Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 165 (2019) 126–135

Procedia Computer Science

www.elsevier.com/locate/procedia

INTERNATIONAL CONFERENCE ON RECENT TRENDS IN ADVANCED COMPUTING 2019, ICRTAC 2019

Smart Plant Watering System with Cloud Analysis and Plant Health Prediction.

Arathi Reghukumar[a*], Vaidehi Vijayakumar[b]

[a] Vellore Institute of Technology,Chennai.
[b]Mother Teresa Women's University, Kodaikanal.

Abstract

One of the major global concerns of our current era is water scarcity. This makes it all the more important to design water-efficient systems for irrigating our crops and plants. The main aim of this paper is to design a watering system for status of the plant health. The existing methods do not follow a convenient approach to visualize the recorded sensor data graphically nor do they have any facility to predict and inform the farmers about the health status of the plant Our design involves analyzing the moisture content, temperature of surrounding in which the plant grows to decide whether to release water or not from the electric motor and the data of the sensors will be displayed in graphical form on an Adafruit cloud page, which is an IOT platform (hardware and software interface) and this is further used to analyze the plant health and send an email alert to the farmer or person concerned. The system thus conserves water while irrigating the plants whilst avoiding the compulsion for continuous human supervision.

Keywords: Sensors; Cloud Analysis; IOT; Smart Irrigation;

1. Introduction

Developing countries like India suffer a great deal from water scarcity. With this in mind, it is vital to come up with innovative methods of irrigation. The proposed system comes up with one such method to water

*Corresponding author. Tel.: 9003210773.
E-mail address: arathi1107@gmail.com

1877-0509 © 2019 The Authors. Published by Elsevier B.V.

10.1016/j.procs.2020.01.088

the plants while conserving water and eliminating the need for a human to be continuously supervising the system

Our objective is to:

- Analyse sensor readings and decide when to irrigate the plant.
- Predict the general health status of the plant.

For achieving our first objective, we use a number of sensors to measure soil and environment conditions and compare it with the threshold to decide if the plant requires watering based on a decision model. For achieving our second objective, we set various growth conditions that are required for optimal growth of the plant. We further use this model to predict the plant's health.

Some features of our model are:

- Automatic start and stop of electric motor based on water threshold value existing already in the plant.
- Flame detector accompanied by a buzzer to set off an alarm in case of fire.
- E-mail alert sent to the farmer regarding the current health of the plant based on certain optimal conditions that are required for plant's sustainability.

The cited paper "Sensor based Automated Irrigation System with IOT: A Technical Review"[1] comes up with a solution to minimize human intervention in irrigation by proposing a method for automatic irrigation of the crops by detecting changes in temperature and humidity of the surroundings. However, the system has no facility to view the recorded sensor data and predict health of the plant. Further, there is also no facility to alert the user in case of fire. Our paper provides the user with a convenient method to view recorded data both in an LCD display as well as on a cloud platform. Further, it also analyses the recorded sensor data and comes up with a prediction of the plant's health. The system also alerts the user in case of a fire accident using an alarm. Additionally, this system comes with a facility to alert the farmer or user in case the health of the plant is deteriorating or assure them that the health of the plant has been consistently good.

The paper is organised as follows: Section 2 is Literature Survey, whereas Section 3 elaborates about the proposed Smart plant watering system. Section 4 covers the results obtained and discussion

2. Literature Survey

In 2011, Marios C et al [2], proposed a WSN-based smart home-irrigation system, which consisted of special sensors, heterogeneous motes and actuators. The proposed system was fully adaptive to specific water needs that different plants might have. Atta R, Boutraa T, Akhkha A[3] in 2011 put forth a system to measure soil moisture value and decide the water amount to be provided for a field of wheat crops. A mobile field data acquisition system and wireless sensor networks were used to do so. Kumar A et al [4] in 2014 presented an Irrigation System that used a sensor to measure soil moisture and to control the amount of irrigation water in water deficient areas. The data acquired through the sensor is sent through XBEE wireless communication modules to the server controlling the water supply to the crops. In 2015, Gainwar S. D and Rojatkar D. V. [5] proposed a system to measure the soil parameters such as moisture, temperature and pH. The system used an LM35 sensor in addition to a pH 100 sensor and a HS-220 sensor to measure the required parameters and better inform the farmers about the soil status and how it could be improved to achieve optimum plant health.

Kansara K et al [1] in 2015 proposed a system that continuously measured the amount of water in the tank and estimated the quantity of water that would need to be pumped for watering the plant at a given time. Further

Archana P and Priya R [6] in 2016 proposed a similar Irrigation system that made use of Atmega328p microcontrollers. The system was to sense the soil moisture value and decided if the crop needed to be irrigated. The Solar powered irrigation system proposed by Balaji V R and Sudha M [7] in 2016 implemented a watering system that made use of solar panels as a means to run the motor and irrigate the plant. They used a boost converter and a hygrometer to achieve this. Gondchawar N and Kawitkar R S [8] in 2016 suggested a system which included a smart GPS based remote controlled robot to do tasks like weeding, moisture sensing, spraying, keeping vigilance, bird and animal scaring etc. The system also included an irrigation system with smart controls and intelligent decision making model based on real time field data. As a third feature, humidity maintenance, temperature maintenance and moisture maintenance was included.

In 2017, Roopaei M, Rad P and Choo K.K.R [9] put forth a system that used Thermal Imaging for the Irrigation management and used Cloud computing for managing water source-related data prior to the irrigation process. Further in 2017, Rawal S [10] suggested a system that used an Arduino processor to measure and maintain appropriate soil moisture threshold. The system also consists of a webpage that displays the current water sprinkler status. Garcia A.M et al [11] in 2018, proposed a Smart Irrigation System that used Solar Energy and coupled it with the energy needs of the Irrigation network. The system also avoided the production of a considerable amount of carbon-dioxide.

As seen above, the existing methods do not provide a user-friendly platform to visualise graphical output from sensors. Also, there is no convenient facility to predict and intimate the farmers about the health status of the plants, if they were to continue to grow in the current environmental conditions.

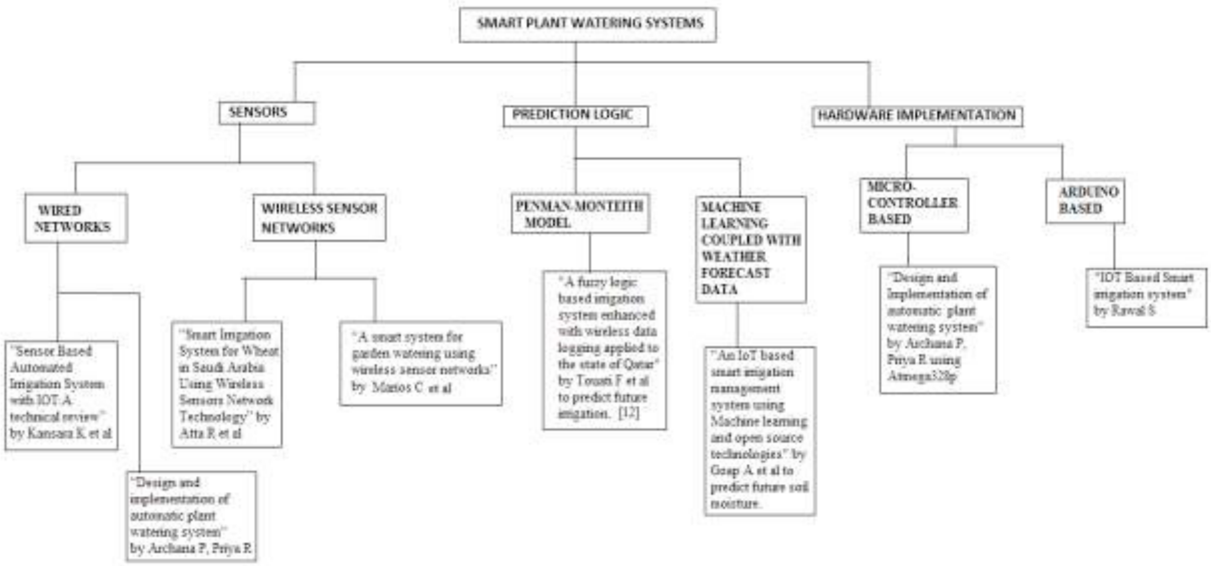

Figure1. Taxonomy diagram

Figure 1 shows the Taxonomy of existing Smart Plant-Watering Systems that have been proposed in the past years.

3. Proposed smart plant watering system

Our design records sensor values such as soil moisture, temperature, pH of the soil to decide if the plant needs to be irrigated. The recorded sensor values remain in the system until it is manually shut down by the user. All of these values could be used by the system to predict the health status of the plant. We used a decision tree model to achieve this. In addition, we visualize the data obtained from the sensors on the cloud platform and use this further to automatically evaluate the plant's health. Once this evaluation is done, the system sends an email regarding the current status of the plant health to the farmer. The system makes use of an Arduino UNO board. The flame sensor, pH sensor and DHT sensor are connected to pin 4,pin 2 and pin A0 of the compiler board respectively. The DHT sensor measures both temperature and pH for the system. The buzzer is connected to pins 5, while the WiFi module is connected to pins 0 and 1 of the Arduino board. The LCD display is connected to pins 7,8,9,10,11,12 of the Arduino, whereas, the ULN2003 IC is connected to pin 6 of the compiler. Thus is the overall architecture of the system.

3.1 Materials and Methods

This project proposes a Plant watering system which provides farmers with real time insights on their crops through a cloud platform and additionally makes use of machine learning algorithms for prediction. A low-cost system has been proposed so as to make it available to everyone. Following that, 24/7 farm monitoring is provided through a collection of sensors in order to have a better control over the soil moisture content, the possibilities of fire, pH and temperature. The data about humidity and temperature of the soil is also sent to the cloud platform-Adafruit IO, which is a free platform for cloud computing (ESP8266) through a Wi-Fi module. The data is then transformed into graphical representation to be viewed conveniently. The Arduino code is used to evaluate the sensor values with the threshold and take appropriate action. If the value is lesser or far greater than the threshold value, then water has to be either pumped or restricted respectively. All the sensor values are then displayed in our cloud platform (Adafruit IO). This is used for making prediction about the plant's health. Additionally, the cloud platform provides the user with a visual and graphical representation to conveniently analyse the sensor recordings and the health statistics of the plant such that they could figure out what could be the possible factor for any changes in the plant's health status. The prediction is also sent via an email to the farmer so that the farmer can make arrangement for the optimal conditions that the plant require for its growth.

Table 1. Sensor Threshold values

Result from decision tree	Physical value of the sensor	Decision to do
Soil moisture	If >1100	Stop pouring water
	If <1100	Pour water
Temperature of atmosphere	If > 25c and<50c	Plant healthy
	If <0c and>50c	Plant unhealthy
Humidity of atmosphere	If >50 and<80	Plant healthy
	If <50 and>80	Plant unhealthy
Flame	If input 1	Alarm
	If zero // no smoke	Do nothing

PH sensor	If pH >5.5 and pH<7.0	Plant healthy
	If pH<5.5 and pH<7.0	Plant unhealthy

Table.1 shows that soil moisture threshold is set at 1100 for optimal plant health. If it is recorded to be greater than 1100 we need not irrigate, if less we need to. This would be particularly helpful on rainy days when the soil moisture content would be higher than the threshold value set. Thus, the system prevents the plant from being over-watered, water wastage and ensures proper respiration.

3.2 Hardware

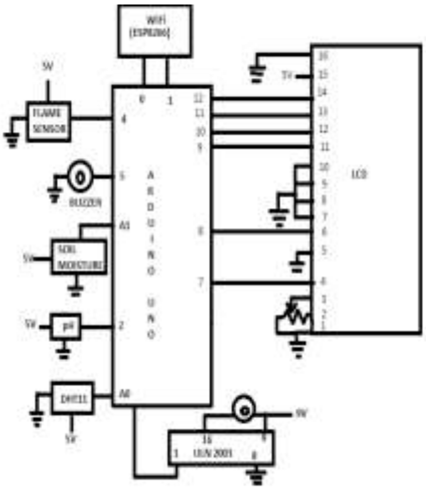

Figure 2. Architecture of the proposed plant watering system

Figure 2 shows the Arduino's connections made for our proposed system. First the sensors are connected to the Arduino in the following pins. Flame sensor is connected to pin 4 in the Arduino buzzer, which is by itself connected to pin 5 of the Arduino board. The IC (ULN2003 used for driving the motor) is connected to the 6th pin of Arduino. The DHT sensor is connected to pin A0 of the Arduino. The pH sensor is connected to pin 2 of the board. The soil moisture sensor is connected to pin A1 of the ESP8266 Wi-Fi module, which is in turn connected to pin 0 and pin 1 of the Arduino, whereas, the LCD display is connected to pins 7, 8, 9, 10, 11, 12 of the Arduino.

The code that is required for collecting sensor values and comparing with the threshold values is written in the Arduino compiler, verified and uploaded onto Arduino. Once uploaded, it stays in the Arduino until and unless we reset it. The sensor value readings that are collected can be viewed via the serial monitor too1 present in Arduino compiler. The values are also displayed in LCD connected to the system. LCD displays four values-soil moisture, humidity, temperature and pH values. We can view continuous readings of these sensors as the plant grows. The Arduino checks the threshold values and takes actions accordingly. When the moisture value recorded is lower than the set threshold, information is sent to ULN2003 motor driver IC that drives the motor pump to pour water. In the event of a flame being detected, the system directs to turn the alarm on.

The Wi-Fi module that we use here is ESP8266. This is responsible for sending the collected sensor readings to the cloud platform-Adafruit. Adafruit IO is user-friendly and free of cost cloud platform. We can view all details such as current readings and previously recorded readings. We can also visualize the data in a graphical form. We use publish-subscribe communication model using MQTT protocol to publish data onto cloud and subscribe it whenever we want. We also collect the sensor readings in a .csv file. This .csv file is helpful in processing the data.

3.3. ARDUINO UNOBOARD

Arduino Uno is a microcontroller board based on the ATmega328P datasheet. It has an operating voltage of 5V and an input voltage of 6-20V with 7-12V being the recommended limit. It has provision for 14 I/O pins, including 6 that provide PWM output. In addition, there are 6 Analog input pins. It allows a DC Current of 40 mA per I/O pin and a DC current of 50 mA for 3.3V pin.

3.4. SENSORS

3.4.1 SOIL MOISTURE SENSOR (YL38)

A Soil Moisture sensor aims to accurately measure the soil's moisture content. When the moisture value recorded is above the threshold, low level (0V) will be the output and if recorded to be below the threshold, high level (5V) will be the output. The digital pin reads the current soil moisture value to see whether it is above threshold or not. The threshold voltage can be adjusted with the help of a potentiometer.

3.4.2 pH SENSOR

pH is a measure of acidity or alkalinity of solution which can be determined by the relative number of hydrogen (H+) or hydroxyl (OH-) ions present in the solution. A pH value below 7 is considered acidic and one above 7 is considered basic. The pH can change with temperature.

3.4.3 FLAME SENSOR

It is an Infrared sensor which is used to detect light/flame by relating it to the variation of Infrared radiation. The sensor covers up to 10m at angles of ±15 degrees. PIR is similar to outdoor light with motion detector, reacting to movements that heat radiating objects make.

3.4.4 DHT SENSOR (DHT11)

The DHT11 sensor is a basic and ultra low-cost humidity and digital temperature sensor. It makes use of a capacitive humidity sensor and thermistor to measure surrounding air, and outputs a digital signal on the data pin. It is relatively simple to use, but it requires precise timing to grab data.

- Good for 20-80% readings of humidity value with 5%accuracy
- Good for 0-50°C readings of temperature value ±2°Caccuracy
- It is low cost
- It has 3 to 5V power and I/O
- It has a max current usage of 2.5mA during conversion

3.5 WI-FI MODULE (ESP8266)

The ESP8266 is a self – contained System on Chip (SOC) with integrated TCP/IP protocol stack that gives the microcontroller access to a Wi-Fi network. Each of these modules can simply be connected to the Arduino device to procure Wi-Fi ability. The module has a powerful on-boarding process and has a high storage capacity which allows it to be integrated with sensors used as well as other application specific devices that may be present. The module collects all of the recorded sensor data, later transferring it to the cloud by using Wi-Fi connectivity.

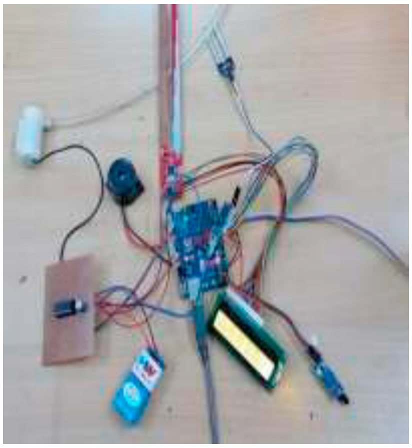

Figure 3. Hardware Implementation.

The integrated system with sensors is shown in Figure 3.

In order to study from the recorded sensor data and to predict the health status of the plant, we need to use suitable Machine Learning Algorithm. The machine learns from recorded hypothesis set and gives a resultant hypothesis based on the recorded data, thus eliminating the need for a human to physically examine all the recorded data and make the prediction.

3.6 Machine Learning

For the Machine Learning aspect of the system, we use FindS algorithm to predict if the plant is healthy or not. It is a supervised learning algorithm. The algorithm attempts to find the most specific hypothesis that is found to be consistent with the training data. The instances in the training dataset are given a class label of either "Healthy" or "Not Healthy". Hence any new instance is analysed along with the data in the training dataset to assign its associated class label. The maximally specific hypothesis is then found based on these assigned class labels.

The basic structure of the algorithm we use is given by:

1. Initialize h to first specific hypothesis in H
2. For a positive training instance x

   For each of the attribute constraints a in h

      If constraint a, is satisfied by training instance x

         It is assigned a value "pass" for the first constraint

         Check for the next constraint that x satisfies

3.Output hypothesis h

Here we mention optimal conditions that a plant requires for its growth. That is- a plant requires an ambient temperature of 25-50° c, humidity of 50-80, soil moisture within a range of 1000- 1500. These are our hypothesis set. The machine learns these hypotheses from the FindS algorithm and comes up with a final hypothesis. This final hypothesis is used for prediction. If all the hypothesis parameters are within the given range, the maximally specific hypothesis would be ["True", "True", "True", "True"] and the plant is considered healthy, if not it is considered to be unhealthy.We also use SMTP (Simple Mail Transfer Protocol) to send the predicted result (whether the plant is healthy or not) via a Gmail account to the farmer.

3.7 Prediction

Only if each of the hypothesis parameters is assigned a value of “True”, is that hypothesis set considered a healthy hypothesis one. Experimental averaging is used to come up with the maximally specific hypothesis from the list of hypothesis sets. For example, if for a set of parameters, the pH value is recorded as 6, the temperature is measured as 30c and soil moisture value is measured as 1020 with humidity at 70%, the hypothesis set would be [“True”, ”True”, ”True”, ”True”]. If for another set, the pH value is measured at 13, the temperature is 29c, soil moisture is recorded as 1015 and humidity given by 40%., the hypothesis values would be [“False”, “True”, “True”, “False”]. If there exists a third set with parameter values for pH, temperature, soil moisture and humidity as 12, 31c, 1022 and 54%, the hypothesis set would be [“False”, “True”, “True”, “True”]. Finally, by averaging, the maximally specific hypothesis would be [“False”, “True”, “True”, “True”] and the plant is considered unhealthy.

4. Results and discussion

The soil and environment parameter values were measured for a basil plant and the recorded sensor values was visualised on the Adafruit IO cloud platform along with graphical visualisation. The hypothesis sets obtained using findS algorithm on the recorded sensor data is displayed as output on the serial monitor. The final maximally specific hypothesis set obtained on averaging is also displayed as output on the monitor. An e-Mail stating the health status of the plant was also received.

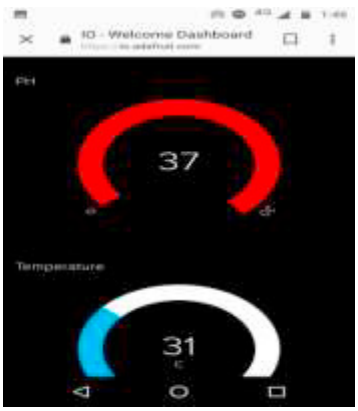

Figure 4. pH and temperature displayed on Cloud page

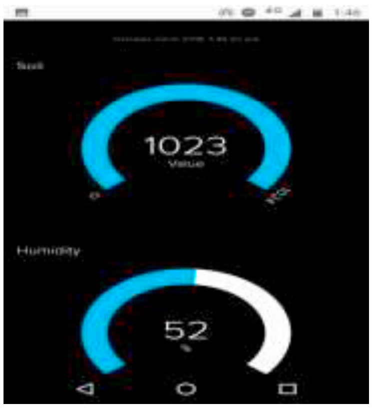

Figure 5. Soil Moisture and Humidity displayed on Cloud

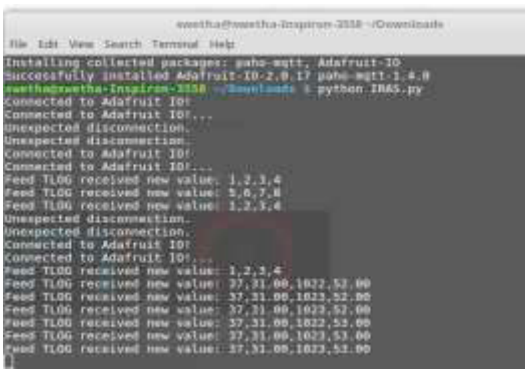

Figure 6. Sensor Readings Obtained

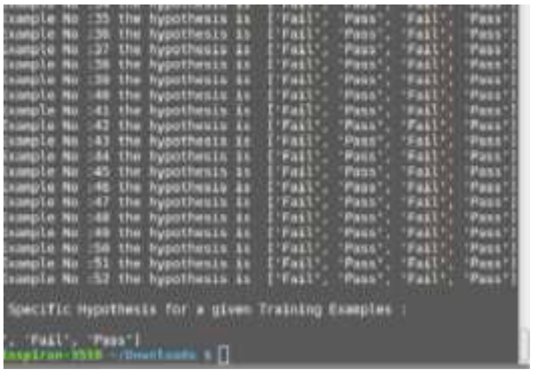

Figure 7. Result Hypothesis Set

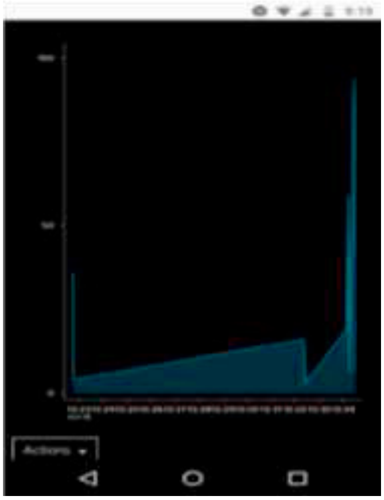

Figure 8. Graphical display of readings on Ada fruit IO.

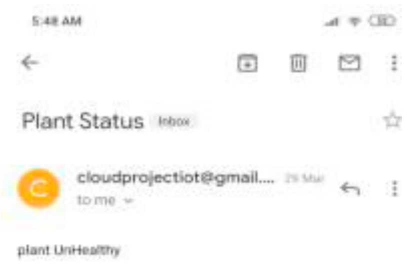

Figure 9. Plant Health status displayed in mail

Figure 4 shows the recorded pH and temperature readings. Figure 5 shows the soil moisture and humidity readings on cloud. Figure 6 shows the readings on Serial Monitor and Figure 7 shows the hypothesis results on Serial Monitor. Figure 8 shows the graphical display of readings on Ada fruit IO while Figure 9 shows the plant health prediction as sent through mail.

Table 2. Comparison of existing methods and proposed method.

	EXISTING METHOD	PROPOSED METHOD
Cost	The existing automatic plant watering systems are not cost efficient as they use expensive robots for moisture sensing and chargeable web services.	With the use of cost efficient sensors and a free-of-cost cloud platform, the system is extremely cost efficient.
Computational complexity	The computational complexity is typically high.	The computational complexity of the findS algorithm we use is relatively low with a time complexity of O(n)

Table 2 continued…

Memory	Most existing systems do not use cloud services and come with in-system storage giving rise to memory constraints.	The cloud platform used provides a convenient and memory-efficient way of storing and analysing the recorded data.
Usefulness	The existing automatic irrigation systems do not come with plant health prediction features.	The proposed system comes with a prediction facility coupled with an automatic e-mail service to alert the user about the health status of the plant.
Additional Features	Most existing systems do not come with an automatic fire detection feature.	The system comes with an effortless and inexpensive fire detection facility that sets off an alarm in case of any fire accident.

Table 2 compares and contrasts features of the proposed system with that of the systems currently in use.

5. Conclusion:

This design implements a Plant watering system that is innovative, time-saving, user-friendly in addition to being more efficient than the currently existing systems. Four soil and environment parameters such as pH value, soil moisture, flame and temperature values are measured to decide whether the plant needs irrigation. The system also predicts the health status of the plant and sends an e-mail alert to the person concerned. Owing to the server updates, the user can be informed about the crop health and environments conditions anytime and anywhere.

References:

[1] Kansara.K, Zaveri.V, Shah.S, Delwadkar .S, Jani.K.(2015) "Sensor Based Automated Irrigation System with IOT:A technical review" International Journal of Computer Science and Information Technologies (IJCSIT), 6(6): 5331-5333.

[2] Marios C, Angelopoulos, Nikoletseas S, Theofffaaanopoulos G.C (2011) "A smart system for garden watering using wireless sensor networks" Mobiwac 11 Proceedings of the 9th ACM international symposium on Mobility management and wireless access, 167-170.

[3] Atta R, Boutraa T, Akhkha A (2011) "Smart Irrigation System for Wheat in Saudi Arabia Using Wireless Sensors Network Technology" International Journal of Water Resources and Arid Environments 1(6): 478-482

[4] Kumar A, Kamal K, Arshad M.O, Mathavan S, Vadamala T "Smart irrigation using low-cost moisture sensors and XBee-based communication" IEEE Global Humanitarian Technology Conference(GHTC 2014)

[5] Gainwar.S.D, Rojatkar.D.V.(2015). "Soil Parameters Monitoring with Automatic Irrigation System "International Journal of Science, Engineering and Technology Research (IJSETR), 4(11):3817-3820.

[6] Archana, Priya. R, (2016)."Design and Implementation of Automatic Plant Watering System" International Journal of Advanced Engineering and Global technology (IJAEGT), 4(1):1567-1570.

[7] Balaji.V.R, Sudha.M, (2016) "Solar Powered Auto Irrigation System". International Journal Of Emerging Technology in Computer Science and Electronics (IJETCSE), 20(2):203-206.

[8] Gondchawar N, Kawitkar. R.S (2016) "IOT Based Smart Agriculture" International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), 5(6): 838:842.

[9] Roopaei M, Rad P and Choo K.K.R (2017) "Cloud of Things in Smart Agriculture: Intelligent Irrigation Monitoring by Thermal Imaging", IEEE Cloud Computing, 4(1): 10-15.

[10] Rawal S (2017) "IOT Based Smart irrigation system", International Journal of Computer Applications, 159(8): 7-11.

[11] Garcia. A.M, Garcia. I.F, Poyato.E.C, Barrios.P.M, Diaz.J.A.R (2018), "Coupling irrigation scheduling with solar energy production in a smart irrigation management system", Journal of Cleaner Production, 175: 670-682.

[12] Touati F, Al-Hitmi M, Benhmed K, Tabish R (2013) "A fuzzy logic based irrigation system enhanced with wireless data logging applied to the state of Qatar", Computers and Electronics in Agriculture, 98: 233-241.