Autonomous Crop Irrigation System using Artificial Intelligence

Savita Choudhary, Vipul Gaurav, Abhijeet Singh, Susmit Agarwal

Abstract: Agriculture plays a significant role in the economy and its contribution is based on measurable crop yield which is highly dependent upon irrigation. In a country like India, where agriculture is largely based on the unorganized sector, irrigation techniques and patterns followed are inefficient and often lead to unnecessary wastage of water. This calls for the need of a system which can provide an efficient and deployable solution. In this paper, we provide an Automatic Irrigation System based on Artificial Intelligence and Internet of Things, which can autonomously irrigate fields using soil moisture data. The system is based on prediction algorithms which make use of historic weather data to identify and predict rainfall patterns and climate changes; thereby creating an intelligent system which irrigates the crop fields selectively only when required as per the weather and real-time soil moisture conditions. The system has been tested in a controlled environment with an 80 percent accuracy, thus providing an efficient solution to the problem.

Index Terms: artificial intelligence, irrigation, internet of things, prediction algorithms, machine learning, and water conservation

I. INTRODUCTION

India follows traditional agricultural methods in irrigation practices [1]. Irrigation is a significant factor in determining the crop yield and largely varies with the geographical, climatic, and topological factors. Farmers primarily depend on personal monitoring and their experience in irrigating the fields, and as a result, irrigation becomes largely inefficient and irregular. India, therefore requires a simple irrigation solution on which the farmers can depend indefinitely, which can adapt to the local climatic conditions, and accurately predict the quantity of water required by the crops in real time to ensure judicious use of water resources, and also a better crop yield. The main concern in India is not water shortage but water wastage, and poor utilization of the resources due to lack of awareness, facilities, and infrastructure. Due to wastage of water, the country has already suffered through immense drought conditions, varied rainfall patterns, and huge economic losses due to the destruction of crops [2].

Traditional automatic irrigation systems are not suitable for India, as they cannot adapt to the changing rainfall patterns and do not respond well to geographical changes. Thus, we have developed an intelligent system which can study the patterns of rainfall in a region, and predict weather conditions in order to adapt to the geography, thus predict the quantity of water needed for irrigation, to minimize wastage and increase the crop yield [3]. To achieve this, we make use of Node MCUs, and soil moisture sensors, placed inside waterproof boxes and spread evenly throughout the area to be irrigated. All these nodes are connected to a Raspberry Pi 3B+ via wireless LAN [4]. The system analyses soil moisture content through the deployed sensors, which is used to predict the quantity of water required to irrigate the area with an appreciable accuracy. It is a highly autonomous system which requires little to no human intervention once deployed in a field. The system developed makes use of Random Forest Regressor to predict the weather [5]. It makes use of the traditional data of rainfall patterns and weather data. It has the ability to slowly adapt to the region-specific climatic conditions and its accuracy improves with every prediction made. The system is designed to be updated every 30 minutes as a small regular interval which makes it power efficient as well. After every such regular interval, it updates the dataset as per the new sensor data provided. The system switches on the motor or pump if the soil moisture content detected is insufficient and based on the sensor data we make use of Partial Least Square Regression (PLSR) algorithm to predict the quantity of water required [6]. With this, we are able to calculate the time interval for which the motor should pump water. Also, in case the system detects sufficient soil moisture or rain is predicted then the pump is not switched on while the sensor measures the soil moisture capacity after the precipitation.

The system designed is power efficient, water efficient and low on maintenance. The systems are scattered throughout the area of the farm. Thus, we can switch on the drip or sprinkler for a particular area rather the entire farm in order to increase efficiency [7]. This helps in minimization of water wastage, a better understanding of crop water capacity and patterns required for efficient irrigation. Also, the nodes work on a response based system so it makes identification of any malfunction easier. The health of the nodes can be monitored through a mobile app based on the mapping of the farm and area specified for the irrigation. Thus, the system promotes low maintenance and proves to be effective.

2. METHODOLOGY

The system designed consists briefly of two major components:

Manuscript published on 30 May 2019.
***** Correspondence Author (s)

Savita Choudhary, Department of Computer Science, Sir M. Visvesvaraya Institute of Technology, Bengaluru, India.
Vipul Gaurav, Department of Computer Science, Sir M. Visvesvaraya Institute of Technology, Bengaluru, India.
Abhijeet Singh, Department of Computer Science, Sir M. Visvesvaraya Institute of Technology, Bengaluru, India.**Susmit Agarwal**, Department of Computer Science, Sir M. Visvesvaraya Institute of Technology, Bengaluru, India

DOI:10.35940/ijeat.E1010.0585S19

1. Machine Learning to predict the amount of rainfall and the soil moisture content.
2. Economic Hardware Implementation using Internet of Things.

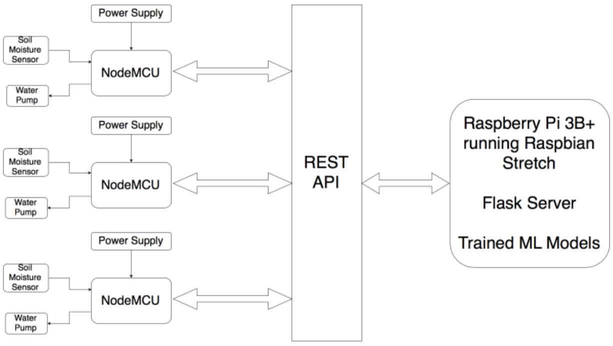

Figure 1: Architecture of the deployed system

1. *Central Controller*

A central system is used to control the nodes, and to decide the time for which the pump attached to any particular node is switched on. This implementation uses a Raspberry Pi 3B+, running the Debian Linux-based Raspbian Operating system as the central controller [4].

The controller uses a Python3 script to execute all its functions. Support scripts are imported to the main script, and the control logic calls methods from the support scripts as and when required. The functions include communication with the nodes, data preprocessing and deriving inferences using Machine Learning algorithms. Some of these functions are implemented as Webhooks using Flask Framework.

The machine learning models are not trained on the Raspberry Pi. Instead, training is performed on a high-end computer, and the trained models are then deployed in the production environment.

2. *Distributed Nodes*

Each node is comprised of an ESP8266 [8] based microcontroller board (NodeMCU) and an array of analog sensors. These sensors are used to fetch information about weather conditions, which is converted to digital 10-bit integers using the Analog to Digital converters present on each NodeMCU. An algorithm converts this data to a 32-bit floating point value required by the inference model, on the NodeMCU itself.

Each node contains additional support hardware to provide required electrical power to the NodeMCU and the sensors and to assist easy debugging in the event of failure of the Node.

The power circuit consists of a 9V battery input given to a 3.3V regulator. The 3.3V output is then used to power the NodeMCU and the sensors [13]. A common Vcc line and GND line are used to supply power to the aforementioned components. Additionally, the 9V supply is directly provided to an L293D motor drive IC, which powers the water pump based on the inputs it receives from the NodeMCU.

The implementation may also include one or more solar panel(s) to power the NodeMCU and the sensors, using battery power only when the pump is required.

The preprocessed data, along with some additional bytes used to identify each individual node, is sent to the central controller via the HTTP GET request.

3. *The Network*

The nodes are connected to the central controller through a shared WiFi access point, over the IPv4 protocol [9]. A client-server architecture is used for communication between the nodes and the central controller. The nodes do not communicate with each other.

This implementation uses a Flask (Python) server on the central controller. The server extends a URL in the form of a webhook. The nodes connect to the server via this webhook to send data. The HTTP GET request is used to transmit the sensor data after post-calculation to the server [14]. The default response of the server conveys two messages to the node that sent the request:

1. If the pump attached to the particular node has to be switched on or not, and
2. The duration for which the pump has to be switched on.

The duration is calculated using the output of the regression algorithm. Each node sends a request to the server at an interval of 30 minutes.

The network is thus essentially a wireless Local Area Network. Since each node works independent of other nodes, incidental failure of a node does not result in the collapse of the entire system. Additionally, isolation of the failed node becomes easy.

The IPv4 Address of the central controller is hard-coded into each NodeMCU. The central controller is assigned a static IPv4 address using the access point settings. The port used for communication is set to 8080 for this implementation.

4. *Rainfall Prediction using Machine Learning*

The rainfall prediction involved a two-phase solution:

- Prediction of Probability of Rainfall in the next 30 minutes
- Estimation of the Amount of Rainfall

The first phase is to help the network to realize whether there is a chance of rainfall to occur in the next 30 minutes or not and the device keeps checking the status at regular intervals of time when it is switched on periodically. This is achieved by predicting the probability of rainfall to occur in the next 30 minutes.

The system is made power efficient by not constantly keeping it on for checking the status of rainfall rather checks at regular time intervals itself. The amount of rainfall depends on multiple parameters including mean of temperature, pressure, maximum and minimum humidity as well as the mean dew associated with the air [15]. The dataset used is accountable for local areas and regions since it contained parameters which can be generalized to predict the rainfall and had random values of all the parameters.

The data used is taken from the rainfall data available on the Government of India Portal for local regions. The initial deployment of the model took account of the data provided for the state of Punjab primarily for wheat crop. The data was divided into three sets, namely, training, validation and testing, with a percentage of 70, 15 and 15 respectively. The values of the dataset were easily accountable with Random Forest Classification and Regression. The rainfall estimation involves many parameters highly dynamic and prone to changes in real time and therefore it made an artificial neural network [16] and support vector machines inefficient comparatively [17].

The idea is to design a deployable model and therefore random forest regression proved to be more adaptive to dynamic real time data and gives nearly accurate results for weather prediction needed to determine the whether it would rain or not [5].

Once it has been determined that the rainfall would occur or not the next challenge is to predict the amount of rainfall which would occur. This majorly deals with the historic precipitation data available for different regions and the model has to adapt to a huge amount of such historic data spread over a wide time period. For this, we took the historic data available for rainfall for different states provided by the Government of India for research purposes. The rainfall data involved non-linear patterns and has varied intensities associated with it.

The system is designed to adapt to any state and also updates the dataset with the new rainfall values which is being recorded with the help of the Raspberry Pi. The overall implementation of the system makes it highly dynamic and more accurate as well as adaptable to any region with onset of time of deployment.

The rainfall estimation in India involved monthly and yearly patterns which were studied through use of Autoregressive Integrated Moving Average Time Series popularly known as the ARIMA Model used for rainfall estimation from historic data [10]. The results were found to be as expected with variations in intensity of rainfall and non-linear patterns observed uniquely for each state.

The dataset used is described as follows:

Serial	Description	Observation
1	Mean Temperature (monthly)	Celsius
2	Mean Dew (monthly)	Range of Temperature
3	Mean Pressure (monthly)	Millibar
4	Minimum Humidity	Gram per cubic metre
5	Maximum Humidity	Gram per cubic metre
6	Maximum Temperature	Celsius
7	Minimum Temperature	Celsius
8	Maximum Dewpoint	Celsius
9	Minimum Dewpoint	Celsius
10	Maximum Pressure	Millibar
11	Minimum Pressure	Millibar

Table 1: Parameters used for Rainfall Prediction

The details of the rainfall patterns were determined through the time series [18] and the probability of the rainfall was determined already combining both of which we get the desired output.

5. *Estimation of Soil Moisture Content required*

The decision system takes into account the water level content required by the crop to calculate its irrigation requirements. In traditional applications, these decisions are taken by the farmer based on their experience or an expert agricultural expert. Information from various sources is collected; these include weather statistics, crop and soil properties, moisture content data collected from soil sensors deployed in the fields, and moisture influencing factors such as evapotranspiration [11]. An agricultural expert analyzes this data to decide an estimate of the amount of water that the crop would require on a particular day.

Based on this idea, our autonomous irrigation system is developed. We collect historical data which consists of decisions taken by an agricultural expert for the water content required by a crop based on available information and current crop condition. The collected information is analyzed using machine learning to predict the water requirements for the crop in real-time.

Decision history by an agricultural expert is used to evaluate our system's performance. The decisions taken by the machine learning model are evaluated against those taken by the agricultural expert. The machine learning system is to be trained with historical data and decision reports of the agricultural expert, taking into account the known water level requirement of the particular crop and various real-time factors mentioned previously. The aim of the machine learning model is to be as close as possible to the decisions taken by an agronomist, which are used as ground truth for evaluation. We tested various machine learning models to achieve the best performance with the minimum necessary computation required. Figure 1 illustrates a schematic representation of the machine learning system.

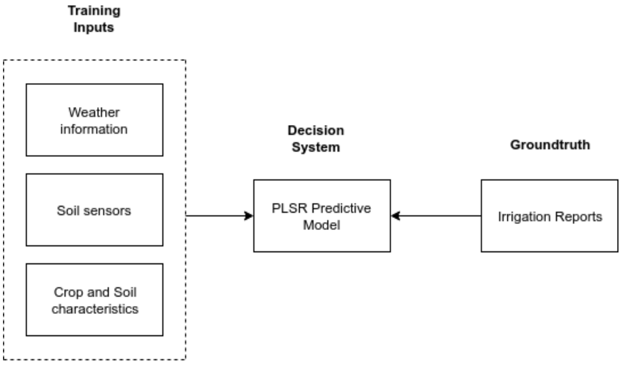

Figure 2: Design of the moisture content decision system

The machine learning based decision system, when trained with accurate data, can provide a precise estimate of the irrigation requirements of the crop in consideration, given the required real-time information including the crop and soil condition. Table 2 shows the set of possible factors to be considered as input to the decision model.

Serial	Input	Source
1	Soil Temperature	Soil sensors
2	Soil Water Potential	Soil sensors
3	Volumetric Water Content	Soil sensors
4	Rainfall	Weather Stations
5	Wind Speed	Weather Stations
6	Temperature	Weather Stations
7	Relative Humidity	Weather Stations
8	Global Radiation	Weather Stations
9	Dew Point	Weather Stations
10	Vapour Pressure Deficit	Weather Stations
11	Crop Evapotranspiration	Crop & Soil Properties

Table 2: Set of possible inputs to the decision system

Evapotranspiration is derived from the words “evaporation” and “transpiration”. It refers to the process of moisture escaping from the soil and crop to the atmosphere. Carefully analyzing the evapotranspiration rates of a crop is essential to estimating its irrigation requirements [19].

The water to be irrigated to the crop is the amount of water level (moisture content) required to compensate for the evapotranspiration moisture loss from the field. Hence, finding out the evapotranspiration of a crop field is a major deciding factor in estimating the irrigation requirements of the crop. The FAO Penman-Monteith formula can be used to calculate the reference crop evapotranspiration (ET_0) on a daily basis, using information from weather stations or alternatively weather sensors [20]. The crop evapotranspiration can be calculated under standard conditions using the single crop coefficient formula given below:

$$ET_c = K \,.\, ET_0$$

Here K_c denotes the crop coefficient. It depends on various factors such as the crop type, crop evaporation, crop growth stage and weather information.

We’ve used the Partial Least Square Regression (PLSR) [21] algorithm to design the machine learning decision model. It is a statistical algorithm that identifies fundamental relations between input and output variables. Predictor (input) variables, denoted by *X*, are defined as the observed variables that are measured for providing input to the decision model. Response variables, denoted by *Y*, are the outputs that must be estimated provided the input.

We use the PLSR technique among other regression algorithms since it efficiently tackles cases when the number of inputs is much higher than the number of output variables, the outputs are noisy and there exists a high probability of having multicollinearity among the input variables. The multicollinearity problem occurs when input variables are highly correlated, due to redundancy between meteorological factors and soil sensors data. We find out that all of these factors appear in our irrigation decision system.

The training set {*X*, *Y*} of S samples is used to train the PLSR model. It comprises of the predictor input matrix $X=[x_1, ..x_i, ..x_S]^T$ and the response output matrix $Y=[y_1, ..y_i, ..y_S]^T$. Here x_i is a vector of *N* elements which contain all the soil sensor and weather data measured at a particular day *i*. Since in this model, the output is only the time of irrigation recommended for that day, y_i is a scalar, containing the variable to be estimated at a given day *i*.

The trained PLSR model takes the predictor matrix as input which contains the soil and weather information for a particular day, and predicts the soil moisture percentage required for the crop. This soil moisture percentage helps us to calculate the minutes of irrigation required as a function of the area of the crop field and the power of the electric motor being used.

III.RESULTS AND DISCUSSION

The system designed is a smart irrigation solution based on artificial intelligence which is makes use of the soil moisture content and the moisture requirements of the crop to make the entire process of irrigation automatic. Its core benefit is its efficiency and economic feasibility.

The main idea behind the rainfall estimation proposed is to get the almost correct estimation of the rainfall of a specific local region as well as the annual rainfall to get data for future estimation of the rainfall in states. The variation in the non-linear patterns of rainfall for India over the historic data is carefully analysed and patterns observed in the historic period from 1920 to 2000 are as follows:

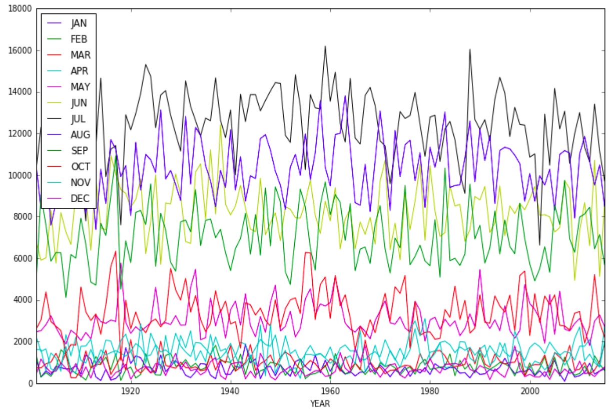

Figure 3: Time Series Plot of Indian Rainfall from 1920 to 2000

The annual rainfall predicted for the state of Punjab using Time Series ARIMA Model is 545 mm for the year 2015 which is nearly to the original value of 549.5 mm with an accuracy of 83%. This is further verified with the monthly rainfall prediction in the state.

The system predicts the probability of rain through this and thus if the probability of rain comes out to be higher than 0.5 then the motor is not switched. If the probability of rain is lesser than 0.5 then the soil moisture requirement is calculated provided the soil moisture sensor data thereby switching on the motor for only the required time period automatically.

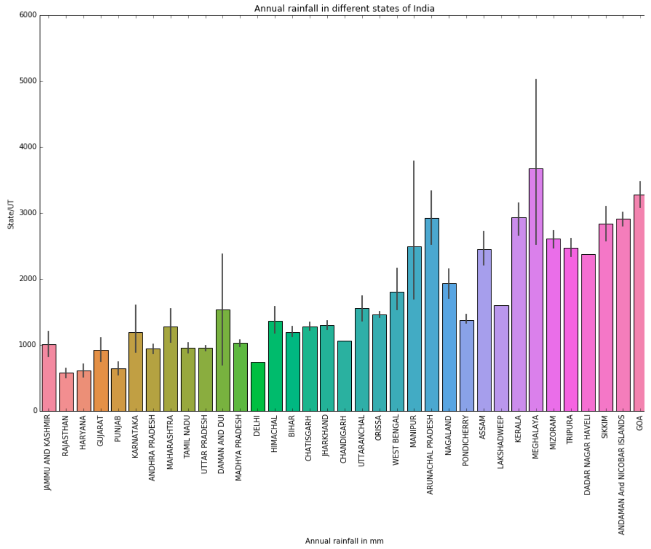

Figure 4: Annual Rainfall Prediction (in mm) for Indian States

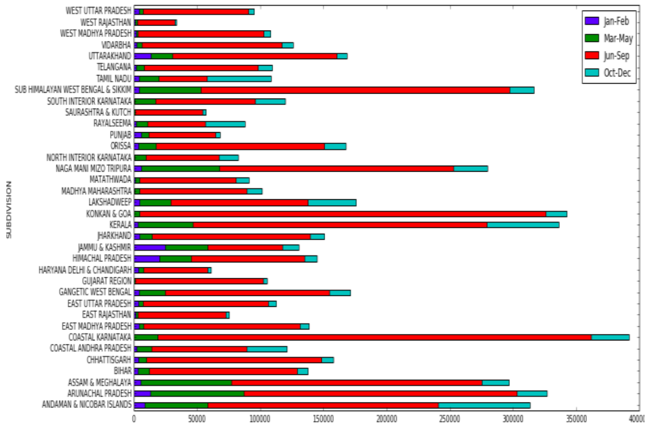

Figure 5: Monthly Rainfall Prediction (in mm) for Indian States

IV.CONCLUSION

The autonomous irrigation system we developed uses machine learning and predictive algorithms to add intelligence to existing concept of automatic irrigation systems. The methods discussed in this paper can help increase irrigation efficiency while decreasing effort required and aids water conservation compared to current irrigation methods. The system currently depends on weather station information for its calculation. This dependency can be replaced with on-premise sensors for deployment in rural areas widely found in the Indian subcontinent and arid regions where water is available in limited quantity.

REFERENCES

1. India - History of Irrigation, FAO - United Nations, **2014**
2. Anjal Prakash, "*Water in India: Situations and Prospects*", Water Policy, **2014,** pp. 425-441
3. Monteith, J.L., Steps in crop climatology. In: Challenges in Dryland Agriculture: A Global Perspective. P. Unger (ed.). pp. 273-282. Proc. of the Int. Conf. on Dryland Farming (14-18 Sept., 1998). Amarillo/Bushland Texas, **1990**
4. Bhagyashree K Chate "*Smart Irrigation System using Raspberry Pi*", International Research Journal of Engineering and Technology (IRJET), Vol 3, Issue 5, May, **2016,** p-ISSN: 2395-0072
5. S. Karthick, "*Weather Prediction Analysis using Random Forest Algorithm*", International Journal of Pure and Applied Mathematics, Vol 118, No. 20, **2018,** pp 255-262
6. Antonio P. Leone, "*Prediction of Soil Properties with PLSR and vis-NIR Spectroscopy: Application to Mediterranean Soils from Southern Italy*", Current Analytical Chemistry, **2012,** pp 283-299
7. Schneider, A.D., and T.A. Howell. 1995. "*Reducing Sprinkler Water Losses.*" In: Proc. Central Plains Irrigation Short Course. Garden City, KS. Feb. 7-8, **1995,** ISSN:0918-5623
8. K. Jyotsana Vanaja, "*IOT Based Agriculture System using Node MCU*", International Research Journal of Engineering and Technology (IRJET) , Vol:05, Issue:03, March, **2018,** p-ISSN: 2395-0072
9. Dinya Abdulahad Aziz, "*Web Server Based Smart Monitoring System Using ESP8266 NodeMCU Module*", International Journal of Scientific and Engineering Research, Vol 9, Issue 6, June, **2018,** ISSN:2229-5518

10. M. Sidiq, "*Forecasting Rainfall with Time Series Models*", IOP Conference Series Materials Science and Engineering ,September, **2018**

11. R.G. Allen, "*An Update for Calculation of Evapotranspiration Reference* ", ICID Bulletin, Vol.**43,** Issue.**2**, **1994**

12. Fan TongKe "*Smart Agriculture Based on Cloud Computing and IOT", Journal of Convergence Information Technology*, January **2013**

13. Pradeep Singh, "*3.3 V Step Down Power Supply for ESP8266*", Article by IOT Bytes
14. Mohammad Ali Jazayeri, "*Implementation and Evaluation of Four Interoperable Open Standards for the Internet of Things*", Sensors (Basel), September, **2015**
15. Shabib Aftab, "*Rainfall Prediction using Data Mining Techniques: A Systematic Literature Review*", International Journal of Advanced Computer Science and Applications, Vol 9, No. 5, **2018**
16. E.J. Hernandez, "*Rainfall Prediction: A Deep Learning Approach*", International Conference on Hybrid Artificial Intelligence Systems, **April**, **2016**
17. Fethi Ouallouche, "*Improvement of Rainfall Estimation from MSG data using Random Forests Classification and Regression* ", Atmospheric Research, **211**, May **2018**
18. M.J. Langkwist, "*A Review of Unsupervised Feature Learning and Deep Learning for Time Series Modelling*", Pattern Recognition Letters, **June, 2014**

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

19. Nouri, Hamideh; Beecham, Simon; Kazemi, Fatemeh; Hassanli, Ali Morad. "*A review of ET measurement techniques for estimating the water requirements of urban landscape vegetation*". Urban Water J. 10 (4): 247–259, **2013**
20. R.G. Allen, “*Crop Evapotranspiration-Guidelines for Computing Crop Water Requirements*”, Food and Agriculture Organization of the United Nations, **Rome**, **1998**
21. Wold, S., Ruhe, A., Wold, H., Dunn, I., W., The Collinearity Problem in Linear Regression. The Partial Least Squares (PLS) Approach to Generalized Inverses. SIAM J. Sci. Stat. Comput. 5, 735 743. doi:10.1137/0905052, **1984**