RES MILITARIS
REVUE EUROPEENNE D ETUDES EUROPEAN JOURNAL OF MILITARY STUDIES
Social Science Journal

Optimizing Smart Irrigation Systems with Predictive Analytics and Machine Learning for Enhanced Water Management

P. Vyshali[1], Y. Shiva Sai Teja[2], T. Venkatesh[2], O. Mahesh Kumar[2]

[1]Assist. Professor, [2]UG Scholar, [1,2]Department of Computer Science & Engineering (Data Science),

[1,2]Kommuri Pratap Reddy Institute of Technology, Ghatkesar, Hyderabad, Telangana.

ABSTRACT

Efficient water management is a vital component of contemporary agriculture, and the use of intelligent irrigation systems has garnered considerable interest for its ability to enhance water efficiency and maximize crop productivity. Predictive analytics is crucial in smart irrigation systems since it empowers farmers to make informed choices by utilizing both current and past data. Conventional irrigation techniques sometimes depend on predetermined schedules or human observations, which may not precisely reflect the true water needs of crops. In addition, certain current smart irrigation systems employ rule-based methodologies that just take into account fundamental environmental conditions, which may result in less-than-ideal water distribution. These strategies may not effectively adjust to dynamic environmental conditions and may not completely harness the capabilities of predictive analytics. This research introduces a predictive analytics method for efficient water management in intelligent irrigation systems. The method utilizes machine learning techniques and relies on temperature and humidity data obtained from Node-MCU. Machine learning algorithms are utilized to predict future water needs using up-to-date data, enabling the system to choose the most suitable irrigation schedule for each crop.

Keywords: Smart irrigation, Predictive analytics, Water management, Machine learning.

1. INTRODUCTION

Water management is a critical aspect of agriculture, particularly in countries like India where agriculture is a primary source of livelihood for a significant portion of the population. Efficient water management is essential to ensure food security, sustainable agricultural practices, and optimal use of water resources. With the advent of technology, intelligent irrigation systems have emerged as a promising solution to improve water efficiency and maximize crop yields. Historically, Indian agriculture has relied heavily on traditional irrigation methods, including surface irrigation techniques like flood irrigation, which are often inefficient and lead to significant water wastage. The Green Revolution of the 1960s and 1970s marked a significant shift in Indian agriculture, introducing high-yielding variety seeds, chemical fertilizers, and improved irrigation practices. However, these methods also increased water usage, leading to over-extraction of groundwater and depletion of water resources. In recent years, the Indian government and various organizations have recognized the need for sustainable water management practices. Initiatives such as the Pradhan Mantri Krishi Sinchayee Yojana (PMKSY) aim to enhance irrigation efficiency through the adoption of micro-irrigation techniques like drip and sprinkler systems. Despite these efforts, the adoption of advanced irrigation technologies remains limited due to factors such as high initial costs, lack of awareness, and insufficient technical knowledge among farmers.

Water Usage in Agriculture

- Agriculture accounts for approximately 80-90% of total water consumption in India.
- Traditional irrigation methods, which are used by the majority of farmers, have an efficiency rate of only 30-40%, leading to significant water losses.

Groundwater Depletion

- India is the largest user of groundwater in the world, with over 60% of irrigated agriculture and 85% of drinking water supplies depending on it.
- The rate of groundwater extraction is alarming, with some states like Punjab and Haryana experiencing severe groundwater depletion.

Adoption of Smart Irrigation Systems

- As of recent reports, the area covered under micro-irrigation (drip and sprinkler) is around 12.7 million hectares, which is only about 15% of the total irrigated area in India.
- The government aims to expand this area significantly to improve water use efficiency and crop productivity.

Technological Initiatives

- Various startups and tech companies in India are developing smart irrigation solutions that utilize Internet of Things (IoT) devices, sensors, and machine learning algorithms to optimize water usage.
- These systems can provide real-time data on soil moisture, weather conditions, and crop water requirements, enabling precise irrigation scheduling.

Relevance of Predictive Analytics and Machine Learning

— Predictive Analytics: Predictive analytics involves using historical and real-time data to forecast future events. In the context of smart irrigation systems, predictive analytics can help in predicting the water requirements of crops based on various factors such as soil moisture levels, weather conditions, and crop type. This enables farmers to make informed decisions about when and how much to irrigate, reducing water wastage and improving crop yields.

— Machine Learning: Machine learning algorithms can analyze large datasets collected from various sensors and IoT devices deployed in the fields. These algorithms can identify patterns and correlations in the data, leading to more accurate predictions of crop water needs. By continuously learning from new data, these systems can adapt to changing environmental conditions and provide optimal irrigation schedules.

— Node-MCU: Node-MCU is an open-source IoT platform that can be used to collect data from sensors measuring temperature, humidity, and soil moisture. This data can then be processed by machine learning models to predict future water requirements and optimize irrigation schedules.

2. LITERATURE SURVEY

According to the Food and Agriculture Organization (FAO) of the United Nations, it is estimated that around 70% of all water withdrawal worldwide is due to agricultural applications [1], contrasting the industrial sector at 20% with municipalities' local infrastructure for services and domestic water use taking the remaining 10%. This seems a logical percentage distribution given that around 2000 to 3000 L of water are required to grow food per person daily [2]. Nonetheless, what is more concerning regarding this volume of water is that 93% never returns to its original source, signifying an apparent complete loss of the resource. Irrigation efficiency refers to the ratio of water the crop uses to the total amount of water extracted from the source [3]. Different factors affect irrigation efficiency, like water run-off, evaporation, and deep percolation. Water efficiency mostly depends on the hydraulic infrastructure and irrigation method, while surface irrigation has a water efficiency from 50% to 65%, sprinklers range from 60% to 85%, and drip irrigation from 80% to 90% [4]. Surface irrigation

implies surface evaporation, which contributes to water loss. Sprinkler technology reduces water loss but, still, the applied water evaporates off the leaves of the crop canopy. In contrast, drip irrigation delivers water directly to the plant's root zone, reducing losses due to run-off and evaporation [5]. In any case, water efficiency can be considerably improved when a sensor-based smart irrigation system is installed over the hydraulic infrastructure [6]. Notwithstanding, food production is stated to rise in the following ten years and for many decades to come. In [7], the author states that the demand for food and agricultural products is projected to further increase by up to 70% by 2050 in order to satisfy the requirements for an estimated 10-billion-person population by then. That, in addition to the growing effect of climate change on water shortage worldwide, can have terrible consequences in the near future regarding resource allocation and availability for agricultural purposes. Vulnerable communities in arid regions would potentially suffer the consequences of water scarcity and global warming more [8]. Moreover, severe social conflicts have already occurred in rural communities due to the unfair assignation of water resources for agricultural activities [9]. Therefore, technology and data-driven solutions for water management are required to improve resource efficiency, reduce water waste, and contribute to sustainable agriculture practices [10]. The waste and overuse of water resources for crop irrigation is a relevant topic that has been addressed by precision agriculture from different perspectives [11]. In this sense, automatic irrigation systems aim to optimize water utilization while helping farmers to improve crop yields by providing the right amount of water, at the right time, in the right place in the field [12]. To control the amount of water used during irrigation, typically these systems conduct measurements of soil moisture levels (volumetric water content), environmental parameters (solar radiation, wind speed, air temperature, air humidity), and crop conditions (canopy temperature, chlorophyll content, trunk diameter).

3. PROPOSED METHODOLOGY

The comprehensive methodology outlines the step-by-step process for conducting research on predictive analytics for smart irrigation systems. It involves data collection, preprocessing, model training, evaluation, and ultimately, the application of predictive insights to optimize water management practices. This research can significantly enhance the efficiency and sustainability of water usage in agriculture and landscaping. Figure 1 shows the proposed system model. The detailed operation illustrated as follows:

Node MCU Dataset: The research begins by collecting data from Node MCU devices installed within the smart irrigation system. These devices likely capture data such as soil moisture levels, weather conditions, and potentially other relevant parameters. This dataset serves as the foundation for the subsequent analysis and modeling.

Data Preprocessing: Raw data collected from Node MCU devices may contain inconsistencies, outliers, or missing values. Data preprocessing involves cleaning and transforming the dataset to ensure it is suitable for analysis. This may include removing duplicates, handling missing data, and addressing outliers.

Label Encoding: In predictive analytics, it's crucial to convert categorical variables, into numerical values that machine learning models can understand. Label encoding assigns a unique numerical label to each category.

Defining of Training and Testing Data for Classification and Regression: The dataset is split into two subsets: one for classification tasks and another for regression tasks. Classification tasks may include predicting whether the irrigation pump should be turned on or off based on current conditions, while regression tasks may involve predicting the number of liters to be watered or the time required to supply water.

Data Splitting for Classification: For classification tasks, the dataset is further divided into training and testing sets. The training set is used to train the machine learning model, while the testing set is reserved to evaluate its performance. A common split might be, for example, 70% of the data for training and 30% for testing.

Data Splitting for Regression: Similarly, for regression tasks, the dataset is divided into training and testing sets. Regression models are trained on the training set and evaluated on the testing set to predict continuous values like the number of liters to be watered or the time needed for water supply.

Random Forest Classifier Model: For classification tasks (e.g., determining when to turn the pump on or off), a machine learning model like Random Forest Classifier is employed. Random Forest is an ensemble learning method that combines multiple decision trees to make accurate predictions. It's suitable for tasks where data may have complex relationships.

Random Forest Regression Model: For regression tasks (e.g., predicting liters of water needed or time for water supply), a Random Forest Regression model is used. Similar to the classifier, it's an ensemble method that can handle both linear and non-linear relationships in the data, making it effective for predicting continuous values.

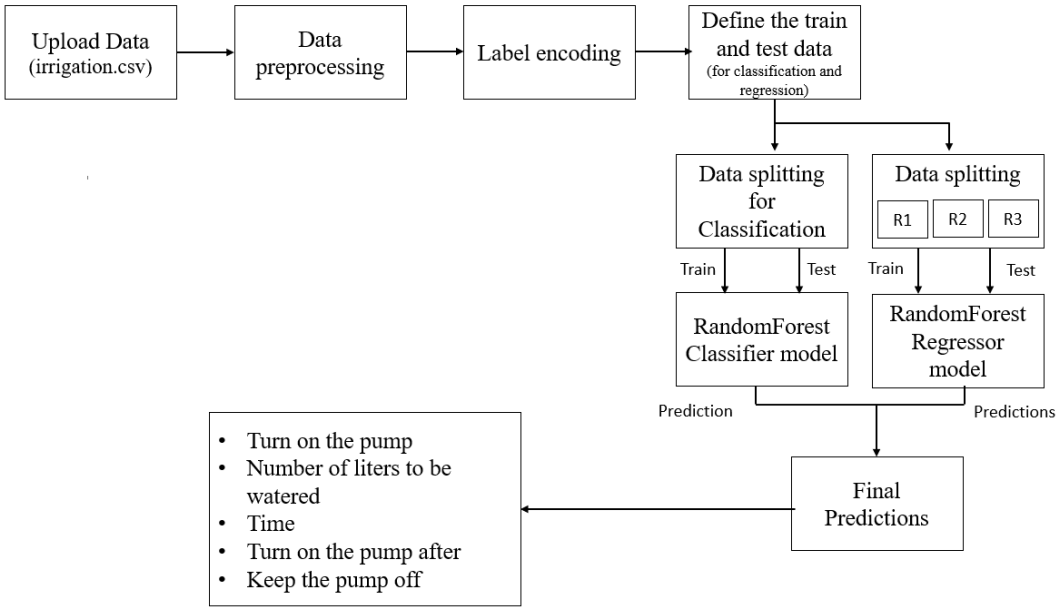


Figure 1: Proposed optimal water management for smart irrigation system.

Final Predictions: Once the models are trained and evaluated, they are used to make predictions. For classification, the model predicts whether to turn the pump on or off based on current conditions. For regression, it predicts the number of litters to be watered and the time required for water supply. Additionally, it can be used to determine when to turn the pump on and off based on the predicted supply time.

3.2 Random Forest Classifier

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree,

the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

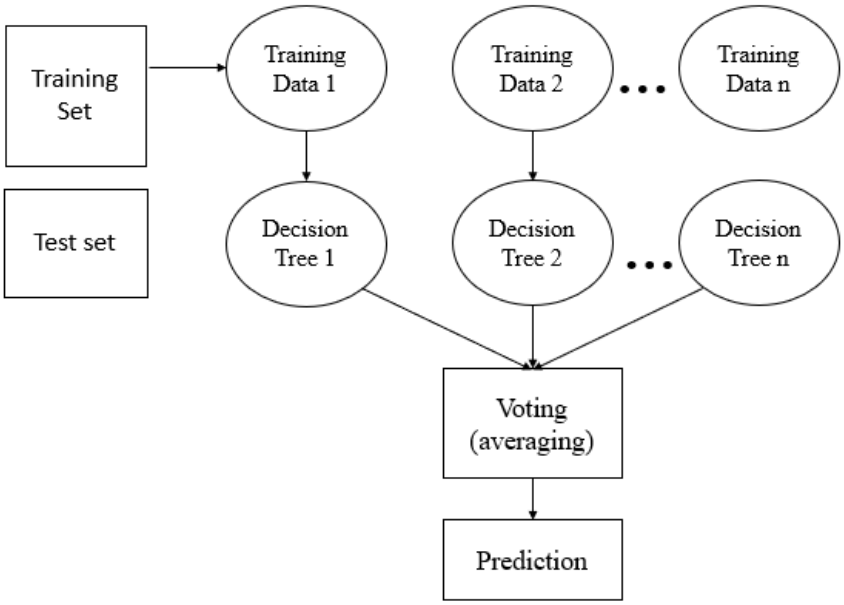


Fig. 2: Random Forest algorithm.

3.2.1 Random Forest algorithm

Step 1: In Random Forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

3.2.2 Important Features of Random Forest

- **Diversity**- Not all attributes/variables/features are considered while making an individual tree, each tree is different.
- **Immune to the curse of dimensionality**- Since each tree does not consider all the features, the feature space is reduced.
- **Parallelization**-Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.
- **Train-Test split**- In a random forest we don't have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.
- **Stability**- Stability arises because the result is based on majority voting/ averaging.

3.2.3 Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random Forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

Below are some points that explain why we should use the Random Forest algorithm

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

3.2.4 Types of Ensembles

Before understanding the working of the random forest, we must look into the ensemble technique. Ensemble simply means combining multiple models. Thus, a collection of models is used to make predictions rather than an individual model. Ensemble uses two types of methods:

Bagging– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest. Bagging, also known as Bootstrap Aggregation is the ensemble technique used by random forest. Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as row sampling. This step of row sampling with replacement is called bootstrap. Now each model is trained independently which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting is known as aggregation.

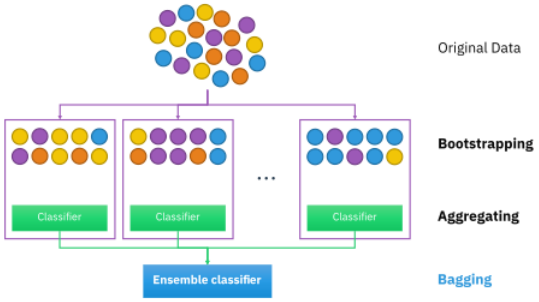


Fig. 3: RF Classifier analysis.

Boosting– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XG BOOST.

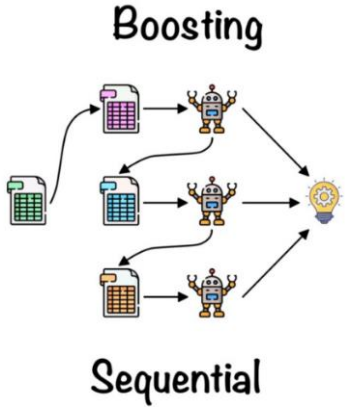


Fig. 4: Boosting RF Classifier.

3.3 Random Forest Regressor

A Random Forest regression is a powerful ensemble learning technique used in machine learning for both regression and classification tasks. It is based on the concept of decision trees and combines the predictions of multiple decision trees to improve the overall accuracy and robustness of the model.

Data Preparation: The process begins with a dataset that contains input features (independent variables) and corresponding target values (the variable you want to predict).

Bootstrapping (Random Sampling): Random Forest creates multiple subsets of the original dataset through a process called bootstrapping. This means that for each tree in the forest, a random sample of the data is taken with replacement. Some data points may appear multiple times in a subset, while others may be omitted.

Tree Building: For each subset of data, a decision tree is constructed. The construction of each tree involves selecting the best feature to split the data at each node. This selection is typically based on criteria like Gini impurity or mean squared error reduction.

Random Feature Selection: An essential aspect of Random Forest is that it introduces randomness during the tree-building process. Instead of considering all features at each node, it randomly selects a subset of features to choose from. This helps to decorrelate the trees and make them more diverse.

Tree Growing: The decision trees are allowed to grow until a stopping criterion is met. This may involve specifying a maximum depth for the tree or setting a minimum number of samples required to split a node.

Ensemble Aggregation: Once all the trees are constructed, they are used to make predictions on new data points. For regression tasks, the predictions of individual trees are averaged (or sometimes weighted) to obtain the final ensemble prediction. In the case of classification tasks, a majority vote is typically used.

Random Forest Prediction: The final prediction from the Random Forest ensemble is the average (or weighted average) of the predictions made by individual trees. This prediction tends to be more accurate and less prone to overfitting compared to a single decision tree.

Evaluation: The performance of the Random Forest regression model is evaluated using appropriate metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or R-squared (R2) on a separate validation or test dataset.

3.4 Advantages

The outlined methodology for predictive analytics in smart irrigation systems offers several notable advantages, making it a powerful approach for optimizing water management in agriculture and landscaping:

- Data-Driven Decision Making: By utilizing data collected from Node MCU devices and employing machine learning models, this methodology enables data-driven decision-making in water management. This means that irrigation decisions are based on real-time and historical data, leading to more informed and precise actions.
- Precision Irrigation: One of the primary advantages is the ability to achieve precision in irrigation. Machine learning models, such as Random Forest Classifier and Regression, can determine exactly when and how much water is needed for different crops and conditions. This precision minimizes over-irrigation, reducing water wastage and associated costs.
- Resource Efficiency: The methodology optimizes the allocation of resources, including water, energy, and labor. By accurately predicting water requirements and automating the irrigation

process, it maximizes resource efficiency, leading to reduced operational costs and improved sustainability.

4. RESULTS

Figure 5 represents the initial dataset employed in the context of smart irrigation. This dataset serves as the fundamental source of information for all subsequent analyses and modeling. It contains various data points, each capturing different aspects of the smart irrigation system, such as soil moisture levels, weather conditions, crop types, and potentially other pertinent variables. This raw dataset reflects the real-world measurements collected from sensors, such as those embedded in Node MCU devices deployed throughout the irrigation setup. These data points represent unprocessed observations and may contain missing values, outliers, or inconsistencies. Before meaningful analysis can occur, this raw data must undergo a crucial data preprocessing step, which involves cleaning the data to rectify errors, handling missing information, and ensuring that the dataset is structured for machine learning or statistical analysis.

	crop	moisture	temp	pump	water_liters	time	days
0	cotton	638	16	1	6380	5.0	2
1	cotton	522	18	1	5220	1.0	2
2	cotton	741	22	1	7410	1.8	3
3	cotton	798	32	1	7980	2.5	2
4	cotton	690	28	1	6900	3.2	3
...	...	...	...	...	...	...	...
172	cotton	853	29	0	0	0.0	0
173	cotton	922	23	0	0	0.0	0
174	cotton	998	28	0	0	0.0	0
175	cotton	966	16	0	0	0.0	0
176	cotton	950	13	0	0	0.0	0

177 rows × 7 columns

Figure 5: Sample dataset used for Smart irrigation

Figure 6 presents the same dataset as Figure 5 but after undergoing data preprocessing. This processed dataset has been cleaned and transformed to ensure that it is suitable for analysis. Preprocessing steps include handling missing values, removing duplicates, encoding categorical variables, and addressing any inconsistencies. Figure 7 displays a subset of columns from the dataset, specifically focusing on the features used for classification tasks. These features are essential variables that serve as input to machine learning models designed to predict whether the irrigation pump should be turned on or off based on the provided conditions. Examples of features could include weather conditions, soil moisture levels, and other relevant environmental factors. These columns represent the predictive variables that help the model make informed decisions regarding the operation of the irrigation system.

	crop	moisture	temp	pump	water_liters	time	days
0	0	638	16	1	6380	5.0	2
1	0	522	18	1	5220	1.0	2
2	0	741	22	1	7410	1.8	3
3	0	798	32	1	7980	2.5	2
4	0	690	28	1	6900	3.2	3
...	...	...	...	...	...	...	...
172	0	853	29	0	0	0.0	0
173	0	922	23	0	0	0.0	0
174	0	998	28	0	0	0.0	0
175	0	966	16	0	0	0.0	0
176	0	950	13	0	0	0.0	0

177 rows × 7 columns

Figure 6: Sample dataset used for Smart irrigation after preprocessing.

Figure 8 represents the target column within the dataset. This column typically contains the variable of primary interest, which, in this case, is likely binary in nature, indicating whether the irrigation pump should be turned on (coded as 1) or off (coded as 0) based on the given conditions. The target column serves as the ground truth for training and evaluating classification models. It is the variable that the machine learning algorithm aims to predict accurately, and it forms the basis for assessing the model's performance in determining when to activate or deactivate the irrigation pump. Figure 9 illustrates the output or predictions generated by a regression model designed to estimate the number of liters of water needed for irrigation. These predictions are based on input features from the dataset, such as soil moisture levels, weather conditions, and other relevant factors.

	crop	moisture	temp
0	0	638	16
1	0	522	18
2	0	741	22
3	0	798	32
4	0	690	28
...	...	...	...
172	0	853	29
173	0	922	23
174	0	998	28
175	0	966	16
176	0	950	13

177 rows × 3 columns

Figure 7: Features columns of a dataset used for classification.

```
0      1
1      1
2      1
3      1
4      1
      ..
172    0
173    0
174    0
175    0
176    0
Name: pump, Length: 177, dtype: int64
```

Figure 8: target column of the dataset

Figure 10 presents performance metrics used to evaluate the accuracy of the regression model's predictions for the number of liters of water required. Common regression metrics, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R2), may be included to assess the quality of the model's predictions and its ability to estimate water requirements effectively.

```
array([7892.67215566, 7467.14272287,    0.        , 6630.11696143,
       7679.67345029, 7250.09517416,    0.        ,    0.        ,
          0.        ,    0.        , 7368.86576331,    0.        ,
       6682.32558938,    0.        , 7174.91313335,    0.        ,
          0.        , 7623.61410117, 7243.68082351,    0.        ,
       6035.62907877, 7386.51590219, 5318.89821636,    0.        ,
          0.        ,    0.        ,    0.        , 6682.32558938,
       8269.72605516, 6301.82351378, 6810.90813808,    0.        ,
          0.        , 5370.47394827, 5321.86964493, 7010.68567138])
```

Figure 9: Prediction results for number of Liters to be watered.

```
MAE: 41.20710363623162
MSE: 7463.521712095189
RMSE: 7463.521712095189
R-squared: 0.9993808656844166
MAPE: 0.00655841639669914
```

Figure 10: Performances metrics for predicting number of Liters to be watered.

Figure 11 displays the predictions generated by a regression model regarding the amount of time required to complete the irrigation process. These predictions are based on input features such as soil moisture levels, weather conditions, and other relevant factors. This figure provides estimates of the duration needed for the irrigation system to supply water effectively. Figure 12 probably includes performance metrics designed to assess the accuracy of the regression model's predictions regarding the time required for irrigation. These metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), or other relevant measures to evaluate how well the model estimates the irrigation duration. Figure 13 represents predictions for the number of days before the land needs to be watered again following irrigation. These predictions take into account factors such as soil moisture retention and crop type to estimate the optimal timing for subsequent irrigation. This information is vital for maintaining soil health and crop growth. Figure 14 presents performance metrics that evaluate the accuracy of the model in predicting the optimal timing for the next irrigation cycle. These metrics provide insights into how well the model optimizes the irrigation schedule, contributing to efficient water management. Figure 15 probably displays predictions generated by a classification model that

determines whether the irrigation pump should be turned on or off based on the input features. These predictions are binary, indicating the recommended action of either activating the pump (1) or deactivating it (0) under specific conditions. Figure 16 comprises a detailed classification report for the Random Forest classifier model. This report typically includes metrics such as precision, recall, F1-score, and support, providing a comprehensive assessment of the model's performance in classifying when to activate or deactivate the irrigation pump. Figure 18 serves as a summary or visualization of the overall predictions and outcomes generated by the smart irrigation system. It integrates both the classification (pump on/off) and regression (water volume, time, and irrigation schedule) results, providing a holistic view of the system's recommendations for efficient water management in the context of smart irrigation.

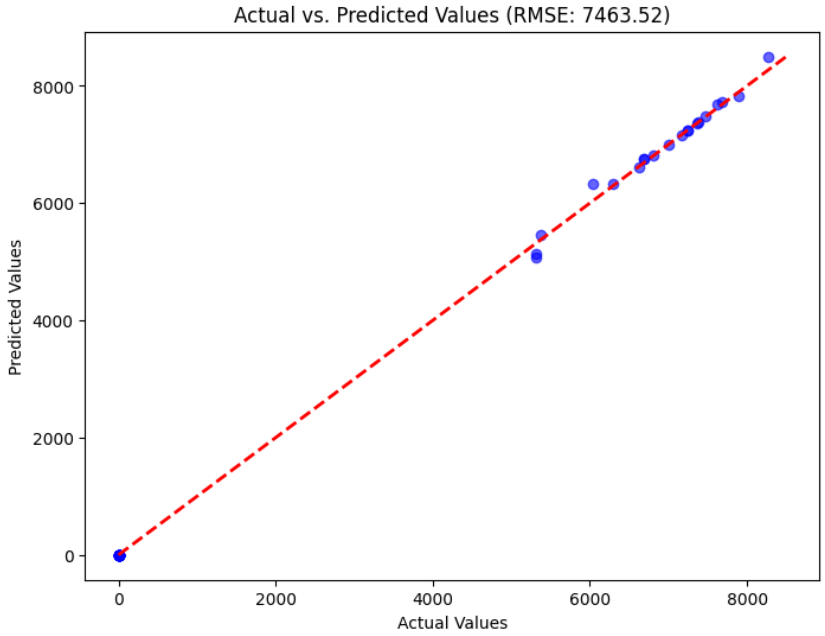


Figure 10: plot for representation of performance metrics

```
array([4.17209082e+00, 4.47396762e+00, 1.00694444e-02, 4.83294580e+00,
       4.56805642e+00, 4.37526591e+00, 5.41941392e-03, 3.05172197e-02,
       5.41941392e-03, 0.00000000e+00, 4.45580500e+00, 3.59366336e-02,
       5.28176349e+00, 1.75204248e-02, 5.36459899e+00, 0.00000000e+00,
       2.37090695e-02, 4.01487762e+00, 4.89021821e+00, 2.88533003e-02,
       4.57369709e+00, 4.13878292e+00, 4.35573882e+00, 3.59366336e-02,
       3.59366336e-02, 5.41941392e-03, 5.41941392e-03, 5.44620210e+00,
       4.58279162e+00, 4.59503549e+00, 5.49011539e+00, 5.41941392e-03,
       0.00000000e+00, 5.21529288e+00, 3.84540763e+00, 4.48803114e+00])
```

Figure 11: Prediction results for time to water the land

```
MAE: 0.7955464403198949
MSE: 1.6974048941647812
RMSE: 1.6974048941647812
R-squared: 0.6883515291065416
MAPE: 0.5334893743764575
```

Figure 12: Performance evaluation for time to water the land

```
array([4.02097630e+00, 4.83547510e+00, 1.00694444e-02, 5.31984498e+00,
       4.17190177e+00, 4.89021821e+00, 5.41941392e-03, 3.05172197e-02,
       5.41941392e-03, 0.00000000e+00, 5.15957965e+00, 3.59366336e-02,
       5.28176349e+00, 1.75204248e-02, 4.74017798e+00, 0.00000000e+00,
       2.37090695e-02, 4.01487762e+00, 4.89021821e+00, 2.88533003e-02,
       3.12414713e+00, 4.77713789e+00, 4.35573882e+00, 3.59366336e-02,
       3.59366336e-02, 5.41941392e-03, 5.41941392e-03, 5.44620210e+00,
       4.58279162e+00, 4.01784067e+00, 5.49011539e+00, 5.41941392e-03,
       0.00000000e+00, 5.21529288e+00, 4.20122361e+00, 5.32081197e+00])
```

Figure 13: Prediction results for days to water the land after the land is watered

```
MAE: 0.8292671546884979
MSE: 1.730691859356397
RMSE: 1.730691859356397
R-squared: 0.6913804943110105
MAPE: 0.5320610908196342
```

Figure 14: Performance metrics for days to water the land after the land is watered

```
array([1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1,
       1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1], dtype=int64)
```

Figure 15: Prediction results for classification of pump on or off

RandomForestClassifier classification_report:

	precision	recall	f1-score	support
0	1.00	0.94	0.97	17
1	0.95	1.00	0.97	19
accuracy			0.97	36
macro avg	0.97	0.97	0.97	36
weighted avg	0.97	0.97	0.97	36

Figure 16: Classification report for Random Forest classifier

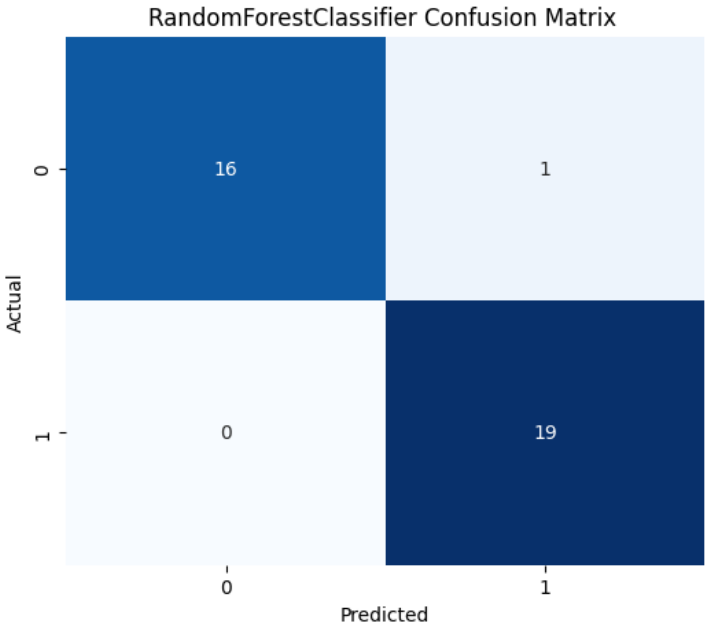
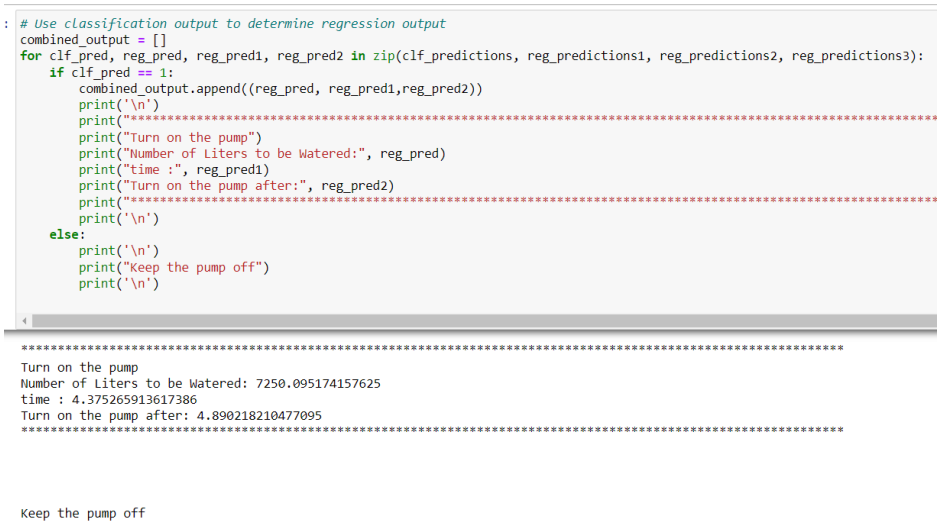


Figure 17: Confusion matrix heatmap for Random Forest classifier



```
# Use classification output to determine regression output
combined_output = []
for clf_pred, reg_pred, reg_pred1, reg_pred2 in zip(clf_predictions, reg_predictions1, reg_predictions2, reg_predictions3):
    if clf_pred == 1:
        combined_output.append((reg_pred, reg_pred1,reg_pred2))
        print('\n')
        print("*****************************************************************************************************************")
        print("Turn on the pump")
        print("Number of Liters to be Watered:", reg_pred)
        print("time :", reg_pred1)
        print("Turn on the pump after:", reg_pred2)
        print("*****************************************************************************************************************")
        print('\n')
    else:
        print('\n')
        print("Keep the pump off")
        print('\n')
```

```
*****************************************************************************************************************
Turn on the pump
Number of Liters to be Watered: 7250.095174157625
time : 4.375265913617386
Turn on the pump after: 4.890218210477095
*****************************************************************************************************************



Keep the pump off
```

Figure 18: Prediction results for Smart irrigation.

Table 2 compares the overall performance comparison of various ML models. Accuracy is a measure of how well a model correctly predicts both the positive and negative classes. In this table, the accuracy percentages indicate the overall correctness of the models' predictions. For example, the Naive Bayes Classifier achieves an accuracy of 72%, while the RFC (Random Forest Classifier) achieves an accuracy of 97%. This suggests that the RFC model performs significantly better in terms of overall accuracy.

Precision measures the proportion of true positive predictions among all the positive predictions made by the model. It indicates how well the model avoids false positives. For the Naive Bayes Classifier, the precision for the positive class is 83%, while for the RFC classifier, it is 97%. The RFC model demonstrates higher precision, meaning it has a lower rate of false positive predictions. Recall, also known as sensitivity, measures the proportion of true positive predictions among all actual positive instances. It indicates how well the model captures positive cases. In this table, the Naive Bayes Classifier has a recall of 72%, while the RFC classifier has a recall of 97%. The RFC model excels in capturing positive instances, resulting in a higher recall.

The F1-score is the harmonic mean of precision and recall. It provides a balanced measure of a model's performance, considering both false positives and false negatives. For the Naive Bayes Classifier, the F1-score is 85%, whereas for the RFC classifier, it is also 97%. The RFC model demonstrates a better balance between precision and recall, leading to a higher F1-score.

Table 2: Overall performance comparison of proposed ML models.

Model name	Accuracy (%)	Precision (%)	Recall (%)	F1-score
Naive bayes Classifier	72	83	72	85
RFC classifier	97	97	97	97

Table 3 presents a detailed comparison of the class-wise performance metrics for two machine learning models: the Naive Bayes Classifier and the RFC. The models are evaluated based on their ability to classify instances into two classes: "Pump OFF" and "Pump ON."

- **Pump OFF**: This row represents performance metrics for the "Pump OFF" class.
    - **Precision**: Precision for "Pump OFF" measures how accurately the model predicts instances when the pump should be turned off. For the Naive Bayes Classifier, the precision is 0.73, indicating that 73% of the predicted "Pump OFF" instances were correct. In contrast, the RFC classifier achieves a perfect precision of 100% for this class, meaning it correctly identifies all instances of "Pump OFF."
    - **Recall**: Recall (or sensitivity) for "Pump OFF" measures the model's ability to capture all actual instances when the pump should be turned off. The Naive Bayes Classifier has a recall of 0.86, signifying that it captures 86% of the actual "Pump OFF" instances. The RFC classifier has a recall of 94%, indicating that it captures 94% of the "Pump OFF" instances.
    - **F1-score**: The F1-score for "Pump OFF" is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. The Naive Bayes Classifier achieves an F1-score of 0.85 for "Pump OFF," while the RFC classifier attains an F1-score of 0.97. The RFC model demonstrates a more balanced performance in terms of precision and recall for this class.

Pump ON: This row represents performance metrics for the "Pump ON" class.

- **Precision**: Precision for "Pump ON" measures how accurately the model predicts instances when the pump should be turned on. The Naive Bayes Classifier achieves a precision of 0.88, indicating that 88% of the predicted "Pump ON" instances are correct. The RFC classifier achieves a precision of 95% for this class.
- **Recall**: Recall for "Pump ON" assesses the model's ability to capture all actual instances when the pump should be turned on. The Naive Bayes Classifier has a recall of 0.97, signifying that it captures 97% of the actual "Pump ON" instances. The RFC classifier achieves a perfect recall of 100%, indicating that it captures all "Pump ON" instances.
- **F1-score**: The F1-score for "Pump ON" is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. The Naive Bayes Classifier achieves an F1-score of 0.88 for "Pump ON," while the RFC classifier attains an F1-score of 0.97. The RFC model demonstrates a more balanced performance for this class as well.

Table 3: Class-wise performance comparison of proposed ML models.

Model name	Naive bayes Classifier		RFC classifier	
	Pump OFF	Pum ON	Pump OFF	Pum ON
Precision	0.73	0.88	100	95
Recall	0.86	0.97	94	100
F1-score	0.85	0.88	97	97

5. CONCLUSION

Efficient water management is essential for the sustainability and productivity of agriculture, especially in a country like India, where water resources are under significant stress and agriculture is a primary livelihood for millions. Traditional irrigation methods, while widespread, often lead to inefficient water use and are not well-equipped to adapt to dynamic environmental conditions. The

integration of smart irrigation systems, enhanced by predictive analytics and machine learning, offers a promising solution to these challenges. These advanced technologies enable the collection and analysis of real-time and historical data, allowing for precise and adaptive irrigation scheduling. This leads to optimized water use, reduced wastage, and improved crop yields. Node-MCU-based systems can gather crucial data on environmental factors such as temperature and humidity, which machine learning models can then use to forecast future water needs accurately. By continuously learning and adapting to new data, these systems can provide more accurate and timely irrigation recommendations, ensuring that crops receive the right amount of water at the right time. The adoption of such technologies in India has the potential to revolutionize agricultural practices, making them more sustainable and resilient. However, achieving widespread adoption requires addressing challenges such as high initial costs, lack of awareness, and the need for technical training among farmers. In conclusion, leveraging predictive analytics and machine learning in smart irrigation systems can significantly enhance water management in Indian agriculture. This approach not only conserves vital water resources but also supports higher agricultural productivity and food security, contributing to the overall economic and environmental well-being of the country.

REFERENCES

[1]. Koncagül, E.; Tran, M.; Connor, R. The United Nations World Water Development Report 2021: Valuing Water; Facts and Figures. Technical Report, UNESCO. 2021. Available online: https://www.unesco.org/reports/wwdr/2021/en/download-report (accessed on 3 May 2023).
[2]. Omran, H.A.; Mahmood, M.S.; Kadhem, A.A. A study on current water consumption and its distribution in Bahr An-Najaf in Iraq. Int. J. Innov. Sci. Eng. Technol. 2014, 1, 538–543.
[3]. Grafton, R.Q.; Williams, J.; Perry, C.J.; Molle, F.; Ringler, C.; Steduto, P.; Udall, B.; Wheeler, S.A.; Wang, Y.; Garrick, D.; et al. The paradox of irrigation efficiency. Science 2018, 361, 748–750.
[4]. Munir, M.S.; Bajwa, I.S.; Naeem, M.A.; Ramzan, B. Design and Implementation of an IoT System for Smart Energy Consumption and Smart Irrigation in Tunnel Farming. Energies 2018, 11, 3427.
[5]. Hunter, M.C.; Smith, R.G.; Schipanski, M.E.; Atwood, L.W.; Mortensen, D.A. Agriculture in 2050: Recalibrating Targets for Sustainable Intensification. BioScience 2017, 67, 386–391.
[6]. El-Fakharany, Z.M.; Salem, M.G. Mitigating climate change impacts on irrigation water shortage using brackish groundwater and solar energy. Energy Rep. 2021, 7, 608–621.
[7]. Pluchinotta, I.; Pagano, A.; Giordano, R.; Tsoukiàs, A. A system dynamics model for supporting decision-makers in irrigation water management. J. Environ. Manag. 2018, 223, 815–824.
[8]. Sharma, A.; Jain, A.; Gupta, P.; Chowdary, V. Machine Learning Applications for Precision Agriculture: A Comprehensive Review. IEEE Access 2021, 9, 4843–4873.
[9]. Zhai, Z.; Martínez, J.F.; Beltran, V.; Martínez, N.L. Decision support systems for agriculture 4.0: Survey and challenges. Comput. Electron. Agric. 2020, 170, 105256.
[10]. Bu, F.; Wang, X. A smart agriculture IoT system based on deep reinforcement learning. Future Gener. Comput. Syst. 2019, 99, 500–507.
[11]. Gutierrez, J.; Villa-Medina, J.F.; Nieto-Garibay, A.; Porta-Gandara, M.A. Automated Irrigation System Using a Wireless Sensor Network and GPRS Module. IEEE Trans. Instrum. Meas. 2014, 63, 166–176.
[12]. Lozoya, C.; Mendoza, C.; Aguilar, A.; Román, A.; Castelló, R. Sensor-Based Model Driven Control Strategy for Precision Irrigation. J. Sens. 2016, 2016, 9784071.