JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Smart Irrigation System: Using IoT

[1]Prof. Nikhil Ingale, [2]Pranoti Patil, [3]Sarika Parvin, [4]Samreen Khan

[1]Assistant Professor, [2,3,4] Third Year Students
[1,2,3,4]Department of Civil Engineering,
[1,2,3,4]Rizvi College of Engineering, Bandra, Mumbai.

Abstract: This project aims to develop a IoT-based Smart Irrigation System with the help of a user-friendly interface for optimizes water usage by leveraging connected devices and sensors. This involves the monitoring of soil parameters such as humidity, temperature, and soil moisture, leading the user for efficient irrigation practices. This paper explores the system's significance and limitations, emphasizing its applicability across diverse domains—agriculture, small-scale irrigation, and roadways/highways. By utilizing IoT, the system makes real-time decisions regarding water supply. Additionally, the paper outlines the development of a user-friendly UI for seamless interaction and easy decision making.

***Index Terms* – IoT (Internet of Things), UI (User Interface), Logistic Regression and Soil Parameters.**

I. INTRODUCTION

Agriculture, as the largest livelihood provider in India, faces the critical challenge of meeting rising food demands due to population growth. To support increased agricultural production, efficient water management becomes paramount. Freshwater consumption for irrigation is a significant concern, with agriculture currently accounting for 83% of India's total water usage. Unfortunately, unplanned water utilization often leads to wastage, necessitating urgent solutions that minimize water loss without burdening farmers. Over the past years, farmers have increasingly adopted computer systems and software tools to organize financial data, track transactions, and monitor crop health. In the digital age, where information drives decision-making, agriculture has transformed into a data-intensive industry. Farmers must collect and analyze vast amounts of data from diverse devices, including sensors and farming machinery, to enhance production efficiency and facilitate informed communication.

The emergence of open-source Arduino boards and affordable moisture sensors presents an opportunity to create intelligent devices. These devices can continuously monitor soil moisture content and adjust irrigation in real time, ensuring optimal water usage for fields and landscapes. Additionally, this technology extends beyond traditional agriculture. Consider the application of smart irrigation to plants and trees alongside roadways and highways, where water conservation is equally crucial. This paper, we propose the development of a user-friendly interface that leverages IoT capabilities. By remotely monitoring irrigation systems based on sensor values, farmers and users can make informed decisions about watering. This system streamlines their work, promoting efficient resource utilization and sustainable practices.

II. AIM & OBJECTIVES OF STUDY

Aim: To design and implement an IoT-based Smart Irrigation System UI that optimizes water usage in agriculture and related fields, addressing the critical challenge of meeting rising water demands while minimizing water wastage.

Objectives:

- **Efficient Water Management:** Develop a system that monitors critical parameters (humidity, temperature, soil moisture) to ensure precise irrigation practices.
- **User-Friendly Interface**: Create an intuitive UI for farmers and users to remotely monitor and control irrigation systems based on real-time sensor data.
- **Extending Beyond Agriculture:** Explore applications of smart irrigation in diverse areas, including roadside plantations and highways.
- **Resource Conservation:** Minimize water loss by making informed decisions about irrigation, promoting sustainable practices.

III. REVIEW OF LITRATURE

Ayush Pande (2019) has proposed the use of Arduino Mega and sensors to monitor soil moisture and temperature, automating irrigation when moisture thresholds are met. This benefits farmers and nursery professionals by replacing manual methods. After considering various methodologies, the Evolutionary Prototype Methodology was chosen for its ease of understanding, rigid stages, and ability to easily implement changes. It allows for regular client feedback and prioritizes the most needed functionality. Other methodologies were rejected due to various reasons such as cost, lack of focus on design, and unsuitability for small projects. After

testing, the "PlantCommunicator" was developed successfully, proving to be cost-effective, easy to operate, and user-friendly, making it suitable for real-time scenarios and beneficial for farmers and agriculturists.
[1]

A.Anitha et.al. (2020) has suggested the following steps to apply the IoT based project using the available data and hardware components. The primary objective of their proposed study was to develop a framework for remotely monitoring soil moisture from distant locations and effectively managing soil moisture to optimize crop production. The proposed Internet of Things (IoT)-based system is designed to achieve this goal. The prototype framework allows real-time monitoring of agricultural land and ensures soil moisture maintenance. By implementing this concept, countries can transition toward sustainable agriculture. The framework is expected to generate and store records over time. While the core approach and control remain consistent, actual implementation would require adjustments to sensing components, technologies, and source code. The proposed system was demonstrated using ThingSpeak, an IoT analytics platform that aggregates, visualizes, and analyzes real-time data streams in the cloud. ThingSpeak also supports the execution of MATLAB code. This platform is particularly useful for prototyping and proof-of-concept IoT systems that involve analytics. In the proposed system, data is collected from various sensors, including soil moisture level, area temperature, air moisture, and water level. These sensors are connected to a breadboard, which interfaces with an Arduino board. The data is then transmitted to the Arduino Integrated Development Environment (IDE), where programmed instructions extract and process the data. If the data is invalid, the process terminates.[2]

M Gayathri et. al. (2021) in her work proposes a smart farming system using drip irrigation and IoT. The system uses sensors to monitor field conditions and groundwater levels, with data accessible via a webpage. Notifications are sent via SMS when the field is dry or groundwater levels drop. The system automates irrigation using a Node MCU, relay, and water pump. When soil moisture levels drop, the system automatically turns on the pump and turns it off when sufficient moisture is achieved. This method improves water and crop management by directing water to the plant roots and minimizing evaporation. The system also uses Wi-Fi and cloud computing for live monitoring, providing efficient and accurate data. Notifications about groundwater levels can be sent whether online or offline. This system enhances traditional agricultural irrigation processes by reducing water wastage through drip irrigation.[3]

Dr. C K. Gomathy et. al. (2022) has proposed that the Smart Irrigation System integrates moisture, humidity, and temperature sensors with a microcontroller. When soil moisture is low, the motor automatically turns on, and a notification is sent to the user. The system employs low-cost soil moisture sensors, temperature sensors, and Wi-Fi modules to continuously monitor the field. Data from these sensors are transmitted to an online server and stored in a database. Decision-making for irrigation occurs at the server, activating the motor based on soil moisture levels. Additionally, water level monitoring ensures timely notifications. Overall, this system efficiently manages irrigation, reduces water waste, and minimizes power consumption. By automating irrigation, it benefits farmers and conserves resources.[4]

IV. METHODOLOGY

A. Data Analysis of cotton crop

To analyse the data of cotton crop and its relation to moisture, temperature, and pump status. The data is obtained from a CSV file with 200 rows and 4 columns. The columns are crop, moisture, temp, and pump. The crop column has only one value, which is cotton. The moisture column represents the percentage of water content in the soil. The temp column represents the temperature in degrees Celsius. The pump column represents whether the pump is on or off.
Following steps are carried out for the testing and analysing of the test samples from the data available

- **Identifying Imbalance**: The first step is to identify if there is an imbalance in the data. In this case, the value_counts () method has already shown that there are 150 rows with pump value as 0 (off) and 50 rows with pump value as 1 (on), indicating an imbalance in the data.
- **Resampling Techniques**: To address this imbalance, resampling techniques can be used. There are two main methods:
  - **Oversampling:** This involves increasing the number of instances in the minority class by randomly replicating them in order to present a higher representation of the minority class in the sample.
  - **Undersampling:** This involves reducing the data points from the majority class to make it equivalent to the minority class. This method is usually used when the quantity of data is sufficient. By keeping all samples in the minority class and randomly selecting an equal number of samples in the majority class, a balanced new dataset can be retrieved.
- **Implementing Resampling:** In Python, the imbalanced-learn library can be used to apply these resampling techniques. It offers several methods for oversampling and under sampling.
- **Splitting the Data:** After balancing the data, it should be split into training and testing sets to evaluate the performance of the model.
- **Training the Model:** Post data balancing and splitting, the model can be trained using the balanced dataset. This ensures that the model is not biased and can generalize well when introduced to new data.

Data balancing should only be done on the training dataset to ensure the model generalizes well to unseen data. The test set should reflect the original distribution of the data to provide an accurate measure of model performance.

B. UI (user interface) Modelling

User Interface (UI) Design is a key aspect of digital products, focusing on creating accessible, understandable, and usable interfaces. UI includes the layout, aesthetics, and interactive elements of a digital interface, facilitating effective communication between users and the system.

UI Design Process: The process involves stages like research (understanding user needs, preferences, behavior), wireframing (outlining the layout and structure of the interface), prototyping (developing interactive prototypes), visual design (applying visual elements to enhance aesthetic appeal and usability), and testing and iteration (conducting usability testing and refining the design based on user feedback).

Key components of UI include visual design (layout, color schemes, typography, imagery), interaction design (arrangement of elements, navigation flow, feedback mechanisms), information architecture (content organization and structure), usability (simplicity, efficiency, learnability), and accessibility (features for diverse abilities). UI plays a significant role in User Experience (UX), focusing on the visual and interactive aspects of the interface. A well-designed UI contributes to a positive UX by making interactions intuitive, efficient, and enjoyable. UI shapes the user experience of digital products by creating visually appealing, intuitive, and functional interfaces.

V. RESULTS AND DISCUSSION

After preparing the data, some exploratory data analysis (EDA) is performed to understand the distribution and relationship of the variables.

In the univariant analysis as shown in the below fig. 5.1., `value_counts () ` revealed an imbalanced pump data with 150 off (1) and 50 on (0). `histplot ()`fig.5.2. showed a skewed moisture distribution (20%-40%) and a normal temperature distribution (25°C-35°C).

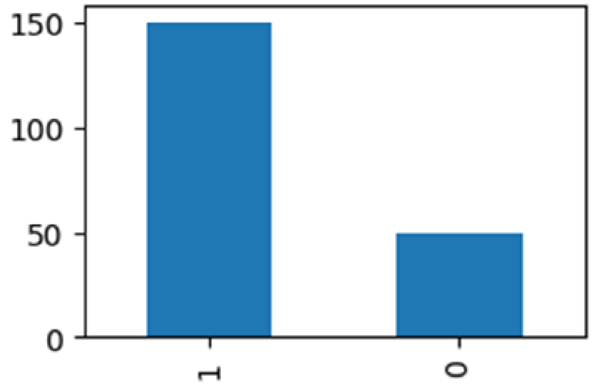

Fig. No. 5.1. Imbalanced data.

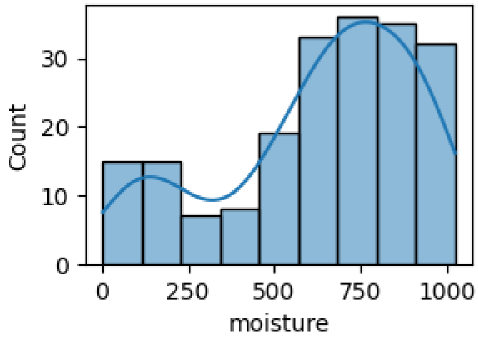

Fig. No.5.2. Moisture distribution with respect to temperature.

In the bivariate analysis fig 5.3., `barplot()` showed higher mean temperature and lower mean moisture when the pump is off (0), confirming a correlation between moisture and pump as shown in the figures below , and a negative correlation between moisture and temperature.

Fig. No. 5.3. Showing mean moisture and temperature when the pump is either on or off

`regplot()` was used to plot moisture and temperature , revealing a negative correlation. The regression line can predict temperature from moisture, and vice versa. Fig. no. 5.4 explains the scatter plot with the regression line provides a visual representation of the relationship between temperature and moisture, suggesting a weak negative correlation. However, the variability in the data indicates that other factors may also be influencing the temperature. This kind of plot is useful in fields like agriculture, where understanding the relationship between variables like temperature and moisture can help in making informed decisions.

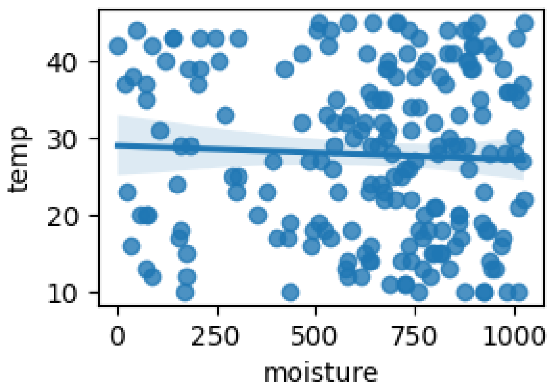

Fig.No.5.4. Regression line in temperature & moisture parameters

Data was balanced using the imbalanced-learn library and SMOTE, resulting in 150 samples for each pump value. The data was then split into training (80%) and testing (20%) sets. StandardScaler was used to standardize the data, with the transformed training and testing data stored in `input_train_scale` and `input_test_scale` respectively. A logistic regression model was trained and tested. The model's intercept and coefficients were accessed for interpretation. The model's performance was evaluated using a confusion matrix, accuracy score, and classification report, all of which indicated perfect prediction. A custom function, `make_classification`, was created to classify new samples based on moisture and temperature. The function was tested and correctly predicted the pump status.

To ensure the data was suitable for the model, it was standardized using the `StandardScaler` method. This process involved transforming the training and testing data to a standard scale by removing the mean and scaling to unit variance. The transformed data was stored in `input_train_scale` and `input_test_scale` respectively. Following the data preparation, a logistic regression model was trained using the standardized training data. The model's intercept and coefficients were accessed, providing valuable insights into the relationship between the features and the target variable. The model was then tested using the standardized testing data.

Hence the performance of the model was evaluated using several metrics. A confusion matrix was generated, providing a detailed view of the model's performance in terms of true positives, false positives, true negatives, and false negatives. Additionally, the accuracy score was calculated, which showed that the model had achieved perfect prediction, indicating its high reliability. To further extend the utility of the model, a custom function named `make_classification` was created. This function takes in new samples based on moisture and temperature and classifies them according to the trained model. The function was tested and was found to correctly predict the pump status, demonstrating its effectiveness. This entire process illustrates a comprehensive approach to data analysis and model building, from data balancing and standardization to model training, testing, and application.

VI. CONCLUSION

This project successfully developed an IoT-based Smart Irrigation System UI to optimize water usage in agriculture and beyond. It monitors humidity, temperature, and soil moisture for precise irrigation. The user-friendly interface allows remote monitoring and control based on real-time data. The project also addressed data imbalance in cotton crop data using resampling techniques and Python's imbalanced-learn library. The UI design process was thorough, considering key components to enhance user experience. The project achieved its objectives, minimized water loss, and promoted sustainable practices, with significant implications for meeting water demands and reducing wastage.

VII. FUTURE SCOPE

The IoT-based Smart Irrigation System developed in this project has demonstrated significant potential for future applications and improvements. Here are some possible future scopes and applications:

1. Expansion to Other Crops: While this project focused on cotton crops, the system can be adapted to other types of crops that have different irrigation requirements. This would make the system more versatile and applicable to a wider range of agricultural practices.

2.Integration with Weather Forecasting Systems: The system could be integrated with weather forecasting systems to anticipate rainfall and adjust irrigation schedules accordingly. This would further optimize water usage and prevent unnecessary irrigation.

3. Advanced Machine Learning Models: More advanced machine learning models could be implemented to improve the accuracy of predictions and make more informed decisions about irrigation.

4.Real-Time Alerts and Notifications: The user interface could be enhanced to provide real-time alerts and notifications to users when certain conditions are met or when manual intervention is required.

5.Application in Urban Landscaping: Beyond agriculture, the system could be used in urban landscaping to maintain public parks, gardens, and sports fields. This would contribute to water conservation efforts in urban areas.

Concluding, the IoT-based Smart Irrigation System has a wide range of potential future applications and improvements that could significantly contribute to water conservation and sustainable agriculture practices. The system's ability to monitor critical parameters, its user-friendly interface, and its versatility make it a promising solution for optimizing water usage in various fields.

REFERENCES

[1] Ayush Pandey (May 2019), Smart Irrigation System, Islington College.

[2] A.Anitha,Nithya Sampath M.Asha Jerlin (2020) , Smart Irrigation system using Internet of Things , International Conference on Emerging Trends in Information Technology and Engineering

[3] M Gayathri, D Arun Shunmugam (2021), A Ishwariya Smart Irrigation System using IoT, Department of Computer Science and Engineering, PSR Engineering College, Volume 9, Issue 3.

[4] C K Gomathy Sri Chandrasekharendra (January 2022), THE SMART IRRIGATION SYSTEM USING IOT, Saraswathi Viswa Mahavidyalaya University, Volume: 08 Issue: 10

[5] Srishti Rawal (February 2017), IOT based Smart Irrigation System, Department of Computer Science, VIT University, Volume 159 – No 8,