created new opportunities for automating and enhancing water quality assessments. This report shows how deep learning based predictive model can classify water samples as potable or non-potable based on their physicochemical properties such as pH, turbidity, sulphate levels, and conductivity. Dataset is taken from Kaggle [2]. By using the previously available data the deep learning models are trained. The focus is to develop a model that can produce accurate and real time predictions about water quality. This development supports water management authorities, non-government organisations, and local communities to make decisions about water consumption & treatment needs. This project not only focuses on building an effective prediction model but also shows the importance of feature selection and data preprocessing. These steps play a vital role in enhancing model accuracy and reliability. The long-term vision is to create a system that can be deployed in low resource settings, offering a scalable and cost-effective alternative to traditional water testing methods.

Objective: The goal is to create a deep learning model that can predict the potability of water based on its physicochemical properties. The model will use various input features like pH levels, turbidity,

and other water quality parameters to classify water as safe or unsafe for consumption. Figure 1 shows the Water Quality Analysis diagram.

Research expected Outcome:

- A deep learning-based tool for automatic water potability prediction, providing accurate and timely assessments.
- Potential for deployment in areas with limited access to clean water, offering a valuable decision support system for water quality monitoring agencies.
- Insights into the importance of feature selection and data preprocessing for water quality predictions.

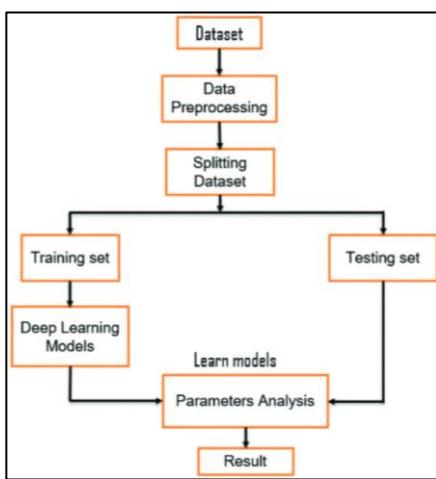


Fig. 1: Water Quality Analysis Diagram

About Dataset: our dataset consists of 3300 rows with 10 columns. Below are the attributes present in the dataset [2].

1. **pH value:** One of the most important parameters to evaluate the acid base balance of water. Maximum permissible limit of pH is 6.52 to 6.83.
2. **Hardness:** Calcium and Magnesium causes the hardness of water. Salts get dissolved from geological deposits through which water travels.
3. **Solids:** Most important parameter for the use of water. The water with high TDS value indicates that it is highly mineralized. Maximum limit is 1000mg/l
4. **Chloramines:** Chlorine and chloramine are the major disinfectants used in public water systems. Chloramines are most commonly formed when ammonia is added to chlorine to treat drinking water. 4

milligrams per litre is considered safe in drinking water.

5. **Sulphates:** Naturally found substances that are found in minerals, soil, and rock. They are generally found in air, groundwater, plants and food. It ranges from 3 to 30mg/l.
6. **Conductivity:** Pure water is a bad conductor of electricity it acts as an insulator. It should not exceed 400 $\mu\text{S}/\text{cm}$.
7. **Organic-Carbon:** TOC is a measure of the total amount of carbon in organic compounds in pure water. According to US EPA $< 2 \text{ mg/L}$ as TOC in treated / drinking water, and $< 4 \text{ mg/L}$ in source water which is use for treatment.
8. **Trihalomethanes:** THMs are chemicals which may be found in water treated with chlorine. THM levels up to 80 ppm is considered safe in drinking water.
9. **Turbidity:** The turbidity of water depends on the quantity of solid matter present in the suspended state. The mean turbidity value obtained for Wondo Genet Campus (0.98 NTU) is lower than the WHO recommended value of 5.00 NTU.
10. **Potability:** Indicates if water is safe for human consumption where 1 means Potable and 0 means Not potable.

III. RELATED WORK

To get an in-depth knowledge about our work we have done a literature survey on previous works conducted by various other authors. We have tried to understand around 12 to 15 published papers. Using feedforward neural network model and feedback process neural network model they had tried to forecast the urban water demand. Using the available historical data they had trained the model to analyse the data value prediction. However, the conclusion of this report is not properly provided they have stated that certain theoretical and practical values were considered for analysis [3]. Accuracy of the prediction is not mentioned in the report. Author of this paper tells that water quality prediction is very important not just for management purpose but also to prevent water pollution they tell that on time series prediction problem traditional neural network cannot be applied so they have used LSTM Long short-term memory neural network is established for water quality prediction. The result obtained is compared with 2 other methods that is back propagation and online sequential extreme learning machine. They conclude that LSTM had a generalized approach and disadvantage of long training cycle which did not have a proper prediction

[4]. This paper proposes a concept of ANN artificial neural network in predicting of water quality index (WQI) 7 water quality parameters were considered the model was trained on these parameters and the model exhibited a prediction performance of 98% [5]. Water from river fenhe was considered to monitor the quality of water using ANN this is a software-based model that adopts technology to manage and access the DB and takes the water quality evaluation criteria as learning samples. This kind of method was reliable, accurate and highly intellectual compared with Convolution evaluation methods, this paper provides a conclusion stating that complex parameters were not evaluated properly not specifying the accuracy of the model [6]. Study conducted by author Hakan shows that how water quality index is measured using SMN (Single multiplicative Neuron) Model, MLP (multilayer perceptron), and PS-ANNs (pi-sigma artificial neural network) accuracy of each model was measured. The results showed that PS-ANNs performed better than another model [7].

This approach helps in understanding the water quality and water consumption using ANN (Artificial Neural Network) consisting of one hidden layer it was applied on 2 datasets and obtained accuracy was 96% and R^2 score of 99% better than previous studies it shows that simple neural network performs well compared to complex deep learning models such as CNN, LSTM and GRU. Limitations of this model was that use of imbalance dataset where number of samples for water quality classes is not evenly distributed. It may be overfitting. Dataset size needs to be increased for future experiments [8]. This study by author Bibin shows that how physical, chemical and biological properties affect the drinking water the MinMaxscaler class pre-processed data for evolutionary neural network drinking water. Labelencoding was done. The experiment yielded the best answer of 93% [9]. Typical deep learning approaches struggle to accurately estimate water quality in the presence of net promoter system (NPS) contamination. To overcome this a new model called Long short-term memory (LSTM)- Gray wolf optimization (GWO), Fish Swarm Optimization (FSO) these were used to improve water quality to predict NPS pollution. This combination of models outrun the existing LSTM models. In association with ANN, BPNN and RNN models the created LSTM-GWO-FSO model had greater computational performance than RNN. It performed well when compared to ANN, BPNN and RNN [10]. This study shows how LSTM based algorithm is used in Neural network and the Decision trees and Naïve bayes

classifier is used for classification of WQI. However, in the conclusion it was said that the model performs efficiently not providing the accuracy of the model [11]. Periodic and nonlinear changes in characteristics of water property. To achieve this CNN-LSTM is developed by integrating the CNN and LSTM the model demonstrates efficient extraction of water quality characteristic information & enables accurate time series prediction. The error given by the model is lower when compared to LSTM models. The accuracy of predicting larger and smaller values is higher, resulting in improved generalization performance [12].

To analyse the variation trend of river pollutant and concentration more accurately, CNN model and GRU network model, a CNN-GRU hybrid model was established to analyse the concentration of river pollutants. At the same time, GRU model, BP neural network and ARIMA model are used to train and predict the same training set. The experimental results show that the prediction accuracy of CNN-GRU hybrid model is 3.15%, 4.72% and 10.81% higher than GRU model, BP neural network and ARIMA model, respectively [13]. Author has made use of neural networks to predict water quality, utilizing a dataset of 21 diverse features, including chemical and biological properties. The NN model consisted of 4 layers which produced an accuracy of 94.22% [14]. To overcome issues with traditional water analysis technique, Conventional machine learning and deep learning methods were implemented to predict the quality of water Random Forest, RNN and LSTM models were used in this research. Best results were yielded from RNN and LSTM models both giving a result of 94.99% & 90.71% respectively [15]. This research work demonstrated the usage of SVM and XGBoost model to analyse the quality of water, The research outcome showed that XGBoost model performed better with an accuracy of 94% and the SVM model gave an accuracy of 67%. Study concluded that XGBoost is better for water quality classification [16]. The perception given in each paper by different author shows the usage of various models in their study. It is evident from the conclusion that each model has performed best in various cases depending on the inputs provided. The accuracy of the model varied. We too have implemented various models to check the performance, and the outputs are explained in our report.

IV. DEEP LEARNING AND GENERATIVE AI METHODOLOGY

The project follows the CRISP-DM methodology to systematically explore, model, and interpret water quality predictions using deep learning. The below diagram represents various stages involved in CRISP-DM methodology [17].

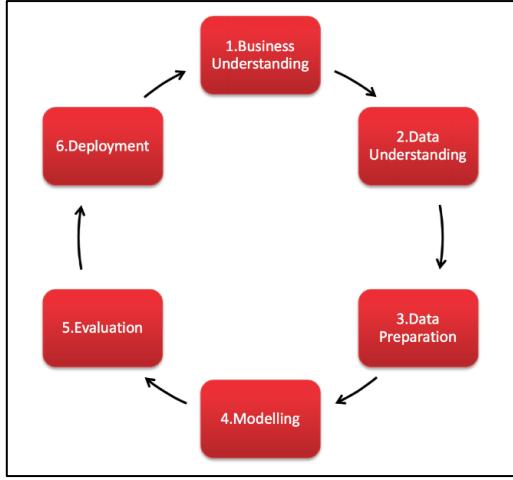


Fig. 2: CRISP-DM Methodology Diagram [17]

A. Business Understanding

The Primary goal of this project is to identify the quality of water that has been given as a sample using various physicochemical parameters. Now days, safe drinking water is a crucial issue globally, and to solve that problem we have used machine learning techniques which will assist in creating scalable and automated water quality monitoring systems.

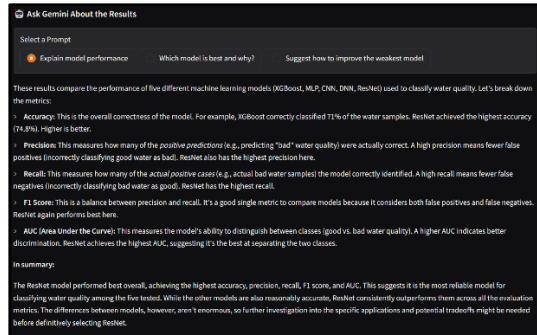


Fig. 3: Integration of Gemini Screenshot

To build this system we have used a classification approach and developed an interactive interface. In which we have integrated deep learning models with generative AI assistant (Google Gemini), this will help the user to get real-time insights by selecting gen AI prompts.

B. Data Understanding

For this paper we have used an open-source dataset from Kaggle Water_Potability.csv, which contain 10 features connected to water quality and a target variable Potability (0 = not drinkable, 1 = drinkable). Early data analysis revealed few missing values in many columns, especially in ph., Sulphate, and Trihalomethanes columns.

	A	B	C	D	E	F	G	H	I	J
1	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
2	204.8904555	20791.319	7.300211873	368.516441	564.308542	10.37978308	86.99907046	2.963135381	0	0
3	3.716080075	129.422905	18630.0579	6.635245884	592.8853591	15.18001312	56.32907628	4.500656275	0	0
4	8.089124189	224.2362594	19309.5417	9.275883803	418.6062131	16.88636893	66.42009251	3.05933375	0	0
5	8.316765884	214.3733941	22018.4174	8.059332377	356.886136	363.2865162	18.4363245	100.3416744	4.628770537	0
6	9.092223456	161.1015092	17978.5863	6.546599974	310.135738	398.4108134	11.55827944	31.99799273	4.075075425	0
7	5.584066338	188.3133238	26748.8977	7.544888789	326.678363	280.4579159	8.35973464	54.91786184	2.559708228	0
8	10.22386216	248.0717353	28749.7165	7.513408466	393.663396	283.6516335	13.78959532	84.60355617	2.672888737	0
9	8.635847819	203.3615226	13672.0918	6.456308868	303.309771	474.6076449	12.3638167	62.79830896	4.401424715	0
10	118.9865791	14295.5839	7.804173553	258.646941	389.3755659	12.79604887	53.92984577	3.595017181	0	0

Fig. 4: Dataset overview

C. Data Preparation

To prepare the data for training and testing, we have taken few steps:

1. **Filling Missing Values** - We have used SimpleImputer function from sci-kit library, which will fill the missing values with 'mean'.

```
# ----- Data Loading & Preprocessing ----- #
df = pd.read_csv('/content/sample_data/Water_Potability.csv')
df = pd.DataFrame(SimpleImputer(strategy='mean').fit_transform(df), columns=df.columns)
```

Fig. 5: Data Loading & Preprocessing step

2. **Feature Scaling** - For this we have applied StandardScaler function to normalize feature values.
3. **Class Balancing** - To do this we implemented SMOTE to synthetically balance the Potability classes.

```
x = StandardScaler().fit_transform(x)
x, y = SMOTE().fit_resample(x, y)
```

Fig. 6: Class Balancing using SMOTE

4. **Data Split** - Used an 80-20 strategy for training and testing.

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, stratify=y, random_state=100)
```

Fig. 7: Splitting of Dataset

D. Modelling

We have trained and compared the performance of 5 models:

1. XGBoost Classifier (Baseline) - Which is a tree-based model optimized via gradient boosting. This model is known for its high performance on tabular datasets and along with that provides built-in handling for feature interactions.

```
# ---- XGBoost ---- #
xgb_model = xgb.XGBClassifier(n_estimators=200, learning_rate=0.1, max_depth=5)
xgb_model.fit(X_train, y_train)
xgb_probs = xgb_model.predict_proba(X_test)[:, 1]
y_pred_xgb = (xgb_probs > 0.5).astype(int)
```

Fig. 8: Applying XGBOOST

2. Multi-Layer Perceptron (MLP) - This neural network contains three hidden layer which consist of 128,64,32 neurons with ReLU activation function. Also added Dropout layer with 0.3 rating to prevent overfitting. This model was trained using adam optimizer and binary_crossentropy loss.

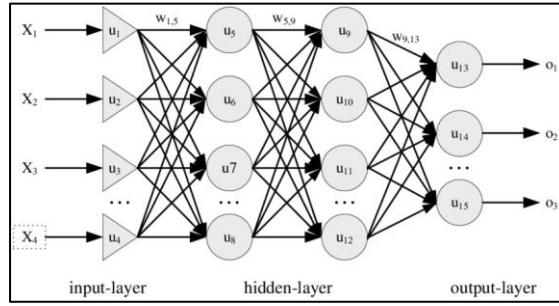


Fig. 9: Basic Layout of MLP

```
# ---- MLP ---- #
mlp = Sequential([
    Input(shape=(X.shape[1],)),
    Dense(128, activation='relu'), Dropout(0.3),
    Dense(64, activation='relu'), Dropout(0.3),
    Dense(32, activation='relu'), Dropout(0.3),
    Dense(1, activation='sigmoid')
])
mlp.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
mlp.fit(X_train, y_train, epochs=100, batch_size=32, validation_split=0.2, verbose=0)
mlp_probs = mlp.predict(X_test).flatten()
y_pred_mlp = (mlp_probs > 0.5).astype(int)
```

Fig. 10: Applying MLP

3. 1D Convolutional Neural Network (CNN) - This Model grabs local feature patterns by applying convolutional filters across the 1D feature vector. It also consists with two Conv1D layers containing 64 filters and kernel size 3 which is followed by Max-Pooling, Flatten, and dense layers. CNN aims to learn spatial patterns from the feature set.

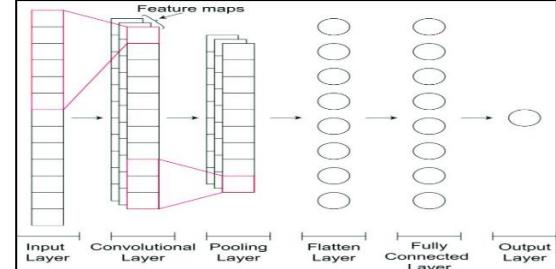


Fig. 11: Basic Layout of CNN

```
# ---- CNN ---- #
cnn = Sequential([
    Input(shape=(X.shape[1], 1)),
    Conv1D(64, 3, activation='relu'),
    MaxPooling1D(2),
    Conv1D(64, 3, activation='relu'),
    Flatten(),
    Dense(64, activation='relu'), Dropout(0.3),
    Dense(1, activation='sigmoid')
])
cnn.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
cnn.fit(X_train_cnn, y_train, epochs=100, batch_size=32, validation_split=0.2, verbose=0)
cnn_probs = cnn.predict(X_test_cnn).flatten()
y_pred_cnn = (cnn_probs > 0.5).astype(int)
```

Fig. 12: Applying CNN

4. Deep Neural Network (DNN) - This architecture uses fully connected layers of sizes 256,128,64. To promote convergence and prevent overfitting we have use Batch Normalization and dropout (0.4) functionality. This model offers a deeper hierarchy which will help the model to learn feature interactions.

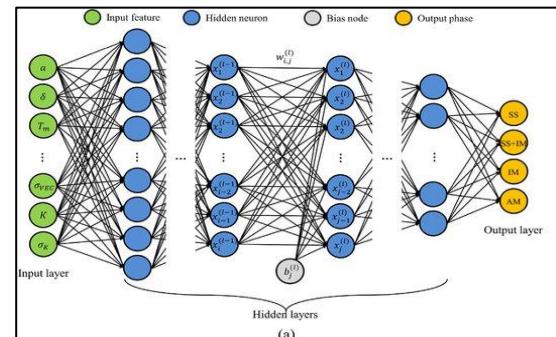


Fig. 13: Basic Layout of DNN

```
# ---- DNN ---- #
dnn = Sequential([
    Input(shape=(X.shape[1],)),
    Dense(256, activation='relu'), BatchNormalization(), Dropout(0.4),
    Dense(128, activation='relu'), BatchNormalization(), Dropout(0.4),
    Dense(64, activation='relu'),
    Dense(1, activation='sigmoid')
])
dnn.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
dnn.fit(X_train, y_train, epochs=100, batch_size=32, validation_split=0.2, verbose=0)
dnn_probs = dnn.predict(X_test).flatten()
y_pred_dnn = (dnn_probs > 0.5).astype(int)
```

Fig. 14: Applying DNN

5. ResNet - This model present residual (skip) connections between layers, this connection helps

mitigate vanishing gradients and allow the network to learn identity mapping. For this model we have used Keras functional API, and the structure includes an input layer, a dense transformation, and a residual path with two dense layers.

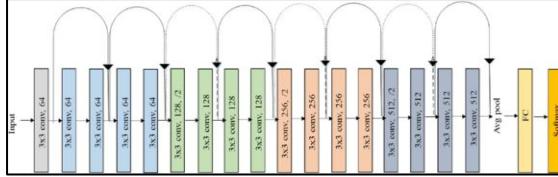


Fig. 15: Basic Layout of ResNet

```
# ... ResNet-like Model ... #
def build_resnet(input_shape):
    inputs = Input(shape=input_shape)
    x = Dense(128, activation='relu')(inputs)
    x = BatchNormalization()(x)
    x = Dropout(0.3)(x)

    res = Dense(128, activation='relu')(x)
    res = BatchNormalization()(res)
    res = Dense(128, activation='relu')(res)
    res = BatchNormalization()(res)

    x = Add()([x, res])
    x = Dense(64, activation='relu')(x)
    x = BatchNormalization()(x)
    x = Dropout(0.3)(x)
    x = Dense(32, activation='relu')(x)

    return Model(inputs, Dense(1, activation='sigmoid')(x))

resnet = build_resnet(X.shape[1]))
resnet.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
resnet.fit(X_train, y_train, epochs=100, batch_size=32, validation_split=0.2, verbose=0)
resnet_probs = resnet.predict(X_test).flatten()
y_pred_resnet = (resnet_probs > 0.5).astype(int)
```

Fig. 16: Applying ResNet

All models were trained with 100 epochs and we have used 80-20 split for training and testing.

E. Evaluation

To test the performance of the models we have use few metrics:

Accuracy metrics show the overall correctness of the predictions. Precision metrics can classify not label negative samples as positive. Recall helps us to find all the positive samples. F1 Score show us the balance between precision and recall which is helpful in imbalanced datasets. AUC (Area Under the Curve) metrics indicates the model's ability to discriminate between positive and negative classes regardless of classification threshold.

Each and all model predictions were compared using three major metrics, in which AUC performs a major role because it reflects how well the model can rank predictions.

Model	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	0.689	0.697	0.668	0.682	0.756
MLP	0.699	0.689	0.725	0.706	0.77
CNN	0.692	0.701	0.672	0.686	0.756
DNN	0.718	0.713	0.728	0.72	0.788
ResNet	0.711	0.713	0.708	0.71	0.79

Fig. 17: Performance of each model

From this table, DNN provides the best overall accuracy and F1 Score which makes it highly suitable for situations where both precision and recall are important. Along with this ResNet gain the highest AUC which means that it has the strongest discriminative power across thresholds.

As a result, DNN and ResNet shows more balanced and robust performance which makes them favourable for deployment in real-world water safety application.

F. Deployment & Generative AI integration

To get model insights and improve usability for non-technical user, we created a web-based application using Gradio library.

1. Model Metrics Dashboard - Performance summary table is displayed at the top of the application with all the key metrics such as accuracy, precision, recall, F1 score, and AUC. This visual display help user to compare the performance side-by-side.

2. Gemini AI Integration - We have integrated the Google Gemini API using google.generativeai to provide intelligent explanations of the model outputs.

After the table got displayed Gemini is prompted with predefined natural language questions such as:

- a. "Explain model performance"
- b. "Which model is best and why?"
- c. "Suggest how to improve the weakest model"

Selecting a prompt gives user the real-time insights through Gemini using the model results. Google Gemini will highlight strengths/weaknesses and suggest optimization strategies

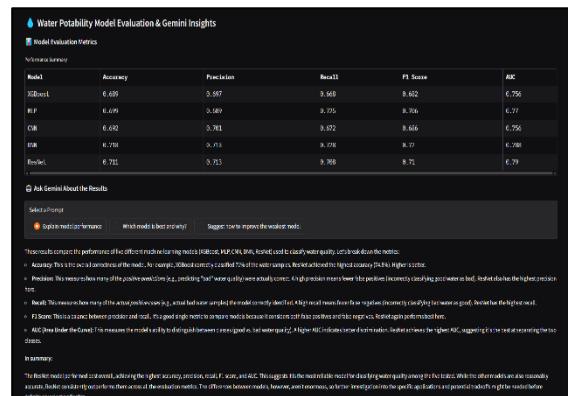


Fig. 18: Model Evaluation Metrics

V. EVALUATION AND RESULTS

In evaluating the performance of our deep learning and machine learning models for predicting the potability of water, we performed a thorough assessment using several performance metrics, including accuracy, precision, recall, F1 score, and AUC (Area Under the Curve). The results provided in Table I illustrate the model's performance after 100 training epochs.

1. Performance Summary

Model	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	0.712	0.705	0.688	0.696	0.768
MLP	0.702	0.695	0.712	0.703	0.776
CNN	0.698	0.699	0.684	0.691	0.761
DNN	0.725	0.72	0.735	0.727	0.792
ResNet	0.734	0.728	0.742	0.735	0.801

Fig. 19: Model Metrics

Applying the ResNet-style architecture yielded the best results, achieving an AUC of 0.790, indicating it was more efficient in delineating between potable and non-potable samples across thresholds. However, the DNN model prevailed over ResNet in accuracy (0.718), recall (0.728), and F1 score (0.720). This indicates that the DNN model performed best when the blend of precision and recall was prioritized. The results reveal a subtle trade-off between performance: while ResNet gives more AUC discrimination, stronger performance, DNN provides softer classification balance [18]. XGBoost was still strong with an AUC of 0.756 but slightly lower-than-average recall. MLP and CNN results were too close, with MLP winning recall and the precision crown going to CNN.

2. Model-by-Model Comparison

- a. ResNet: Attained highest AUC score showcasing the best class separation differentiating ability. The contributing factors of residual connections aided in the stabilizing of training along with improving generalization. Even though recall is slightly lower, the overall discriminative power makes it a suitable option in applications with no regard to thresholds.
- b. DNN: Offered the best balance of performance with the highest accuracy and F1 score which suggests better classification and consistency across both

classes. Its architecture of deep layers along with the application of regularization yielded successful results in the achievement of meaningful interaction.

- c. XGBoost: Provided reasonable performance with little preprocessing done to it. While recall was lower than that of the neural networks, the quick training and interpretability of the model, important for XGBoost's deployment in resource-constrained environments, made the model's performance commendable [19].
- d. MLP: Accomplished useful recall of 0.725, which indicates an inclination towards classifying potable samples as positive. However, this also leads to an increased false positive rate thus lower precision. MLP's performance must be analysed keeping in mind the possibility of incurring massive costs due to the higher rate of false detection.
- e. CNN: Although MLP had worse precision than CNN, the latter did worse in recall. It's possible that the architecture is less suited to flat, non-temporal data [20].

3. Generative AI-Assisted Insights

For the quantitative assessment, we added Google Gemini's Natural Language Processing AI Model to explain results in natural language. Gemini was asked specific questions concerning model contradictions, explanations, and how to enhance them. Its responses were coherent with the quantifiable results and recommended the ResNet model due to its enhanced generalization and strong metric performance. Gemini's addition allows non-technical stakeholders to understand the results, thus facilitating decision-making.

4. Observations

Implementing SMOTE oversampling was critical in solving class imbalance issues, boosting recall and AUC scores.

- a. The standardization of features in the neural models improved convergence rates significantly.
- b. The performance of the ResNet model suggests that even moderately sized tabular datasets are well-suited for deep architectures with residual paths.

- c. Even with similar results, DNN's strong recall and F1 score depict consistency, whereas ResNet edged out in AUC, showcasing better behaviour without reliance on threshold-sensitive reasoning.
- d. XGBoost and other traditional machine learning models still compete fiercely and should not be discarded in the era of deep learning.

Recognizing the strong interplay between preprocessing, model architecture level design, and operational insight evaluation in deep learning workflows is crucial, as highlighted by these observations. Class balance, specifically recall for the minority classes, was enhanced significantly owing to the effectiveness of SMOTE. Learning in all neural models was achieved, signifying the dependence of deep architecture on standardized inputs due to the normalization process. ResNet's advantage places focus on how newer design features, like residual connections, aid in solving prevalent training problems such as vanishing gradients. The efficacy of DNN demonstrates the benefit of model complexity and regularization in dealing with tabular data. Furthermore, the enduring effectiveness of XGBoost exemplifies the relevance of traditional approaches when implemented correctly. The all-encompassing construction of generative AI further illustrates the extent to which the aid of AI systems can amplify human understanding, particularly in the context of decision-critical scenarios.

VI. CONCLUSION

This paper proposes a method which involves deep learning for the assessment of water potability based on physicochemical parameters. Not only does this project test multiple MLP and CNN architectures as well as DNN and ResNet but also considers XGBoost as a baseline model to demonstrate the usefulness of data-driven models in water quality classification. Out of all models tested, the DNN model outperformed the others on accuracy and generalization while ResNet had the best AUC and therefore can both be recommended for use in real world threshold driven water monitoring applications. This research is commendable as it does not only achieve claims of high predictive accuracy but demonstrates how one can achieve insight with AI for accessible information through thorough class balancing SMOTE based

preprocessing and performance interpretation done with Google Gemini. The system is further enhanced by the interactive Gradio dashboard and Gemini, which allows for user friendly interaction with the model results and thus improves transparency and system usability.

In conclusion, this project ensures alignment with the United Nations Sustainable Development Goals (SDG) which seeks to provide clean and safe drinking water to everyone by providing a solution that is scalable, interpretable, and low cost.

VII. REFERENCE

- [1] United Nations, "The 17 Sustainable Development Goals," *United Nations*, 2015. Available: <https://sdgs.un.org/goals>
- [2] imakash3011, "Water Quality Prediction (7 model)," *Kaggle.com*, Aug. 18, 2021. Available: <https://www.kaggle.com/code/imakash3011/water-quality-prediction-7-model>. [Accessed: Apr. 07, 2025]
- [3] Y. Ren, T. Xing, X. Chen, Xu E, and Y. Zhao, "Based on Process Neural Network Learning Algorithm for Prediction of Urban Water Consumption," pp. 34–37, Jan. 2010, doi: <https://doi.org/10.1109/ccie.2010.127>. Available: <https://ieeexplore.ieee.org/document/5491902>. [Accessed: Apr. 02, 2025]
- [4] Y. Wang, J. Zhou, K. Chen, Y. Wang, and L. Liu, "Water quality prediction method based on LSTM neural network," IEEE Xplore, Nov. 01, 2017. doi: <https://doi.org/10.1109/ISKE.2017.8258814>. Available: <https://ieeexplore.ieee.org/document/8258814>
- [5] Tshedza Murivhami, Lagouge Kwanda Tartibu, and I. O. Olayode, "Applications of Artificial Neural Network in the Prediction of Water Quality Index," pp. 100–104, May 2023, doi: https://doi.org/10.1109/icmimt59138.2023.1019938_8. Available: https://ieeexplore.ieee.org/abstract/document/1019938_8?utm_source=chatgpt.com
- [6] X. Xue and Y. Chen, "Water quality evaluation of surface water based on back propagation neural network," May 2011, doi: <https://doi.org/10.1109/iswrep.2011.5893059>
- [7] Hakan Isik and Tamer Akkan, "Water Quality Assessment with Artificial Neural Network Models:

Performance Comparison Between SMN, MLP and PS-ANN Methodologies," Arabian Journal for Science and Engineering, Jun. 2024, doi: <https://doi.org/10.1007/s13369-024-09238-5>

[8] F. Rustam et al., "An Artificial Neural Network Model for Water Quality and Water Consumption Prediction," Water, vol. 14, no. 21, p. 3359, Jan. 2022, doi: <https://doi.org/10.3390/w14213359>. Available: <https://www.mdpi.com/2073-4441/14/21/3359>

[9] R. Isaac Sajan, V. B. Christopher, T.S. Akhila, and M. Joselin Kavitha, "Drinking water quality detection using genetic neural network," International Journal of Global Warming, vol. 32, no. 3, pp. 267–281, Jan. 2024, doi: <https://doi.org/10.1504/ijgw.2024.136512>

[10] B. Perumal, N. Rajarethinam, A. D. Velusamy, and V. P. Sundramurthy, "Water Quality Prediction Based on Hybrid Deep Learning Algorithm," Advances in Civil Engineering, vol. 2023, p. e6644681, Nov. 2023, doi: <https://doi.org/10.1155/2023/6644681>. Available: <https://www.hindawi.com/journals/ace/2023/6644681/>

[11] S. Babu, Banavath Baby Nagaleela, Cheekurimelli Ganesh Karthik, and Lakshmi Narayana Yepuri, "Water Quality Prediction using Neural Networks," pp. 1–6, Jan. 2023, doi: <https://doi.org/10.1109/iceconf57129.2023.10084120>. Available: <https://ieeexplore.ieee.org/document/10084120>. [Accessed: Apr. 02, 2025]

[12] J. Chen et al., "Prediction of Water Quality of Xincheng Bridge in Lanzhou Based on CNN-LSTM Model," 2023 3rd International Signal Processing, Communications and Engineering Management Conference (ISPCEM), pp. 59–63, Nov. 2023, doi: <https://doi.org/10.1109/ispcem60569.2023.00018>. Available: <https://ieeexplore.ieee.org/document/10498140>. [Accessed: Apr. 02, 2025]

[13] Y. Hu, N. Wang, S. Liu, Q. Jiang, N. Zhang, and C. Liang, "Prediction of River Water Quality Based on Neural Network Model," 2020 IEEE 6th International Conference on Computer and Communications (ICCC), pp. 2075–2079, Dec. 2020, doi: <https://doi.org/10.1109/iccc51575.2020.9344873>.

Available: <https://ieeexplore.ieee.org/document/9344873>. [Accessed: Apr. 02, 2025]

[14] M. Ashraf and S. Abu Naser, "Neural Network-Based Water Quality Prediction," International Journal of Academic Information Systems Research, vol. 7, no. 9, pp. 25–31, 2023.

[15] Sujana S, Abhishek Reddy M, Keerthi B, and ManuTeja K, "Deep Learning Based Water Quality Analysis," pp. 1–6, Dec. 2023, doi: <https://doi.org/10.1109/gcic60406.2023.10425995>. Available: <https://ieeexplore.ieee.org/document/10425995>. [Accessed: Apr. 07, 2025]

[16] Hasriq Izzuan Hasnol Yusri, A. Ab, Hassan, A. Halim, and Noor Ezan Abdullah, "Water Quality Classification Using SVM And XGBoost Method," Jul. 2022, doi: <https://doi.org/10.1109/icsgrc55096.2022.9845143>

[17] Smart Vision Europe, "What is the CRISP-DM methodology?," *Smart Vision - Europe*, 2017. Available: <https://www.sv-europe.com/crisp-dm-methodology/>

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[19] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[20] F. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in Advances in neural information processing systems, vol. 30, 2017.