WATER MANAGEMENT

An Engineering Project in Community Service

Phase – III Report

Submitted by

1. Aditya Jaiswal – 20BCE10822

2. Anirudh Praveen – 20BOE10059

3. Chandrika Sharma – 20BCE11032

4. CVSS Yamuna – 20MIM10078

5. Vaibhav Agrawal – 20BCY10090

6. Bhavik Bhosale – 20BOE10009

7. Manan Laddha – 20BCY10115

8. Priyam Shriniwas – 20BCY10095

in partial fulfillment of the requirements for the degree of

Bachlore of Engineering and Technology

VIT Bhopal University
Bhopal
Madhyapradhesh

<<May, 2023>>

Bonafide Certificate

Certified that this project report titled “Water Management” is the Bonafide work of “Aditya Jaiswal – 20BCE10822, Anirudh Praveen – 20BOE10059, Chandrika Sharma – 20BCE11032, CVSS Yamuna – 20MIM10078, Vaibhav Agrawal – 20BCY10090, Bhavik Bhosale – 20BOE10009, Manan Laddha – 20BCY10115, Priyam Shriniwas – 20BCY10095” who carried out the project work under my supervision.

This project report (Phase III) is submitted for the Project Viva-Voce examination held on …………..

15/05/2023

Supervisor

Comments & Signature (Reviewer 1)

Satisfactory 15/05/2023

Comments & Signature (Reviewer 2)

Objective:

The objective of our project is to help detect water levels of the lakes in Bhopal city by taking certain factors into consideration. The basis of this work is Artificial Intelligence and its extensive applicability. The inclusion of atmospheric factors such as rainfall and salt levels as well as prediction models of AI such as perceptron makes this project unique in its own ways.

Introduction:

The need for proper water management is not an unknown aspect of sanitation and hygiene. In fact, it is an alarming one. City corporations are now focusing on water management by incorporating chemical treatments, water recycling, salt removal treatments, and many more. While these could be temporary or sometime-backlashing, simple measures such as keeping the water levels in check could be the answer.

However, checking the water level in regular intervals can prove to be a challenge to manual labor. An AI driven system is an efficient solution to this. A well-programmed computer software that has the intelligence to check the water levels on a timely basis can help.

The need to find a solution to checking water levels begs the question- *Why check the water levels at all?* Unmonitored water levels lead to clogging, overflowing, stagnation, depletion, and many such problems. These in return cause breeding grounds for mosquitoes, flooding, unbalanced salinity, algae formation, mineral sedimentation, etc. An easy solution is to monitor the water levels regularly.

Motivation:

The purpose of the proposed method is to achieve balanced water levels. This helps maintain the hydro-chemical balance, avoid water disasters such as floods, prevent formation of algae-fungal decomposition and alkaline sedimentation. The poorly managed lakes in Bhopal have become an environmental liability. Public health is a major concern today. Swift and efficient measures are needed to ensure health and safety of the city. Our drive to help find a solution to these problems is the focus of our project.

Existing Work:

The challenges pertaining to city water levels have prevailed for a while now. Attempts to resolve the issues have been made before our proposal as well. The objectives of this particular attempt were:

- To provide effective methods of water-quality prediction,
- To put forth a platform-independent method if water-quality prediction
- To provide agencies and corporations with suggestions and steps to maintain the ecological balance in water.

The ideology of the existing work is the implementation of ε-SVR model. The purpose of ε-SVR training method is to solve the regression function in a given sample space. The model considers the following factors- pH, suspended solids, dissolved oxygen, ammonia nitrogen, chemical oxygen demand, and biochemical oxygen demand. The model claimed to have the capacity to simulate water-quality with accuracy, however, with a considerable number of errors. SVR stands for Support Vector Machine. Support Vectors are the data points that lie on the edges of margins. These are the points that are essential for computing the margin value.

While SVM is an advanced supervised learning method, it has certain disadvantages to its applications, some of which are:

- SVM is not suitable for larger datasets.
- Overlapping target classes are a challenge to SVM.
- If the number of features for each data point exceeds the number of training data samples, SVM will be unable to perform.

We aim at overcoming these challenges by employing a different strategy to solve the issue.

Architecture:

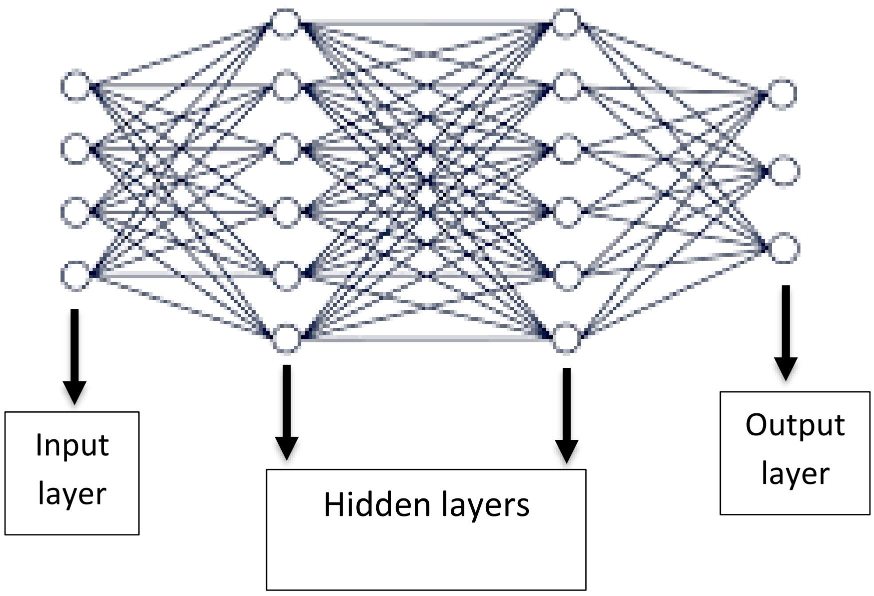
The foundation of our proposed model is Artificial Neural Networks. ANNs are algorithms replicate the functioning of a human brain and are used to create models for complex patterns and prediction problems. The working of ANNs is quite similar to that of Biological Neural Networks, but not identical. ANNs are designed to accept, comprehend, analyze, process, and generate numerical and structured data. The structure of an artificial neuron is inspired by biological neurons. Similar to the soma in a biological neuron, the artificial neuron has input nodes to receive input signals, which are alike the dendrites. The weights of artificial neuron have the same functioning of the synapse in the biological neuron. Post-processing, the output of the artificial neuron is the replica of the axon of the biological neuron.

The major differences between artificial neurons and biological neurons are the learning methods and activations. While synapses are physical links between biological neurons, weights join the nodes of one layer of the ANN to the next layer. The strength of the link is dependent on the value of the weight. A biological neuron learns through its soma that contains a nucleus to help process impulses. This happens due to *synaptic plasticity* which is described as ability of synapses to adjust their strength of functionality based on the reaction needed for a certain change. In ANNs, the parallel learning methodology is *feed-forward* or *backpropagation* in which weights are adjusted in accordance with the error or the difference between the obtained and predicted outcomes. Another major difference is the Activation. In biological neurons, activation is carried out by the *firing rate* when the impulses are strong enough to be communicated to the brain. However, in the case of an artificial neuron, a mathematical function called the *activation function* is used to map the data from the input node to output node, thereby executing activations.

A typical ANN has three layers in its architecture- the input node, the hidden layer, and the output layer. As the name suggests, the input node accepts inputs from users and passes it on to the hidden layer where the input data is processed. The hidden layer can be thought of a distillation layer that identifies patterns in the input that are relevant to the task and processes them to obtain an outcome. It enhances and fastens the efficacy of the network by identifying the key information from inputs. The processed data is further transferred to the output layer. The output layer generates the result after analysis. With the help of activation functions, ANNs convert inputs to more applicable and usable output.

The principle type of ANN used in this project is the Multi-Layer Perceptron. An MLP is an ANN with many layers. Essentially, the architecture of an MLP is same as that of a typical ANN, except there are multiple hidden layers in an MLP. This helps the network to learn complicated non-linear relations and patterns between input and output data. The data flows in a unidirectional manner, through the successive layers of the network.

The goal of training an MLP is to find the optimal weights and biases so as to minimize the value of error. Error is the value of the difference between the obtained and predicted outcome values.



Working principle:

We collected data from the Indian Meteorological Department (IMD) in Bhopal City and Municipal Corporation unit. Our datasets comprise of three factors that play a major role in the prediction process. The factors are-

- Precipitation levels from the weather reports obtained from IMD
- Water levels from the Municipal Corporation (Motia Talab)
- Salt levels from the collected samples

The precipitation levels were collected in *mm* units for seven months of the year 2020, January to July- about 212 days. Some days show drastic change in the amount of rainfall received by the region whereas somedays have a negligible scale. The water levels in the lake show minor changes somedays, although with change in the amount of rainfall there is respective change in the water levels as well.

The salt analysis was conducted in the laboratories of VIT Bhopal. A sample was collected from the sources- *Neelam Park, Kali Mandir, and Police Head Quarter.* Water has many salts in its composition some of which are chlorides, sulfates, nitrates, and phosphates. Each sample was further divided into five test tubes with equal measure of water. Each test tube was tested for a certain salt using the respective reagent. As the salinity increases, the density of the water decreases thus making the water level compact. On the converse, as the salinity of water decreases, the density increases thereby making the water expand. This imbalance in salt levels of water creates an imbalance in the water levels as well.

The methods of implementation used in this project are pertaining to the field of Artificial Intelligence. We incorporated Multi-Layer Perceptron. An MLP is a typical Feedforward mechanism based model. A feedforward neural network involves a sequence of layers oh functional compositions. Every layer results in a set of vectors which are then fed as input to the successive layer for further processing. There are three types of layers- Input layer, Hidden layer, and Output layer. The general application of a multilayer perceptron is for cases wherein the value to be predicted has a vague or extremely large number of factors involved. In our case, the level of water in a certain water body is determined by a vast number of factors. The two factors considered, precipitation and salinity, play a vital role.

While there are several prediction methods using feedforward mechanism, we chose the MLP because it enables users to increase depth and width of the network, and also enhance the flexibility of the function approximation. Along with these reasons, an MLP is rather simple to execute and efficient at the same time. It can quickly establish the relationships between the factors and the value to be predicted. For our project, the relations are as follows-

w – water level; p – precipitation; s – salinity

$w \propto p$

$w \propto s$

Meaning, water level (w) is directly proportional to both precipitation (p) and salinity (s).

The technical implementation in this project involves the use of the AI programming language called *Python*. It is the most widely used language today, especially to implement AI models. The code begins with importing the libraries needed to perform the computations.

The libraries are-

- Numpy
- Pandas
- Keras
- Minmaxscaler

Numpy is the library that is used for numerical computations.

Pandas is the library used in data analysis and manipulation.

Keras is the library for building the neural network.

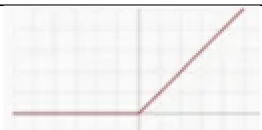
sklearn the library that aids in machine learning and data preprocessing.

Matplotlib is the library used for data visualization.

Then the dataset for water levels was imported from its respective *csv* file using pandas. The dataset contains factors such as precipitation, salinity, and evaporation that affect water level. The net dataset was divided in the ratio of 80:20 using the split function. The input values are normalized with the help og the minmaxscaler function. This function sets all the factors in the range of 0 and 1. The neural network has 64 neurons in the first layer, 32 in the second layer, and 1 neuron in the final layer. The first 80% of the data was used to train the model and the remaining 20% was used to test the model. The model was trained by using the training dataset (80%) for 100 epochs and has a batch size of 32. The model was assessed using the training and testing datasets individually with the help of the evaluate() function. Once evaluated, the value of the water level is predicted using the predict() function. To compare the predicted values and the original values, we used matplotlib to plot a comparative graph thereby visualizing the data. We generated two graphs- one was for the comparative outlook and the other was to visualize the training and testing loss over the respective epochs.

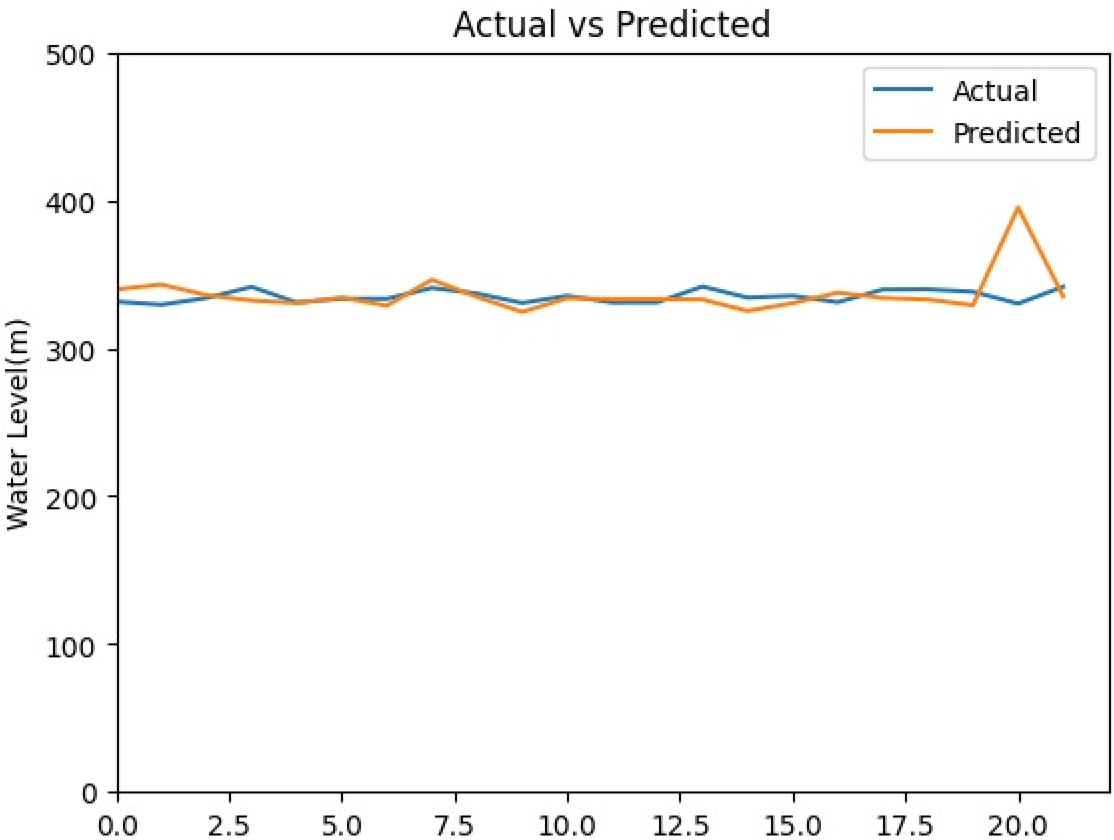
Every ANN needs a certain function that *feeds* the data from one layer to the next, thus carrying out the learning process. This function is called the Activation function. It is a mathematical function and it decides if a neuron should be activated or not. It acts as a virtual switch that turns a node on or off based on the necessity of the node's or layer's functionality. We need an activation function to prevent linearity, meaning it ensures the output is not directly proportional to the input.

In our project, we used the ReLU function. ReLU stands for Rectified Linear Activation Function. It is a piecewise linear function. This means that the output is not monotonous, instead it is determined by the input value. It functions on the basis of chunks of various values of the input. The range of ReLU values range from [0, infinity). The uniqueness of ReLU is that it only accepts positive values as input. If a negative value is given, it will return 0 as the output. This helps our purpose as precipitation and salinity levels have only positive readings. This ensures that incorrect inputs are avoided thereby saving time of execution.

A.function	Plot	Equation	Derivative
ReLU		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$

Result:

This model was trained with real-time datasets to predict water levels in water bodies. Initially we considered 80% of the collected dataset to train the model. The model was trained to be able to consider all factors that it has learnt of. The testing data comprised of 20% of the dataset. Upon testing, we obtained the following results.



The x-axis depicts the data points used from the dataset collected. A data point is the information fed into the machine learning model. The accuracy of a model is determined by the volume and number of data points. The volume of a data can be defined as the number of various inputs we give the model.

The y-axis depicts the water levels against the variables precipitation, salinity, and evaporation.

The graph above gives the visualization of the difference between the actual value of water levels and the predicted value of water levels.

Our machine learning model API is now available on the Render platform, utilising the Flask and FastAPI frameworks to provide seamless access to our advanced machine learning capabilities. With this deployment, we provide a dependable and scalable API that allows for simple interaction with our model.

The Render platform assures our API's continual availability and scalability, allowing for seamless integration into a variety of applications. Our API interface makes user engagement easier by offering well-defined endpoints and simple request types. It is capable of performing a wide range of tasks, including image recognition and natural language processing, and delivers accurate results quickly.

Flask and FastAPI were chosen for their flexibility and efficiency. Flask, a lightweight web framework, provides a strong base, while FastAPI, a high-performance framework, easily manages large traffic and concurrent queries. They work together to provide peak performance even in the most challenging situations.

Our ML model API fits with the cloud environment smoothly by delivering on the Render platform. Render automates difficult deployment processes, load balancing, and scaling, allowing us to focus on providing our users with a dependable and efficient API.

In summary, our ML model API, built with Flask and FastAPI and hosted on the Render platform, provides a simple and robust way to access our machine learning capabilities. Our API provides easy integration into apps, allowing customers to gain useful insights from their data, thanks to its user-friendly interface, scalability, and dependability.

Individual Contributions:

Anirudh Praveen (20BOE10059) and Bhavik Bhosale (20BOE10009):

Anirudh traveled to Bhopal and collected a few samples of water from the lake to assess the area's water's quality. Both the upper lake and the lower lake each had three samples taken. The biotechnological laboratory has been used to store the samples for the further testing of water qualities and it’s components. Bhavik then tested the salinity of those samples.

The salinity of water is measured to check if the water is safe and clean for irrigation and livestock, and in industrial settings to ensure that industrial discharge is free from high levels of dissolved salts. This testing is done by passing an electric current between the two electrodes of a salinity meter in a sample of water. The electrical conductivity of water samples is influenced by the concentration and composition of dissolved salts. The best Salinity level for the water sample is in the range of 1.024 – 1.026 (32 – 35 ppt). One of the most crucial factors is the physicochemical composition of the water. The quality of the water has a direct impact on the wellbeing, development, and production of the fish. The density of bacterial populations may be influenced by the many physic-chemical characteristics of water, including temperature, conductivity, pH, total dissolved solids (TDS), turbidity, dissolved oxygen (DO), hardness, chloride concentration, free CO_2, and total alkalinity. By slowing down the rate of food conversion, lowering weight gain, and decreasing the availability of soluble nutrients in the water body, disturbances in these parameters might have a negative impact on fish output. Human and animal's excreta are considered to be one of the common sources of pathogenic microorganisms in any ecosystem. It is a strong contributor to conductivity and helps determine many aspects of the chemistry of natural waters and the biological processes within them. Salinity, along with temperature and pressure, helps govern physical characteristics of water such as density and heat capacity. Therefore, it's crucial to do regular water tests. Hence, with this motive in our minds, we have done our testing.

Priyam Shrinivas (20BCY10095):

Exploring the application of ReLU and neural networks in specific domains or problem areas. For example, investigating the use of ReLU in reinforcement learning tasks or exploring its effectiveness in natural language processing tasks can provide insights into the strengths and limitations of ReLU in different application domains. Moreover, research can focus on understanding the robustness and generalization capabilities of ReLU-based networks, particularly in scenarios with limited labeled data or in the presence of adversarial attacks. Artificial neural networks (ANNs) have proven to be effective in water level prediction tasks. By leveraging historical data and capturing complex relationships, ANNs can make accurate predictions for applications such as flood forecasting and water resource management. The process involves collecting relevant water level data, preprocessing the data, designing the network architecture, training the model using optimization algorithms, evaluating its performance, and deploying it for real-time prediction. Future work can focus on integrating additional data sources, exploring advanced neural network architectures, analyzing the impact of input features, and developing ensemble models to further enhance prediction accuracy and robustness.

Vaibhav Agrawal (20BCY10090):

As part of our water level prediction project, I was responsible for coding the multilevel perceptron model with ReLU activation function in Python using the Keras library. I first imported the necessary libraries and datasets, and preprocessed the data by splitting it into training and validation sets, and standardizing the input variables. I then designed the neural network architecture using the Keras Sequential model and added multiple dense layers with ReLU activation function. To prevent overfitting, I included a dropout layer and early stopping callback during the training phase. I also optimized the model's hyperparameters, such as the learning rate, batch size, and number of epochs, using grid search and cross-validation techniques. After training the model, I evaluated its performance on the validation set and fine-tuned it to achieve higher accuracy. In the end, the multilevel perceptron model with ReLU activation function successfully predicted water levels with high accuracy, making a valuable contribution to our project.

Manan Laddha (20BCY10115):

In our recent project aimed at predicting water levels, my individual contribution involved creating a multilevel perceptron model with a ReLU activation function. This involved designing the neural network architecture, selecting the appropriate number of layers and nodes, and implementing the activation function. The ReLU activation function was chosen due to its ability to provide faster and more accurate convergence during the training phase. I will also fine-tune the hyperparameters of the model, such as the learning rate and batch size, to optimize its performance. Through extensive testing and evaluation, the multilevel perceptron model with ReLU activation function was found to be highly effective in accurately predicting water levels. Overall, my contribution in creating this model was crucial in achieving the project's objectives and delivering accurate predictions for effective water level management.

CVSS Yamuna (20MIM10078):

My role in this project was to test algorithms and choose the most apt model for our data and motive. AI is a vast domain, there are multiple models that can solve a purpose. However, it is very important to choose the one that fits our requirements. Water level prediction specifically needs a model that can work on large data in a systematic way in order to ensure the relations between all factors are maintained throughout. An artificial neural network was the best fit for our purpose. As compared to a regular linear program, an ANN can execute tasks with greater efficacy. In cases of decline in a certain parameter, the ANN does not collapse. It continues to build relevant patterns and analyze the data. MLP is also considered an ANN except with multiple hidden layers. These hidden layers help process and analyze the input data. More the number of hidden layers, better the analysis, and hence more accurate predictions. Like choosing the right training model, pairing it with the correct activation function was also a challenge. Testing all possible activation functions was necessary. The ReLU function was finalized based on pros such as it does not activate all neurons simultaneously; it activates neurons based on the necessity, and that it does not accept negative values as inputs. This would save a lot of time during real-time execution. Overall, I have learnt a lot during this project. The knowledge I have gained helped me write the report of this project. My knowledge in the domain of Artificial Neural Networks has increased to a considerable extent.

Chandrika Sharma (20BCE11032):

I contributed to the following areas during the course of our project:

UI Design: I was in charge of designing the app's user interface (UI) and developing a platform for users to interact with the API to obtain results. I used Streamlit, a powerful data app development tool, to create a visually stunning and user-friendly interface. I ensured that the UI design was consistent with the project's objectives and properly catered to the app's functionality by working closely with the team. I integrated the app's necessary components and capabilities to ensure a smooth and straightforward user experience.

Deployment: When the frontend was finished, I deployed it to Streamlit Cloud, making it available to people all around the world. This included configuring the app for the cloud and ensuring that it was optimised for the environment. I worked with our team's API developer to verify that the frontend effectively connected with the API to obtain results.

Conclusion: Finally, I believe that my efforts were critical to the success of the water level detection project. I provided users with a platform to engage with the app and access real-time updates and forecasts by designing a simple and user-friendly frontend using Streamlit. The app's deployment on Streamlit Cloud makes it easily accessible to people all around the world, and I am delighted to have played a vital role in its development. I learnt a lot about frontend development and deployment on cloud platforms, and I'm excited to use what I've learned to future projects.

<u>Aditya Jaiswal (20BCE10822):</u>

The goal of our research is to create an AI-based water level detection system for Bhopal's lakes. The project uses a variety of parameters to predict water levels, including rainfall data, water flow data, and historical lake data. The device will provide real-time water level data and will aid in the prediction of probable floods or droughts in the area.

My Contribution: I was instrumental in designing the project's AI model. My primary focus was on building the API for the AI model with the Flask and FastAPI frameworks. The API is in charge of receiving inputs from the model and returning outputs. I created the API model with a JSON structure to make it simple to generate outputs in the terminal. I also hosted and deployed the model on the Render platform, which is a scalable, cloud-based environment for hosting and delivering applications. I have tested the API endpoints to confirm the model's accuracy and stability.

Problems encountered: Creating an AI model presents its own set of obstacles. One of the most difficult tasks was ensuring that the model was accurate and could produce dependable results. We had to work with a significant amount of data and create an algorithm to efficiently process it. We also had to take into account weather patterns, geographical location, and other environmental elements that could alter lake water levels. Another difficulty we encountered was implementing the model on a cloud-based platform. We had to verify that the platform was dependable and capable of handling high quantities of traffic without experiencing any downtime. To ensure that it could handle queries rapidly, we had to optimise the model to make it lightweight and fast.

Conclusion: The project was a success in general, and we were able to create an AI-based water level detecting system capable of providing real-time updates and predictions. My involvement in the project was critical in the development of the API paradigm and its deployment on the Render platform. I received great expertise in constructing artificial intelligence models and deploying them on cloud-based platforms, which will be relevant in future projects.

Conclusion:

This project gave us a platform to grow and prove ourselves in our respective domains. Designing and maintaining water bodies can be a challenge and it needs accurate measures to keep the level of water in check. There have been multiple cases of problems caused by unbalanced water levels. To solve these by just predicting water levels in order to alarm ourselves when needed is the aim of this project. The use of several fields such as biotechnology, computer programming, and artificial intelligence to create a model that can be deployed easily by all users was a journey. The extent of learning we have experienced is unique and we look forward to working for and serving the community more.

Datasets, Tables, and Graphs:

1. Pre-Monsoon Salt Analysis Dataset:

   https://drive.google.com/file/d/1C8k-e71TMeBH1XH7TTEVhmmJ21QQ1Hwu/view?usp=share_link

2. Monsoon Salt Analysis Dataset:

   https://drive.google.com/file/d/1eTvsDhGbF1iPZtvl9GERnnoVwwYRe3x0/view?usp=share_link

3. Post-Monsoon Analysis Dataset:

   https://drive.google.com/file/d/1eXK8RBC4K6gglll8cPlpY7_SUNXnpltC/view?usp=share_link

4. Graphical Visualization of Salt Analysis:

   https://drive.google.com/file/d/1u-Sy-MhX92Bs7y8yxjYf_GgSekPx43Cu/view?usp=share_link

5. Water Levels Dataset:
   https://drive.google.com/file/d/1SH1Ynd2O2AZHXzGIvb0reg1MenwU_ag2/view?usp=share_link

6. Code and Implementation of the project:
   https://colab.research.google.com/drive/1Hlai6SRos9jsoTfyeKwQUA7NBWlCsy9r?usp=sharing

References:

https://pubs.usgs.gov/wri/1993/4209/report.pdf

https://link.springer.com/article/10.1007/s11269-014-0773-

https://github.com/topics/water-level

https://www.hindawi.com/journals/abb/2020/6659314/