##### 3. Methodology Adopted

3.1 Investigative Techniques (Justify the selected investigative technique for your project, 2-3 pages)

|  |  |  |  |
| --- | --- | --- | --- |
| **S.**  **No.** | **Investigative Projects Techniques** | **Investigative Techniques Description** | **Investigative Projects Examples** |
| 1 | Descriptive | Our system is based on speed and simplicity, and aimed at reducing the cost while keeping the functionality the same.  Therefore, having carefully selected  Features are a must.  For this, we explored scientific papers pertaining to water quality in order to see which features will help  Assess water quality.  Moreover, we also checked how these researches handled and automated the collection of data:  1. How their network was made (if any)  2. How they analysed the data, whether it was on site or if data was stored on a database and then analysed on a much more powerful system  3. What algorithms were used to anaslyse the water. | Some of these questions of ours were answered fully, while some left us unsatisfied with the process, which have been talked about more in the previous sections.  The networks usually consisted of a microprocessor on site to which sensor nodes were connected, these collected data and transferred it onto an online database for further processing.  The data was mostly analysed using machine learning methods such as SVM, kNN to classify water, it was not being done on-site. In most cases the data was being collected, uploaded to a computer for analysis using computationally expensive machine learning algorithms like the ones mentioned above. |
| 2 | Comparative | Our project is fundamentally a classification system based on assessing the water quality.  The system will collect data from the water via the help of sensors and then use the collected data to classify the water as suitable for irrigation and farming or unsuitable. This classification will be done on site in order to keep the system independent from the internet. If the water is suitable then it will be pumped to the fields, and if the water is unsuitable then the water will be pumped to a treatment facility or elsewhere.  So, the sooner that the water is classified, the less negative effect it will have on the overall yield of the fields. For example, if the water is classified as unsuitable in three seconds, and the flow of the water is, say, 100 cubic meters per second (100cms) then the fields will still get 300 cubic meters of unsuitable water due to a slower classification algorithm.  Moreover, if the water is suitable and the pump for the treatment facility is turned on beforehand, then 300 cubic meters of suitable water will be sent to the treatment facility, thereby wasting that water.  Finding the optimal machine learning algorithm is thus, imperative, for reducing the overall negative impact of unsuitable water on the fields, as well as reducing the wastage of suitable water.  Thus, for finding the best machine learning algorithm, a combination of Experimental as well as Comparative techniques were used. | Theoretically, k Nearest Neighbours is not a suitable algorithm for our use case as the processing will take long due to constant calculation of Medians and sorting.  Similarly, while pre-trained neural networks can work optimally, they will require a large memory to store weight parameters and activations as the inputs propagate through the network.  Our system uses the Arduino Uno which only has 32k bytes of flash memory.  Support Vector Machines will also require a large calculations, and similarly Random Forest classifiers will require space to store the forest created. The higher the number of trees, the more probabilities to store for majority voting.  By the process of elimination this leaves us with Logistic Regression, Optimized Logistic regression with Stochastic Gradient Descent as well as Decision Tree classifiers. |
| 3 | Experimental | In this technique we used different languages and technologies to see how they would affect the final product.  Beginning with the website, prototype pages were made using React, and for the backend flutter and Django were used.  From these Django was selected for the backend and instead of React, simple javascript has been used for the website as of yet.  Similarly, the classification algorithms were tested with random samples of data. However this was done using python, and more investigation will be done for this. | As stated above, the following algorithms were tested and were fast and required lower memory.  However, right now the memory and time usage has been based on python language.  The second phase of the experiment will be using Cython and storing the code onto the Arduino for calculation, or using full fledged C++ or C for the algorithms. |

3.2.

The Project focuses on determining quality of water and classifying water as good quality or bad quality for the purpose of irrigation as well as automatically diverting bad-quality water to the treatment pump, before going to fields.

The data of water would be collected through the sensors. The sensors used would collect data regarding:

• Ph value

• SAR

• Temperature and

• Turbidity

The SAR value will be specifically calibrated in conjunction with the Electronics (ECE) and Chemistry (CHE) departments of TIET, wherein the ECE department will be helping in the creation of the sensor, while the CHE department will be helping to analysis samples of water to calibrate the sensor.

The sensors used to calculate the pH value, temperature, turbidity etc. are available on various e-commerce platforms for purchasing at a reasonable cost.

The data obtained would be sent to a microcontroller through the help of a LORA module which would transmit data wirelessly through a WAN network with a receiver connected to a microcontroller.

The use of a wireless network is to give the user the freedom to place the sensors wherever the data can be obtained from. Each sensor will be connected to an Arduino which will subsequently be connected to a LORA transmitter. This will constitute a ‘sensor node’.

And each sensor node will transmit data to the main Arduino connected to a LORA receiver as well as the ESP2866 Wi-Fi module.

Once the main Arduino receives the data, then it will be written into the serial monitor of the Arduino uno. From there, the ESP2866 will read the data from the serial monitor and transfer it to the firebase database.

A machine learning model is uploaded on the microcontroller which would determine whether the water quality is suitable or not. The machine learning model would be selected after exploring various models and analyzing their performance based on accuracy. As stated above, time is of the essence and currently Logistic regression, optimized Logistic regression, and Decision tree classifier are in the run to be tested and used.

The data sent onto the Firebase will be displayed and available onto the user dashboard.

As stated before, the classification will be done on site in order to keep the system independent of the internet. This is another reason for the wireless network, since the sensor nodes can be placed at the source of water irrigation, such as a water pump, and the farmer can then place the receiver node near Wi-Fi. It must also be re-iterated that the dashboard is non-essential, our system will work and classify water without connection to the database as well.

Based on the result obtained the water would be directed either to a treatment plant or to the field. This will be done by turning on the respective pump, as well as turning off the other pump.

The raw data obtained will be uploaded on the dashboard where it can be analyzed/visualized. This way the user on logging in can view the data regarding water quality which helps him keep informed.

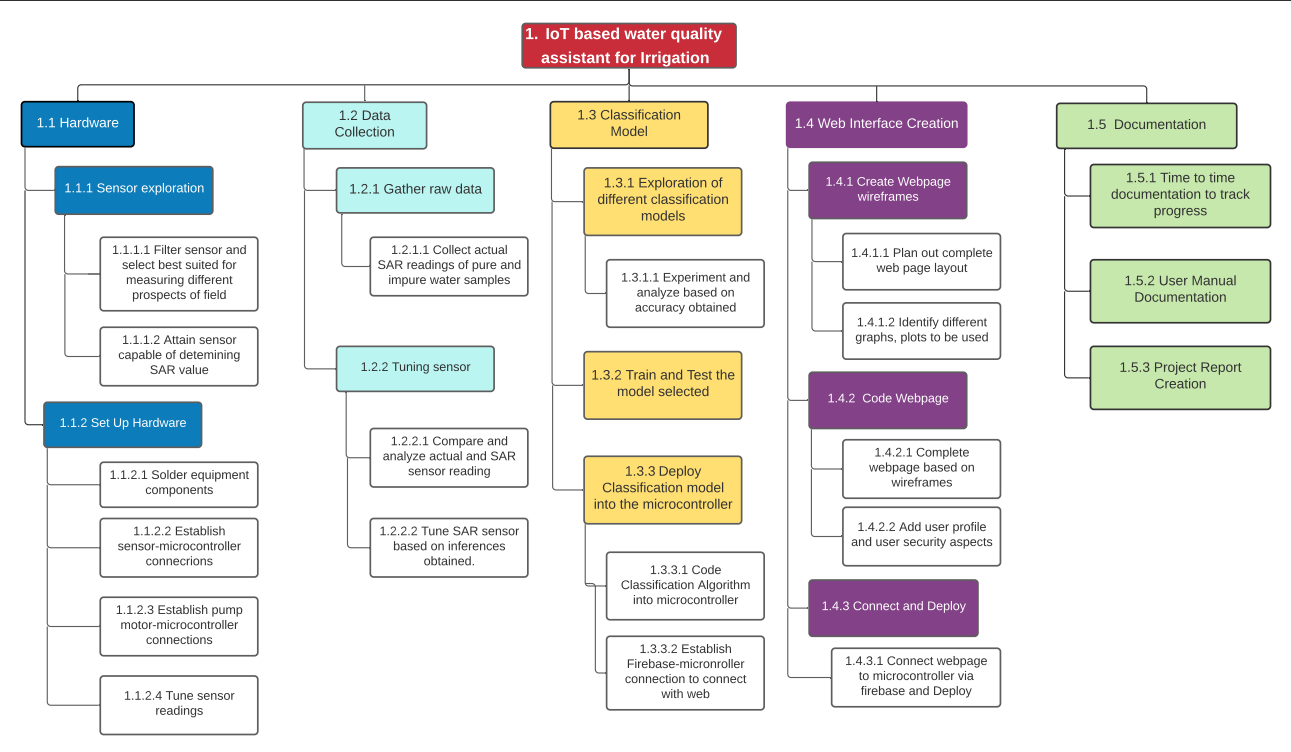
The webpage would contain the user security features so that the user can login and access data safely. The user would have to enter the login details or else the webpage would not show the data. The user would require to have an internet connection in order to access the webpage.

In this way this add-on feature of webpage will help the user understand the data regarding water better and help him economically

Moreover, while researching for this project, our team has realized that structured data for water analysis using Machine Learning algorithms is not widely available. Hence, we are in the process of adding a reinforcement learning model for which the user will have to add the yield of the crops after every season, and based on those results, changes to the classification algorithm can be made.

This data can then, be made available for anyone wishing to use it later for their own interests under the MIT License. However, this is still a hypothetical.

3.3Work Breakdown Structure (along with discussion on workable modules/products)



The work is divided into four sections, Hardware, Data collection, Classification Model and Web Interface Creation.

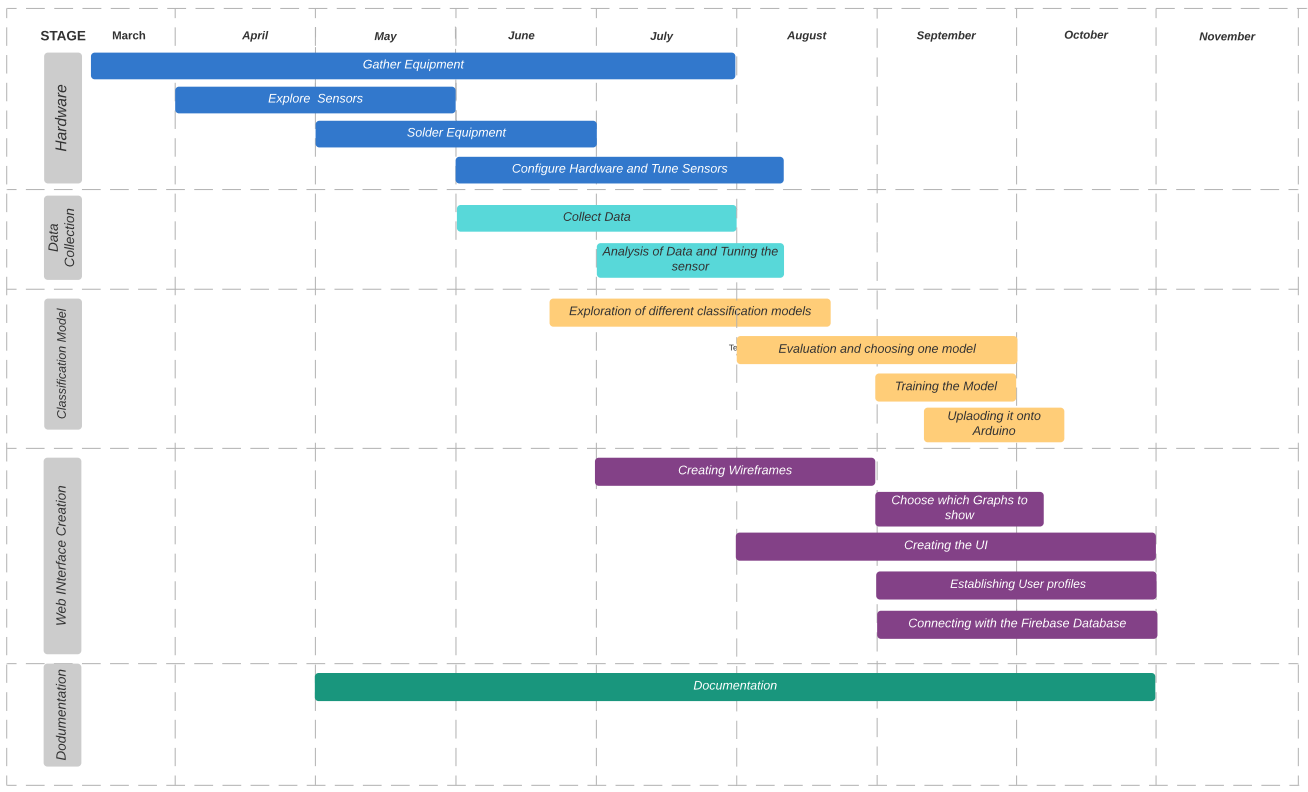
Out of these four, the hardware and Web interface have already been prototyped.

Initially the hardware has been tested using a single ultrasonic sensor in order to check the connection between the Firebase database and ESP2866. The data is collected by the sensor and written onto the Serial monitor. From there, the ESP2866 reads the data from the Serial monitor, and uploads it onto the Firebase database. This is done using Arduino libraries.

The database values are then fetched using python libraries and sent to Django views to be used to create graphs.

A landing page for the web interface has also been created.

The following Gantt chart is based on the above Work breakdown structure.



It can be seen from the Gantt chart that we are well ahead of schedule with only wireless transmission of sensor data, and classifying the water based on sensor data requiring work.

3.4 Tools and Technology

|  |  |
| --- | --- |
| Tool used | Reason |
| Esp2866 wifi module | Used to connect to Wi-Fi and upload sensor data onto the Firebase database |
| Arduino uno | Used for making the sensor and receiver nodes. The classification algorithm will also be uploaded onto the Arduino. |
| Firebase - NoSQL | Used to store user authentication data as well as sensor data |
| Sensors – all | Used to collect data from the water for analysis |
| Python | Used to compare various machine learning classification algorithms and used for the backend of the Website. |
| Django | Python Web Development framework used for the backend of the website. |
| Jquery | Used for AJAX requests to update the graphs in real time without reloading the website. |
| HTML | Used for the basic skeleton of the website |
| CSS | Used to stylize the website and dashboard |
| JavaScript | Used to provide functionality to the dashboard |
| Charts.js | Javascript library used for all the graphs |
| Figma | Online tool used to create the front end and prototypes of the website. |