**International Journal of Agricultural and Environmental Information Systems**
Volume 11 • Issue 4 • October-December 2020

# An Intelligent Irrigation Scheduling and Monitoring System for Precision Agriculture Application

RajinderKumar Mallayya Math, Department of Electronics and Communication Engineering, B.L.D.E.A's V.P. Dr. P.G. Halakatti College of Engineering and Technology, Vijayapur, India & VTU-RRC, Belagavi, India

https://orcid.org/0000-0002-2855-6233

Nagaraj V. Dharwadkar, Department of Computer Science and Engineering, Rajarambapu Institute of Technology, Uran Islampur, India

https://orcid.org/0000-0003-3017-0011

## ABSTRACT

In spite of technological advancements, the farm productivity of Indian agriculture is still on the lower side. The underlying reason for poor farm productivity in India is due to the inefficient usage of agricultural inputs, resulting in low or poor-quality agricultural yields. Water happens to be one of such imperative agricultural input that has a huge impact on agricultural productivity. Precision agriculture systems can take care of irrigation requirements by optimally and efficiently using irrigation water for producing crops having superior quality and quantity. This work proposes a smart irrigation system that can efficiently manage the water requirements of the crop for its optimal growth. The irrigation schedules are developed using a feed forward neural network model that can predict the variation in the soil moisture considering the environmental factors such as temperature, humidity, atmospheric pressure, and the rain. The results indicate the effectiveness of the developed system in predicting the soil moisture with mean square error as low as 0.13 and the R value as high as 0.98.

## KEYWORDS

Feed Forward Neural Network, Internet of Things, Machine Learning, Monitoring System, Precision Agriculture, Smart Irrigation

## INTRODUCTION

Water happens to be the most vital element for sustaining life on the Earth both for humans and animals. Studies show that humans can live without food for three weeks but when it comes to water, humans cannot sustain more than three days at a stretch. The same theory applies to the crops that are grown in the field; they do require water. The depleting levels of the water table, irregular monsoons, climate change (Aryal, J.P., et al. 2019), water contaminations are posing as serious hurdles for developing a sustainable agriculture system (Tripathy S, 2019). In agriculture, even with the irrigated lands with adequate irrigation sources (S. Latha, 2019), the utilization of water for irrigation is not strategic. The problem that needs immediate attention is, how water, being a limited and vital resource can be smartly used for irrigation purpose to produce good quality yields capable of fetching good returns to the farmers. Precision Agriculture (PA) system would definitely be a suitable solution for most of the issues arising in the agricultural domain including irrigation. In order to provide efficient

DOI: 10.4018/IJAEIS.2020100101

water utilization, the PA system needs to be tailor-made for the farmers especially from developing countries. The implementation success of the PA system largely relies on the implementation costs, implementation complexity, deployment time, and ease of maintenance. To ensure cost-effectiveness, the field sensor-based approach is recommended. PA system using field-based sensor approach for data collection, IoT for providing remote monitoring of parameters and using intelligence in the form of ML-based predictive model to provide closed-loop control of field parameters would definitely transform the traditional agriculture into a sustainable one. The reach of the Internet in every nook and cranny of the world has created a huge demand for applications involving *things* rather than *peoples*. Currently, there are around 8.3 Billion IoT connected devices worldwide and it is predicted that by 2025, this number will be more than double what it is today (21.5 Billion). IoT has already started benefitting the users worldwide in all the sectors be it healthcare, industry, education or agriculture. Though the implemented PA systems have seen the progress and success in the agricultural sector in countries like Australia (Jochinke et al. 2007), Belgium, Canada, and the United States to name a few, still the major chunk of farmers are yet to harness the rewarding benefits of it. Going by the literature, there are many implementations of IoT in the agriculture domain. Agriculture can be thought of as a complex system, which consists of sub-systems like soil preparation, seed implanting, irrigation, fertilization, weeding, harvesting, sorting, storing and transportation.

IoT was the main driving force for the development of the agricultural systems providing some of the outstanding solutions to the problems being faced by the farmers in the agricultural and the related domains. Also, security remains the main concern when IoT is involved in providing enterprise and business solutions to the customers. Limited researches focused on the security part of IoT implementations in agriculture as in (L. Vidyashree and B. M. Suresha, 2019), where an encryption method was proposed for securing agriculture data.

The work carried out in this research attempts to design a highly secured agricultural field monitoring and irrigation scheduling system by using IoT and Artificial Neural Network (ANN) based predictive model. The IoT part is responsible for data collection, storage, and visualization. The feed-forward neural network (FFNN) model uses the locally generated datasets from the IoT cloud-server as model inputs. The performance measurement of the FFNN predictive model was done based on MSE and R. The model was able to accurately predict the moisture values in the field with low values of MSE and high values of R, apart from this, the other the salient features of the proposed system are that the developed system uses of low-cost and easily available sensors, the main center of attraction of the developed system is the ESP32 DevKit V1 which hosts an ESP32 SoC MCU (capable of providing high security, ultra-low power requirement with built-in Wi-Fi and dual Bluetooth modules), open-source hardware/software platforms for prototyping and programming, open-source *ThingSpeak*™ IoT platform and API for data storage and visualization which also provides MATLAB® analytics on the cloud. Finally, an Android App is developed by using the *Blynk* platform for providing user-friendly irrigation control and automation in the field along with user notification in the form of email and SMS.

The rest of the paper is organized as follows. Section 1 provides an in-depth literature review of similar implementations. Section 2 deals with the Prototype Design Ecosystem, highlighting the hardware and software requirements, technical specifications along with the platform as a service. Section 3 uses a block diagram approach that highlights the proposed system with various block descriptions. Section 4 describes the in-field experimental setup for irrigation control and location of the setup. In Section 5, the results are tabulated for the proposed ANN model that provides irrigation control and discussion is carried out justifying the use of sensors, microcontrollers, and the ANN model. In section 6, a detailed comparison is presented with similar implementations in the area of interest and highlighting the salient features of the developed system and also the cost analysis is done and compared with other similar systems. Section 7 concludes the paper by revealing important findings and indicating the importance of the proposed system in the present context along with some of the improvements that can be taken up in the future.

## LITERATURE REVIEW

The intensive literature survey carried out in this work helped in understanding the state of art techniques and technologies used by the researchers in developing IoT based solutions towards the agriculture domain. The literature survey also helped in identifying the potential gap that can be filled up by the developed system. Some of the important and relevant contributions by the researchers in the field of precision agriculture are discussed here. A literature review by (Antonis Tzounis et al. 2017; Olakunle Elijah et al. 2018) where the former highlighted the latest IoT technologies that are currently positively impacting the agricultural sector, future impacts of these technologies on the farmers along with the possible challenges that might be a hurdle on the way to sustainability and possible solutions for the challenges while the latter dealt with the benefits and challenges concerned with the IoT data and analytics in the agriculture domain. The key issue pertaining to the challenges were identified as security and the deployment cost.

Sherif Abdelwahab et al. (2016) proposed a global architecture that uses edge computing platforms as a cloud agent for discovering and visualizing sensing resources pertaining to the IoT devices and proposed a new service model for cloud platform as Sensing as a Service. The new trend begin was the use of UAV as a means of monitoring the field parameters along with IoT. Another implementation involving UAV and IoT for addressing the automatic irrigation system J. Aleotti et al. (2018) provided automatic irrigation for the tomato fields by using architecture consisting of a sever, a mobile app and IoT devices for irrigation control.

Soon, it was felt that the cloud platform should be private so that the security is not compromised and also the redundancy and delays in the communication between the sensors and cloud or cloud to users can be minimized. One such implementation was by Tomo Popovic et al. (2017) which developed a private IoT enabled platform with an aim of promoting research in the field of agriculture and ecological monitoring systems.

A safe and reliable cloud-based PA system N. Pavon-Pulido et al. (2017) provided remote monitoring of crops and planning of agricultural tasks from any smart device (PC, smartphone or a tablet). These benefits were demonstrated with several experiments and on-field deployment of the cloud-based approach towards the system. Additionally, Google App Engine Platform as a Service (PaaS) was used in developing the software architecture. Before the crops are selected for cultivation, it is a good idea to check the soil type, its contents in the form of micro-nutrients which would suggest the possible type of crop that can be grown to obtain a higher and premier quality of yields. Md Eshrat E Alahi et al. (2018) used IoT for application involving soil property monitoring. The soil property being nitrate, which was measured using an FR4-based interdigital sensor. It was shown that for longer use of the system, LoRa was preferred over Wi-Fi protocol due to low-power, an energy-saving property of LoRa. Another similar implementation concerned with the soil parameter sensing was based on the design of closed-loop irrigation system Levente J Klein et al. (2018) employing a cloud-based system that used satellite-based images for identifying the irrigation requirements of the crops. The system concentrated on two parameters of agriculture, resource optimization (water) and maximizing yield. Jirapond Muangprathub et al. (2019) also implemented a WSN based system that provided water to the agricultural crops optimally, additionally aimed at providing crop field data management over a smartphone and web application.

Meeradevi et al. (2019) also worked on similar lines of providing optimal water usage in the field by the use of IoT and WSNs. The water requirement by the crops in a particular area was monitored and irrigation was carried out based on the requirements. Authors claimed to have developed the model with low-cost, which would help the farmers, especially from India. Suresh Koduru et al. (2019) came up with a smart irrigation system again riding on the IoT. A component water preservation mechanism was used along with the parameters like soil moisture and weather forecast to effectively utilize the water resources and to avoid groundwater depletion in the future. The irrigation monitoring was provided in the form of a farmer's cockpit. Vaishnavi Bheemarao Joshi and R. H. Goudar, (2019) developed an irrigation system harnessing the capabilities of IoT to control electrical submersible pumps by using smartphones.

The main issue pertaining to the irrigation of farmland is the limited availability of electricity in rural areas. The developed system helps farmers by notifying them of the availability of electricity by an SMS. Based on the notification, the submersible pump used for irrigation can be precisely controlled from a remote location without the requirement of physical presence in the field. The research work by R. Raut et al. (2018) resulted in the development of a multi-parameter monitoring system for soil properties, fertilization, along with irrigation system by using IoT. The main target was to develop an automatic irrigation system that could also provide information about the vital soil nutrients like nitrogen (N), phosphorus (P) and potassium (K). The authors claim the system was capable of helping farmers in saving their time, money and labor. Another implementation (Radu Dobrescu et al. 2019), focused on context-aware multi-parameter monitoring system using IoT and cloud support. The system was able to bring IoT, cloud computing and context awareness under the same umbrella, well supported by a multi-layered architecture capable of providing real-time process control in the agriculture domain. The capability of the system was demonstrated with a case study where the system was implemented on the IBM Bluemix IoT platform. Wen-Liang Chen et al. (2019) also developed an IoT platform named "AgriTalk" for soil cultivation outdoors. The experimentation involved the cultivation of turmeric with the use of IoT and it was shown that the system provided immunity against the problems arising because of the soil cultivation. According to the authors, the AgriTalk was able to enhance the quality of turmeric significantly.

Apart from the field parameters, the environmental factors are also equally responsible for deciding the optimal growth of the in-field crops, so that the crops are provided with the required input resources at the right time. Jorge Gomez et al. (2019) came up with a crop monitoring system based on IoT. A case study was provided for small crops in a rural area, which is deprived of internet connectivity due to the coverage issue. Crop processing stations used text messages in the form of SMS to convey the variations in the field parameters to the users. The data is gathered and also communicate to the cloud-based platform where the users are supposed to subscribe in order to get the notification by using Message Queuing Telemetry Transport (MQTT) protocol. A weather system also acts like an environment monitoring system which relies on the environmental sensors to provide insights into the current weather and also the weather behavior for a couple of days to come. Apart from the growth of crops, the weather also influences other agricultural tasks like irrigation, the application of agrochemicals or fertilizers and harvesting. A local weather station using IoT was developed by (R. K. M. Math and N. V. Dharwadkar 2018), which provided local weather at an agricultural site. The specialty of the proposed system was its low- cost nature (sensors, microcontroller, open-source hardware and software solutions).

Juan Carlos Guillermo et al. (2019) developed an IoT architecture involving WSN for monitoring the agricultural parameters with a case study pertaining to the Cacao crops. The developed architecture targets medium holding farmers by providing a multi-platform application that can be used to monitor climatic conditions along with soil properties which have a strong influence on cacao growth and production. Another architecture for IoT Nurzaman Ahmed et al. (2018) involving fog computing along with Wi-Fi-based network spanning rural areas provided precise control of agricultural farms. The developed network structure was evaluated considering the performance metrics as coverage range, throughput, and latency.

The popularity of IoT in the agriculture sector drew the attention of many researchers towards the horticulture too, which differs from agriculture not only in the scale but also in terms of crop variety involving vegetables and flowers. An IoT based farming system for horticulture was developed by Ajay Mittal et al. (2018) taking the case study of Cabbage and Capsicum. Some of the sensors selected for the experimentation were, air temperature, atmospheric pressure, soil humidity and moisture, wind speed and direction, hourly rainfall and leaf wetness sensor. Authors claim to have obtained a reduction in the cost by about 20% while the improvement in the yield amounted to 10%.

Moving a step ahead, integration of IoT with machine learning by Nagaraj V. Dharwadkar and Vandana R. Harale (2019) was used for monitoring climatic parameters within a greenhouse for tomato crop. The developed system utilized the locally generated data from the sensors like temperature, humidity, light intensity, pH value, and $CO_2$ concentration to effectively model and control the climatic conditions within the greenhouse for optimal growth of tomatoes. Another similar integration of IoT and a genetic algorithm was proposed by Archana P. Kale and Shefali P. Sonavane (2018) for developing a smart farming decision support system that handled optimization, uncertainty with a reduced number of features.

Based on the literature survey, it can be pointed out that most of the researches concentrated to solve the agricultural issues by either using only IoT (no ML) or by developing ML models for some publicly available agricultural datasets. Only limited researches have implemented an integrated version of IoT with ML as in Nawandar, N. K., & Satpute, V. R. (2019) wherein a smart irrigation system was developed with intelligence provided by a neural network-based model. Another similar implementation Goldstein et al. (2018) resulted in an irrigation recommendation system using the Gradient Boosted Regression Tree model. The work carried out in this paper also aims at providing end-to-end irrigation scheduling system by integrating IoT with an ANN model, and the developed system has the important characteristics of quick deployment capability, and its low-cost and user-friendly nature mainly focussing on to the farmers who are having small agricultural lands and are not in a position to procure high-end irrigation systems for solving their irrigation issues.

## PROTOTYPE DESIGN ECOSYSTEM

### Hardware and Software Requirements

The developed irrigation scheduling and control system for precision agriculture can be thought of as consisting of three main components. The sensors, which are required to sense the field parameters, these sensors are embedded on the board along with a microcontroller/microprocessor. The sensors along with microcontroller/microprocessor constitute a single node of a wireless sensor node, these sensor nodes can be spatially and strategically deployed to cover the required portion of the field. IoT architecture consists of IoT devices (microprocessors/controllers and IoT Gateways) and a cloud server providing Platform as a Service (PaaS).

The framework consists of sensors, ESP32 based prototyping board (open-source hardware), open-source software (Arduino IDE) and open-source IoT platform. The prototyping board ESP32 Dev Kit V1 hosts a powerful, low-cost and low power ESP-WROOM-32 MCU integrated with a Wi-Fi and a dual-mode Bluetooth module along with a dual-core 32-bit Tensilica Xtensa LX6 microprocessor. The sensors were selected to provide the precise on-field parameters which include soil moisture sensor (FC-28+LM393), also known as soil hygrometer, a rain sensor (YL-83+LM393), a DHT22 based air temperature and humidity sensor, an atmospheric pressure sensor (BMP180). Table 1 shows the hardware and the software components used for the development of the proposed system.

### Technical Specifications

The microcontroller is the key hardware resource used for the development of the irrigation scheduling system. The sensors are selected keeping an eye on the cost factor while not compromising the accuracy. As the MCU happens to be the heart of the application, the prototyping board uses ESP-WROOM-32 MCU to provide high security in data communication along with its unmatched capability of requiring ultra-low operating powers and added advantage of having a lower price tag comparatively. Table 2 illustrates the main technical features of the ESP32 DevKit V1 prototyping board which offers many outstanding capabilities and unmatched features making it a suitable choice in applications involving continuous monitoring of physical parameters.

**Table 1. Hardware and software component requirements**

| Sl. No. | Parameter/Details | Specifications |
|---|---|---|
| 1. | MCU | Tensilica Xtensa LX6 dual core 32-bit |
| 2. | Maximum Operating Frequency | 240 MHz |
| 3. | Operating Voltage | 3.3 V |
| 4. | Analog Input Pins | 12-bit, 18 Channel |
| 5. | DAC Pins | 8-bit, 2 Channel |
| 6. | Digital I/O Pin | 39 (of which 34 is normal GPIO pin) |
| 7. | Static RAM | 520 KB |
| 8. | Built-in sensors | Touch, temperature and hall effect sensors |
| 8. | ROM | 448 KB |
| 10. | Communication | SPI(4), I2C(2), I2S(2), CAN, UART(3) |
| | Wi-Fi | 802.11 b/g/n |
| | Bluetooth | Classic and Bluetooth Low Energy (BLE) |

## Platform as a Service

Apart from using the Arduino IDE as open-source software for programming ESP 32 SoC, *ThingSpeak*™ IoT platform was selected to provide Platform as a Service (PaaS) to provide services such as data aggregation, data storage, data visualization, and data analysis.

The selection of the IoT platform was primarily done by keeping in mind the implementation complexity, deployment time, and services offered to the users. The only one limitation that can be found with the *ThingSpeak*™ platform is a limited number of data fields (8 data fields), but to get started in the field of IoT and building innovative prototypes, *ThingSpeak*™ would be a great choice.

The support of MATLAB® to *ThingSpeak*™ helps in utilizing the services offered by MATLAB® in the cloud. Apart from the aforementioned services, *ThingSpeak*™ also provides other user services in the form of metadata that can be used to provide additional information about the data channel. The metadata can be provided in the form of JSON, XML, or CSV data. Data mashup can be obtained by integrating the data corresponding to different sources (channels) that can be integrated to altogether to form a new dataset. For example, temperature data corresponding to field 1of channel 850870 can be combined with humidity data corresponding to field 1 of channel 856104 to create new data.

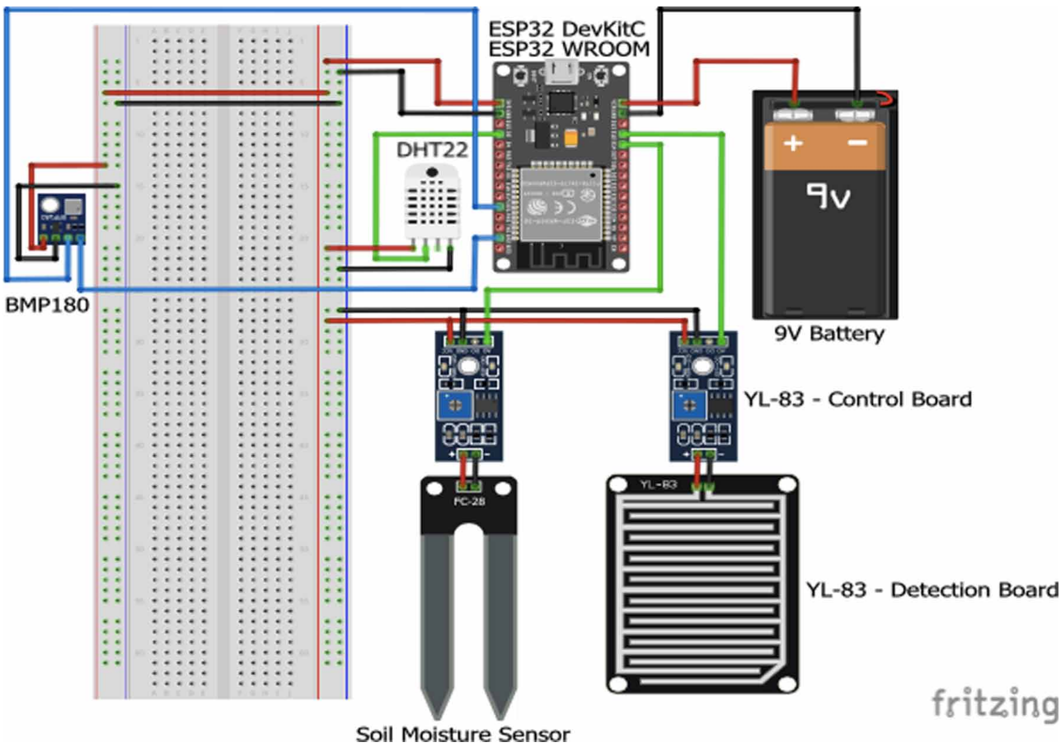
**Table 2. Technical specifications of ESP32 DevKit V1**

| Sl. No. | Hardware Components | Parameters | Sl. No. | Software Components |
|---|---|---|---|---|
| 1 | ESP32 DevKit V1 | Prototyping Board | 1 | Arduino IDE (1.8.9) |
| 2 | DHT 22 | Temperature and Humidity Sensor | 2 | *ThingSpeak*™ |
| 3 | BMP180 | Atmospheric Pressure Sensor | 3 | *Blynk* App 2.27.5 |
| 4 | YL-83+LM393 | Rain Sensor | 4 | *Fritzing* 0.9.3 |
| 5 | FC-28+LM393 | Soil Hygrometer | | |

### Breadboard Connection and Schematic Diagram

To make the design understandable and repeatable, the breadboard wiring and connection diagrams are very helpful. The breadboard and circuit connection diagram were drawn using *Fritzing* software, easy to use and again an open-source platform for the users, as shown in Figure 1 and Figure 2, respectively. The diagrams are self-explanatory.

**Figure 1. Breadboard wiring and connection diagram**




## PROPOSED SYSTEM

The proposed system provides a low-cost means of monitoring key agricultural parameters to provide precise irrigation control by utilizing the remote connectivity and storage functionality offered by IoT with cloud analytics provided by the neural network-based model. The system uses temperature, humidity, soil moisture, rain and atmospheric pressure as the input parameters of the agricultural field. The block diagram is as shown in Figure 3.

The proposed system consists of five parts:

1. **Data Gathering:** The data gathering part consists of the agricultural field parameter sensors, the parameters being temperature and humidity, absolute and relative atmospheric pressure, soil moisture and the rain. The raw data from the sensors is collected by the microcontroller and is pre-processed. As depicted in Figure 3, the sensors along with the ESP32 DevKit V1 hosting ESP-WROOM-32 microcontroller constitutes a single node of a sensor network that has the

Figure 2. Schematic diagram

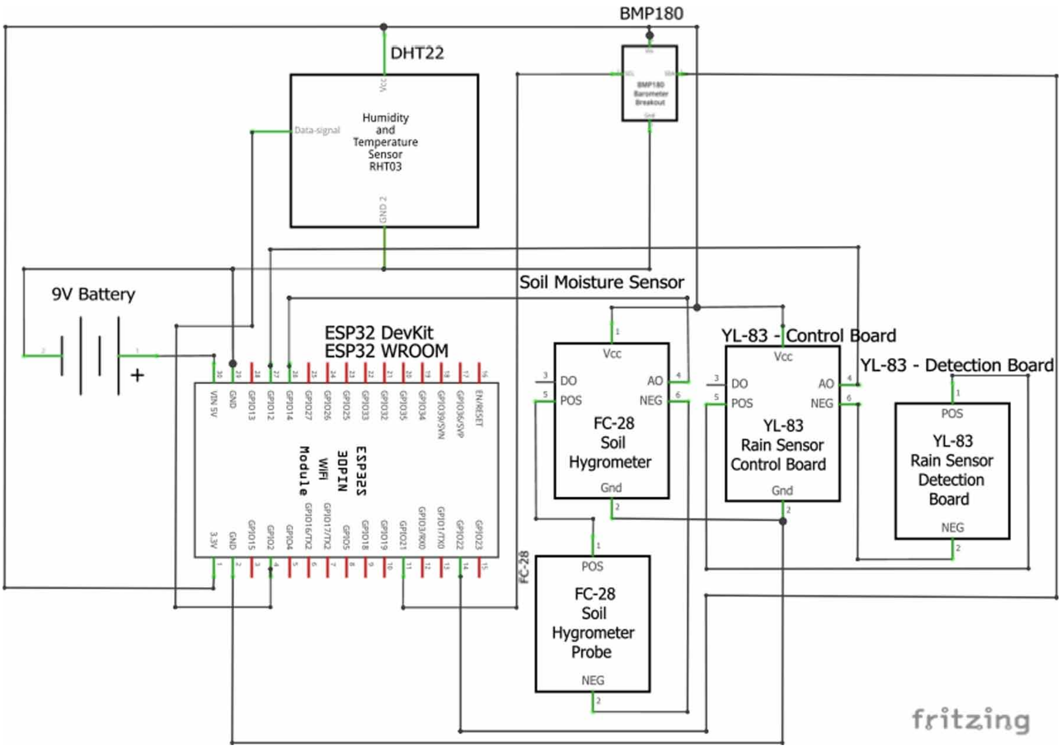
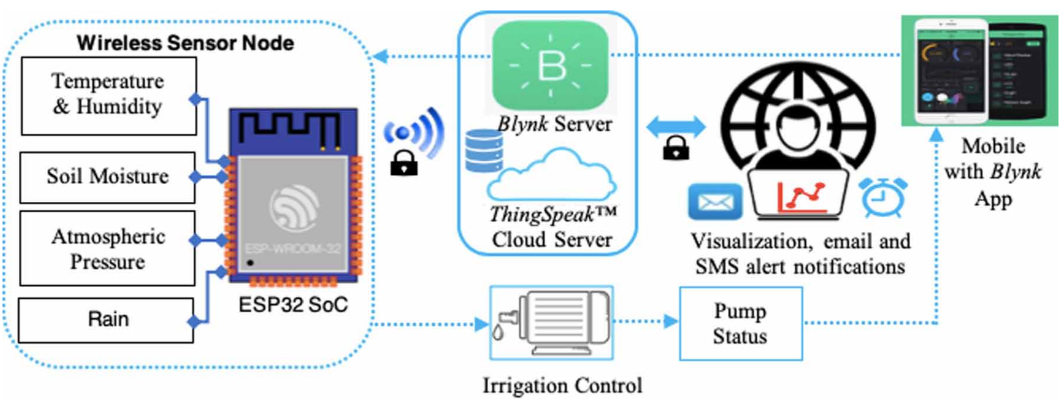


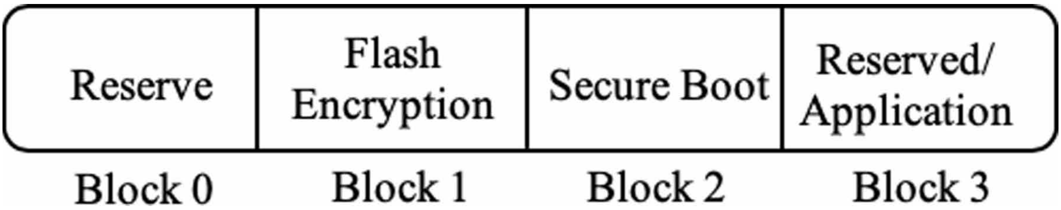

Figure 3. Block diagram of the proposed framework




capability of wirelessly transmitting the data to IoT gateway or router by using the built-in Wi-Fi functionality of ESP32 SoC.

2. **Securing the IoT Design (Security Features of ESP32):** To provide the users with security at the device level, it is required that the hardware devices (for example microcontrollers) should always make sure that the firmware data is not accessible by the unintended users. Once the data is obtained from the sensors, the ESP32 SoC MCU comes into action by securely logging the data on to the *ThingSpeak*™ cloud server via the IoT gateway or the router. When the devices are

Figure 4. ESP32 eFUSE overview




shipped, the application firmware data is usually stored in the flash memory which is external to SoC. The firmware data is vulnerable to the attacks caused by unauthorized access to the firmware data stored in the flash. This act of unauthorized access to the firmware data tenders a serious threat as the data can be faked, modified or tampered intentionally by a potential hacker. To overcome such attacks, ESP32 SoC provides a high level of security by using eFUSE which is a one-time programmable memory to prevent unwanted tampering and modification of the firmware data. The overview of eFUSE used by ESP32 is as shown in Figure 4 with various fields in it.

The ESP32 uses a 1024-bit eFUSE, having four blocks starting with block 0 and goes up to block 4 and each block has 256 bits as shown. The components of interest in eFUSE are:

- ***Flash Encryption:*** The flash encryption block of ESP32 provides support for the application firmware data stored in flash is always encrypted. Hence the manufacturers of ESP32 can ship the firmware which is encrypted, making it secure and tamper-proof. The encryption uses AES keys which are stored in eFUSE and do decryption as and when required. Thus, memory read and write operations on the flash memory are secured. The decryption is possible only with the key which is securely locked in eFUSE. The flash encryption procedure is as shown in Figure 5.
- ***Secure Boot:*** The secure boot feature of ESP32 blocks the untrusted software from executing from the flash, that is if the software is signed by a known entity then only the software is able to execute and get access to the firmware data. If any of the bits of the software bootloader and the application firmware have tampered, the firmware becomes untrusted and the ESP32 will not run or execute the untested software. The secured boot feature offered by ESP32 is as shown in Figure 6 which helps in blocking the untrusted software from executing the application firmware.

The above two features of ESP32 provide a high level of security which is demanded by most of the IoT based applications. Another security feature of ESP32 is at the Transport Layer Security

Figure 5. Flash encryption in ESP32

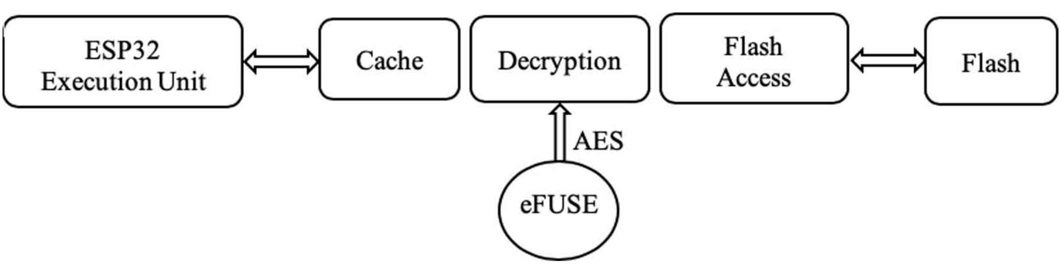
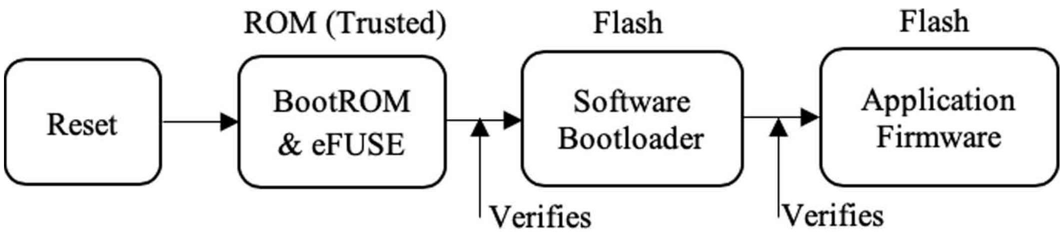


Figure 6. Secure boot feature of ESP32




(TLS) which adds security when the application or the hardware is connected to the Internet for data exchange between the hardware and the cloud platforms.

3. **Data Logging on to ThingSpeak™ for Visualization:** It is evident from the aforementioned security features of ESP32 that the security requirements involving IoT are fulfilled. Apart from providing security, wireless communication capability in ESP32 SoC comes in the form of built-in Wi-Fi and Bluetooth modules that provide wireless connectivity to the Internet at no extra cost. Again, the Wi-Fi connectivity can be obtained either by using Wi-Fi Protected Access (WPA) or WPA2. It is desirable to use WPA2 security as it uses a stronger wireless encryption technique when compared to WPA.

The channel can be kept private or made public, if made public the channel can be shared with other users. Once the channel setting is done and saved, API keys namely READ API KEY can be used for reading the channel fields or charts, while WRITE API KEY can be used for updating the data field in the channel. Apart from providing visualization, the data from the sensors get stored in the channel with corresponding time stamps. The data collected at *ThingSpeak*™ cloud can be exported in the form of a CSV file. The recent data (up to 100 samples) can also be imported in other formats like JSON and XML.

The sensor data is logged on to the cloud in the form of channels, comprising of different fields (maximum of eight fields can be used) and is visualized in the form of Google gauges or charts. The channel feed on the cloud was updated every 15 seconds. The data corresponding to the various fields of the cloud server channel is temperature, absolute pressure, relative pressure, humidity, rain, soil moisture, and dew point. Out of the seven field parameters, the first six parameters were directly measured using sensors while the calculation of the seventh parameter (dew point) was based on temperature and humidity using the following equations, referred from (Blynk documentation) as shown in Equation 1:

$$t_d\left(t, RH\right) = T_n * \left[\frac{log\left(\frac{RH}{100}\right) + \frac{m * t}{T_n + t}}{m - log\left(\frac{RH}{100}\right) + \frac{m * t}{T_n + t}}\right] \tag{1}$$

where $t_d$ = dew point temperature in °C; $t$ = actual temperature in °C;

$RH$ = actual relative humidity in percent (%); $m = 17.62$; $T_n = 243.12$ °C.

It is also possible to calculate the absolute humidity value in terms of grams/m$^3$ by knowing the values of temperature and RH as shown in Equation 2:

$$d_v\left(t, RH\right) = 216.7 * \left[\frac{\frac{RH}{100}.A.exp\left(\frac{m * t}{T_n + t}\right)}{273.15 + t}\right] \tag{2}$$

where $d_v = \text{absolute humidity in grams / m}^3$ ; $t$ = actual temperature in °C

$RH$ = actual relative humidity in percent (%); $T_n$ = 243.12 °C; $m$ = 17.62; $A$ = 6.112 hPa

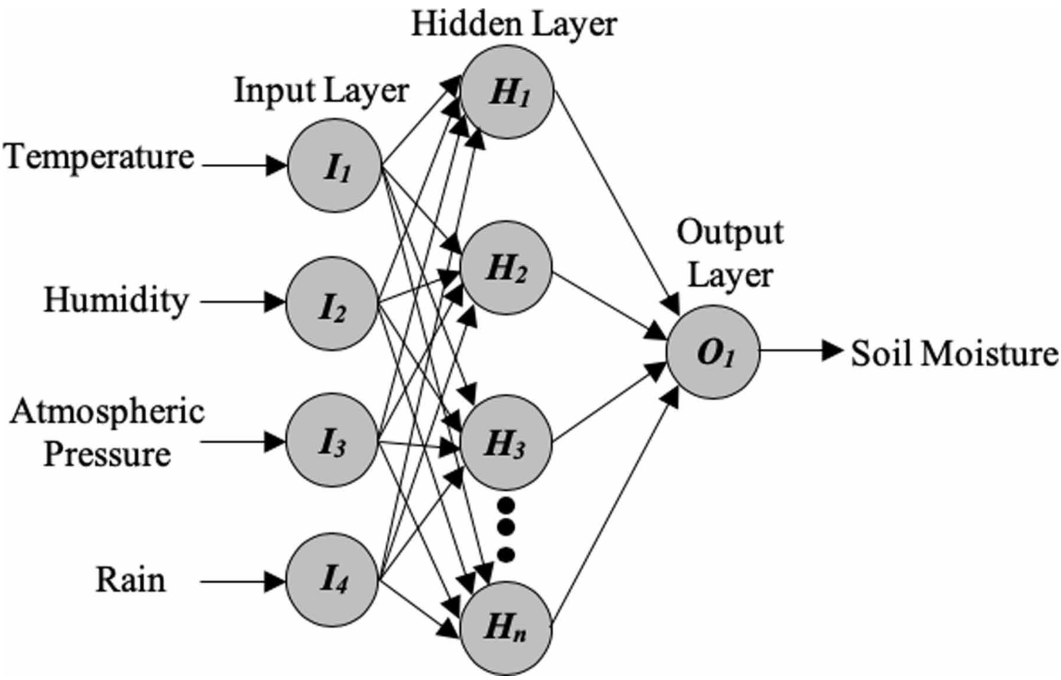
4. **Blynk Platform and Android App Development:** A *Blynk* app is a handy tool for controlling the IoT devices over a smartphone. *Blynk* helps in providing manual control of irrigation remotely by using a smartphone with Internet connectivity. The *Blynk* app sends an email to the user whenever the soil moisture level falls below a lower threshold level or crosses the upper threshold limit. To provide data security to the users, a unique authentication token is generated and sent to the registered email of the user which needs to be used in the sketch before uploading it to the prototyping board.
5. **ANN-Based Predictive Model:** After successful data storage and visualization, the historical data collected over time was used to build an ANN-based model. Before the model is built, the data in the channel was refined so as to make sure that the model accuracy and efficiency are improved. An Exploratory Data Analysis (EDA) (more details on EDA can be found in the Results and Discussion section) was carried out to discover the hidden data patterns in the dataset. As the neural networks are very well known for their consistency in providing more accurate predictions when compared to other prediction models, an ANN-based FFNN model was preferred. The feedforward neural networks have a unique characteristic in which the flow of data is only in the forward direction from inputs to the output. The basic structure of FFNN consists of an input layer where the input vectors are assigned to the model, an output layer producing the output vector which depends on the input vectors. Apart from input and output layers, it also has one or more hidden layers consisting of neurons as shown in Figure 7. The input neurons at the input layer are represented as $I_1$, $I_2$, $I_3$ and $I_4$ corresponding to four input independent parameters. The hidden layer neurons can be varied in numbers and are represented as $H_1$, $H_2$, $H_3$ and so on. The output layer has a single neuron as the dependent variable represented as $O_1$.

## EXPERIMENTAL SETUP AND IRRIGATION SCHEDULING

The experimental setup consists of the field sensors (DHT22, BMP180, FC-28+LM393, and YL-83+LM393) along with ESP32 Development Kit V1. The programming was done in Arduino IDE for fetching the data from the sensors and providing irrigation control. The data collected by the microcontroller is displayed on to the serial monitor of Arduino IDE using serial communication between the microcontroller and the system (PC or laptop).

This is required for programming the microcontroller and for troubleshooting in case of any hardware issues. After successful data collection, the microcontroller uses its built-in Wi-Fi module to connect to the available Wi-Fi network or a hotspot both requiring a correct combination of

Figure 7. Structure of the FFNN model used for prediction




SSID and password. The prototype was mounted within a protective housing to withstand the harsh environment in the field as in Figure 8.

The experimental setup was able to ensure that the data from the sensors are securely logged on to the cloud without any interruption, except a few missing values and outliers due to connection issues.

## Experiment Location

The experimental setup was done on a farm in Tikota Taluk of Vijayapur District in North Karnataka with coordinates as 16°50'17.1"N 75°31'12.2"E as shown in Figure 9. The grape field was spread over 1 acre of land. The developed prototype was set to measure the climatic conditions including soil moisture near the root zone as shown in Figure 10.

## Irrigation Control and Scheduling

The irrigation scheduling and control can be clearly understood by referring to the flow chart as shown in Figure 11. After the successful development of the prototype to be installed in the field, the irrigation scheduling and control is brought about by developing an app using *Blynk* on a smartphone. The *Blynk* email widget is set to send an email to the registered user if the current moisture level $SM$ falls below the lower threshold level $SM_{min}$ or above the upper threshold of $SM_{max}$ based on which a manual or automatic control is provided for irrigation. The manual irrigation control is provided by using an Android app built using *Blynk*. The soil moisture is continuously monitored to be between lower threshold $SM_{min}$ and upper threshold $SM_{max}$. If the current moisture value falls below $SM_{min}$, then an alerting email is sent to the user along with an SMS to switch *ON* the water pump. Once the moisture level reaches the upper threshold of $SM_{max}$, again the alert is sent to the user to switch *OFF* the water pump. Thus, ensuring that the moisture level in the soil is just enough for the proper growth of the crops in the field. A *textlocal* API, an email to SMS gateway was used to send SMS from email. The automatic irrigation scheduling was obtained by using an FFNN prediction model to predict the

Figure 8. Developed prototype with a component view

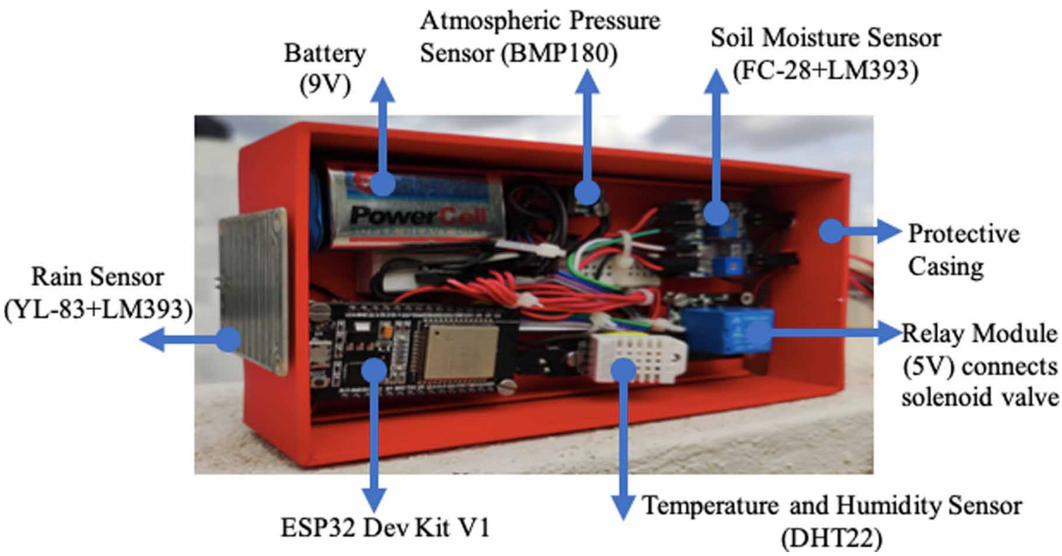



Figure 9. Experiment location


Figure 10. Case study prototype implementation in Grape (Vitis amurensis) field

future values of the moisture based on the input parameters. The prediction of the model was used along with the weather prediction (rain) from dark sky API, to select the irrigation schedule. The rain prediction is obtained in the form of ‘no rain’, ‘moderate rain’ and ‘heavy rain’. Based on the rain prediction by the dark sky, three irrigation schedules were developed each differing in the upper threshold of soil moisture level $SM_{max}$. If the predicted value of moisture requires irrigation and ‘no rain’ is predicted, then irrigation schedule 1 is selected with the upper threshold $SM_{max}$ of 70%. For schedule 2, if the predicted moisture level requires irrigation and chances of rain is ‘moderate’, then the upper threshold of 60% is selected. For schedule 3, if there is a prediction of ‘heavy rain’, then the upper threshold of 50% is selected as in Table 3. If there is rain while the irrigation schedule is *ON*, then the interrupt is sent to turn *OFF* the water pump. While testing the prototype, the schedule 1 was more frequently used there was no, substantial rain in that duration. Using the prototype, the weekly irrigation requirement was reduced by around 10-12 liters of water per irrigation schedule for fully matured vines.

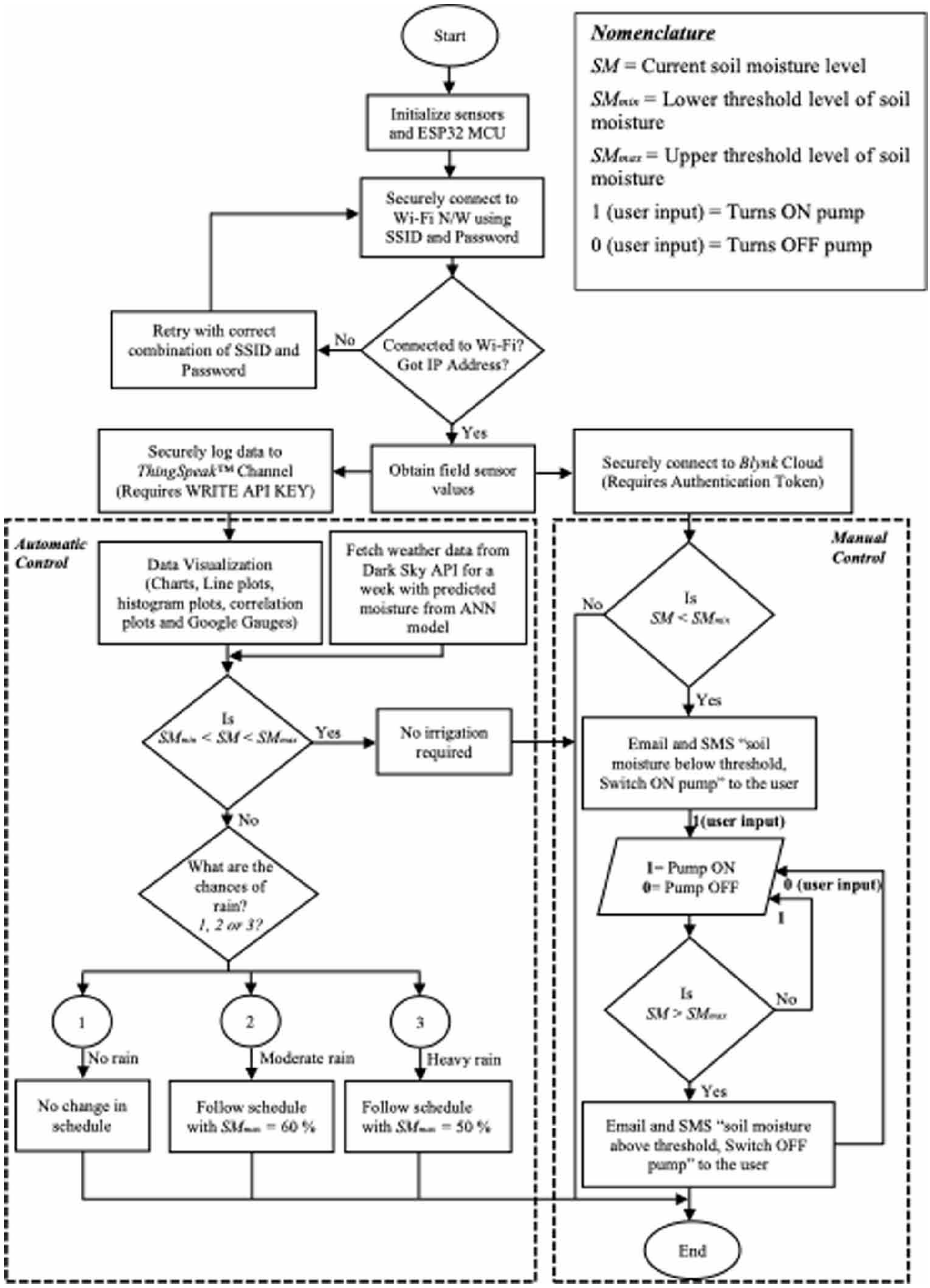
## RESULTS AND DISCUSSION

The data collected at the server was converted into a local dataset containing the input parameters (temperature, humidity, atmospheric pressure, and rain) and the output parameter (soil moisture). The input parameters are the independent variable for the model while the output parameter is the dependent variable. Exploratory Data Analysis (EDA) in the form of descriptive statistics as in Table 4 was used to get insights into the data and understand the pattern and correlations between the predictor and predicted variables. Some of the other parameters that have a high influence on the ML model are Standard Error, Standard Deviation, Sample Variation, and Quartiles. Keeping the computational efficiency of ESP32 SoC and the EDA results in view, the moisture prediction was identified as a regression problem, and since one of the requirements of the developed system was to have high generalization capability, an ANN-based regression model was selected which generalizes well for the new unseen data. The developed PA system for irrigation scheduling was capable of precisely monitoring the agricultural parameters pertaining to irrigation, and the parameters selected for monitoring played a very vital role in ensuring the optimal growth of the crops. To obtain the visualizations for the data collected, the channel ID, READ and WRITE KEYS were used to display the data in the form of spline chart and the gauges as depicted in Figure 12 (a) for temperature and Figure 12 (b) for humidity while Figure 12 (c) shows the use of gauge.

The local dataset consists of 2000 samples generated and exported from the cloud server corresponding to five field parameters. The dataset was divided into input data consisting of four parameters (absolute pressure, relative pressure, temperature, and humidity) while the soil moisture was selected as output or the target data. An ANN-based predictive model was developed to describe the relationship between the predictor parameters and the predicted parameter and thereby also predict the future values of the predicted parameter. The model consists of a multi-input two-layer FFNN in which the hidden layer uses Sigmoid function while the output layer has a Linear function. The model was tested with two sets of data division, in the first set the data were randomly divided into *Training*, *Testing* and *Validation* sets with 70%, 15%, and 15%, while the second division was done with each set having 80%, 10% and 10% of data. The performance of the developed model was tested for loss function Mean Square Error (MSE) and the coefficient of correlation (R). The data division rule yielded a marginal improvement in MSE from 0.14 to 0.13 for division rule of (70:15:15), whereas R value was unaffected as highlighted in Table 5. The MSE is calculated by using the formula given by Equation 3.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{3}$$

**Figure 11. Flowchart for the proposed system**



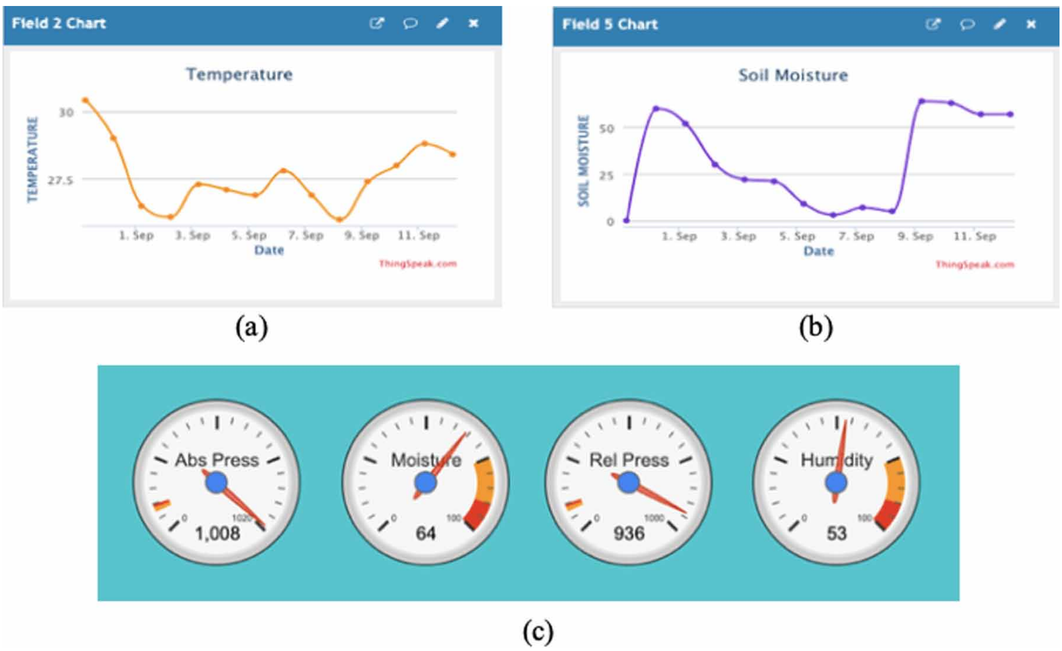

Where n is the number of samples, $Y$ is the actual output while $\hat{Y}_i$ is the predicted value of output.

The MSE and R values were evaluated for the following conditions:

**Table 3. Irrigation schedules with upper and lower thresholds**

| *Sl. No.* | *Irrigation Schedules* | *Lower Threshold ($SM_{min}$)* | *Upper Threshold ($SM_{max}$)* |
|---|---|---|---|
| **1.** | **Schedule 1** | **40%** | **70%** |
| **2.** | **Schedule 2** | **40%** | **60%** |
| **3.** | **Schedule 3** | **40%** | **50%** |

**Figure 12. Data visualization (a) Temperature (b) Soil Moisture and (c) Google gauges**




- Training functions (Levenberg-Marquardt, Bayesian Regularization, Gradient Descent, and Scaled Conjugate Gradient)
- Number of hidden neurons n (5,10 and 15)
- Data division rule (Training: Testing: Validation-70:15:15 and 80:10:10)

When it comes to the training function, Bayesian regularization outperformed the other training functions. The performance of the model showed improvement when the number of hidden layers was increased. The performance of the model was not much affected by the data division rule (70:15:15 and 80:10:10), but the data division rule (70:15:15) yielded better performance in terms of MSE when compared to (80:10:10) rule, while the R-value was same for both. Thus, it can be seen from the results that the best performance was obtained for the Bayesian Regularization function (MSE=0.13, R=0.98 (both for training and testing)) with a number of hidden layers n=15 and data division rule of (70:15:15). For simplicity, only performance characteristics of Bayesian training function is considered, while the results are tabulated for all training functions with a different number of hidden layers and data division rule. The training performance of the model in the form of MSE is as shown in Figure 13 (a), it is seen that the best performance was obtained at epoch 1000. While Figure 13 (b) shows

**Table 4. Descriptive statistics for the dataset**

| Parameters | Humidity | Temperature | Absolute Pressure | Relative Pressure | Soil Moisture |
|---|---|---|---|---|---|
| Mean | 80.96 | 27.87 | 1017.77 | 944.79 | 54.73 |
| Standard Error | 1.00 | 0.14 | 0.28 | 0.25 | 0.97 |
| Median | 83.25 | 28.04 | 1017.71 | 944.77 | 54.29 |
| Mode | 88.20 | 28.95 | 1017.24 | 945.64 | 55.13 |
| Standard Deviation | 7.94 | 1.10 | 2.22 | 2.01 | 7.73 |
| Sample Variance | 63.12 | 1.21 | 4.95 | 4.05 | 59.75 |
| Kurtosis | -0.71 | -0.26 | 0.90 | 1.18 | -1.01 |
| Skewness | -0.42 | -0.63 | -0.31 | -0.35 | 0.12 |
| Range | 31.36 | 4.38 | 12.32 | 11.44 | 28.06 |
| Minimum | 64.73 | 25.10 | 1010.52 | 938.08 | 41.25 |
| Maximum | 96.09 | 29.48 | 1022.84 | 949.52 | 69.31 |
| First Quartile (Q1) | 74.97 | 27.27 | 1016.46 | 943.60 | 48.82 |
| Second (Q2) | 83.25 | 28.04 | 1017.71 | 944.77 | 54.29 |
| Third (Q3) | 86.66 | 28.73 | 1019.03 | 945.80 | 59.80 |

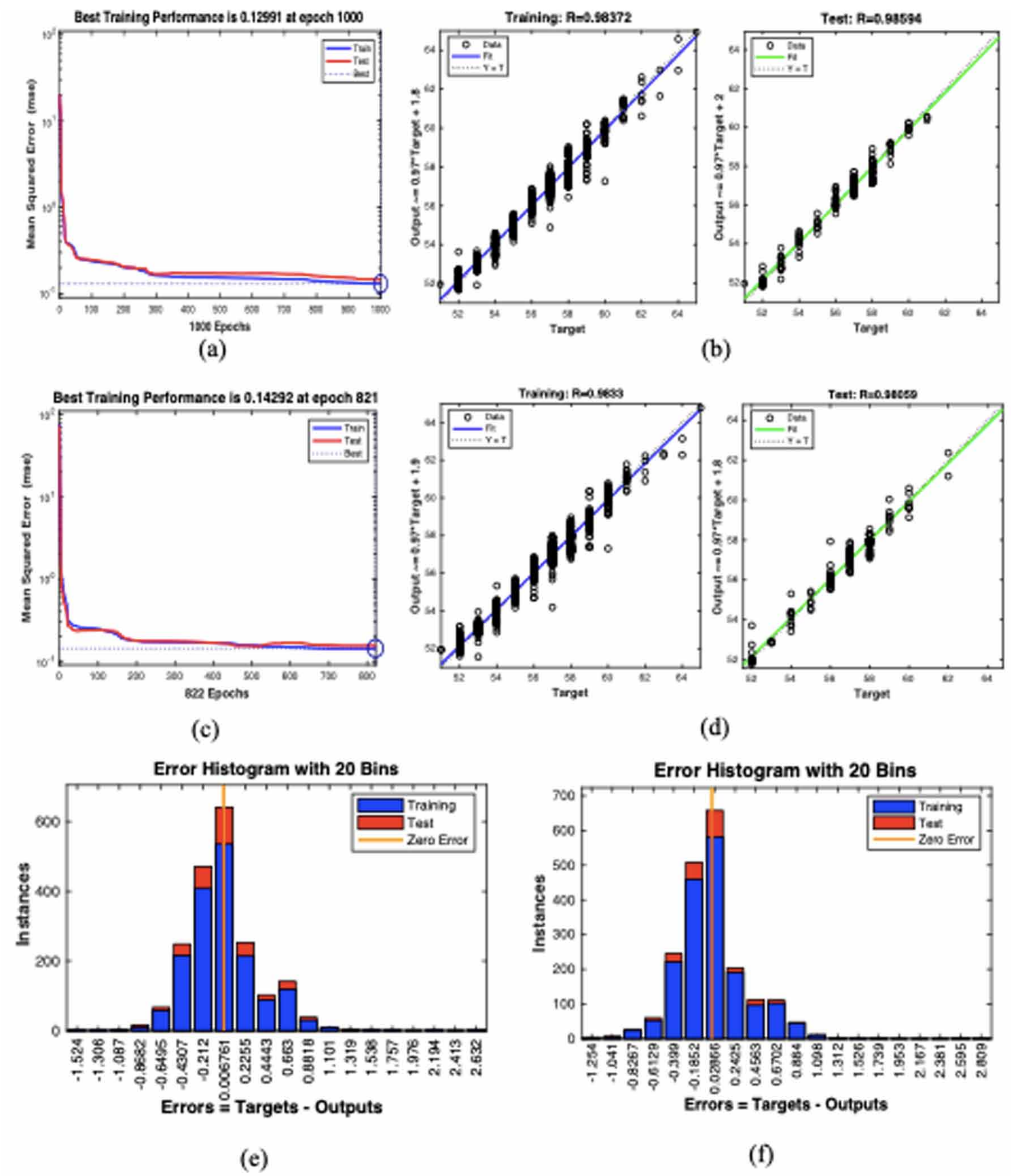
the regression plots for training (0.98) and testing (0.98), indicating high accuracy of prediction. These values were obtained for (70:15:15) data division. Similarly, Figure 13 (c) indicates that the best performance (MSE=0.14) was at epoch 821. The regression performance is as shown in the plot of Figure 13 (d), having R=0.98 for both training and testing. The values in Figure 13 (c) and Figure 13 (d) correspond to data division rule of (80:10:10). The error performance in the form of error histograms is as shown in Figure 13 (e) and Figure 13 (f) for 20 bins of training and testing datasets. The zero-error line almost coincides with zero, indicating small errors. The advantage of using the Bayesian Regularization training function is that its capability of generalizing well when compared to other training functions. Another important observation that was made is the poor performance of the Gradient Descent training function, which produced (MSE= 4.85 and 12.42), respectively for n=15 and data division rule of (70:15:15 and 80:10:10), respectively. After training the model, the prediction capability of the model was tested for the testing data.

The model was very accurate in predicting the soil moisture values based on the input data. The actual variation in the soil moisture data as measured by the soil moisture sensor is shown in Figure 14 (a), while the predicted values for 100 data points are as shown by Figure 14 (b). The results of the developed model for moisture prediction are tabulated in Table 5. Thus, the predictive model is capable of providing near precise prediction of soil moisture value considering the weather conditions along with the field parameters. This prediction would help the farmers to plan their irrigation water usage accordingly so that any inappropriate usage of water can be avoided.

## COMPARISON WITH SIMILAR IMLEMENTATIONS

After the successful design, deployment, and testing of the prototype, it was compared with the similar implementations which were used in addressing the issues arising out of irrigation problems. Important parameters considered for the comparison were based on the implementation costs, complexity, security, ease of user control, IoT devices used (microprocessor/controller/prototyping board), communication technology or protocol used and finally the IoT platform deployed for the

**Figure 13. Various plots corresponding to the model testing and evaluation of Bayesian Training function (a) Training performance (for 70:15:15) (b) Regression plots (for 70:15:15) (c) Training performance (for 80:10:10) (d) Regression plots (for 80:10:10) (e) Error histogram (for 70:15:15) and (f) Error histogram (for 80:10:10)**




monitoring and control of the field parameters. Table 6 gives a detailed comparison of various IoT implementation in agriculture. The comparison shows that most of the implementations rely heavily upon the use of IoT for monitoring and providing irrigation controls based on the field parameters importantly the moisture and other dependent parameters. While the other implementations develop machine learning models and use standard datasets to demonstrate the model's effectiveness in solving either classification or regression problems.

Figure 14. Variation in soil moisture (a) Measured and (b) Predicted

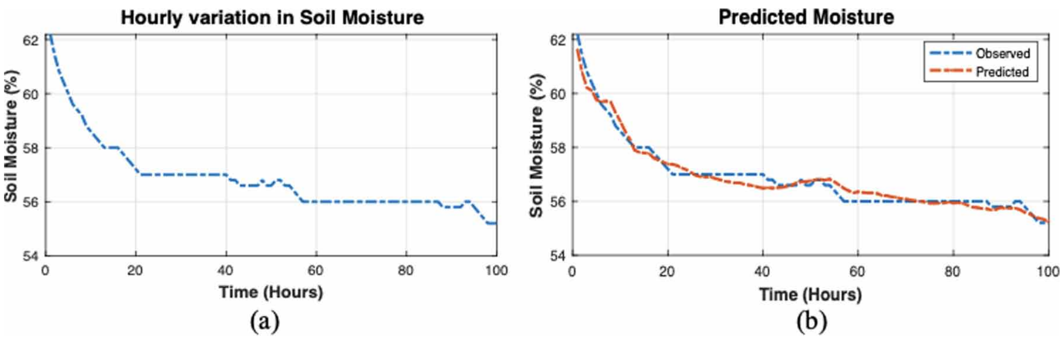



Table 5. Performance evaluation of the proposed model

| *Sl. No.* | *Training Function* | *No. of Hidden Neurons (n)* | *Data division rule:70:15:15* | | | | *Data division rule:80:10:10* | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | **R values** | | | **MSE** | **R values** | | | **MSE** |
| | | | **Training** | **Validation** | **Testing** | | **Training** | **Validation** | **Testing** | |
| 1. | Levenberg-Marquardt *(trainlm)* | 5 | 0.97 | 0.97 | 0.98 | 0.24 | 0.97 | 0.95 | 0.96 | 0.27 |
| | | 10 | 0.97 | 0.97 | 0.95 | 0.27 | 0.97 | 0.97 | 0.97 | 0.26 |
| | | 15 | 0.97 | 0.96 | 0.97 | 0.23 | 0.98 | 0.97 | 0.98 | 0.18 |
| 2. | Bayesian Regularization *(trainbr)* | 5 | 0.98 | - | 0.98 | 0.19 | 0.98 | - | 0.98 | 0.19 |
| | | 10 | 0.98 | - | 0.98 | 0.15 | 0.98 | - | 0.98 | 0.17 |
| | | **15** | **0.98** | **-** | **0.98** | **0.13** | **0.98** | **-** | **0.98** | **0.14** |
| 3. | Scaled Conjugate Gradient *(trainscg)* | 5 | 0.95 | 0.97 | 0.92 | 0.43 | 0.94 | 0.95 | 0.95 | 0.48 |
| | | 10 | 0.95 | 0.96 | 0.93 | 0.26 | 0.94 | 0.95 | 0.93 | 0.47 |
| | | 15 | 0.95 | 0.96 | 0.95 | 0.38 | 0.95 | 0.93 | 0.94 | 0.44 |
| 4. | *Gradient Descent (traingd)* | 5 | 0.39 | 0.40 | 0.41 | 4.87 | 0.05 | 0.15 | 0.02 | 6.43 |
| | | 10 | 0.47 | 0.32 | 0.52 | 18.14 | 0.29 | 0.35 | 0.19 | 20.96 |
| | | 15 | 0.55 | 0.50 | 0.55 | 4.85 | 0.60 | 0.61 | 0.51 | 12.42 |

The developed prototype is not only capable of collecting the field data, more importantly, the data is converted into real-time as well as the historical dataset by using some of the elementary transformations and processing such as time-scaling, outlier detection, missing value substitution, ensuring the dataset of high quality to be inputted to the machine learning algorithm. As no heavy processing was involved, the use of ESP 32 MCU is justified to keep a check on the cost factor. Also, the system's capability for providing irrigation control in the form of manual or automated is an added advantage.

## Prototype Cost Estimation and Analysis

The most important factor that any farmer would like to know is the cost factor involved in the deployment of the system. In countries like India, most of the farmers have very small landholdings (less than 5 acres), hence they will be reluctant to invest if the cost factor is not checked. The cost of the developed prototype was calculated in (U.S Dollars) based on the current conversion rates. The cost estimation does not include labor charges. Table 6 gives the details of the components with the

**Table 6. Cost estimation and analysis of developed prototype**

| SL. No. | Component | *Per Unit Cost (USD)* | *No. of Units* | *Amount (USD)* |
|---|---|---|---|---|
| 1. | ESP 32 DEV KIT V1 | 12.21 | 1 | 12.21 |
| 2. | DHT 22 (Temperature & Humidity) | 4.34 | 1 | 4.34 |
| 3. | BMP180 (Atmospheric Pressure) | 5.12 | 1 | 5.12 |
| 4. | YL-83+LM393 (Rain Sensor) | 2.95 | 1 | 2.95 |
| 5. | FC-28+LM393 (Soil Moisture) | 3.13 | 1 | 3.13 |
| 6. | Single Channel Relay | 2.11 | 1 | 2.11 |
| 7. | Protective Case for the prototype | 5.00 | 1 | 5.00 |
| 8. | AC to DC converter power supply module (12V) | 6.20 | 1 | 6.20 |
| 9. | 12 V solenoid Valve (1/2”) | 16.00 | 1 | 16.00 |
| 10. | Miscellaneous (Connecting wires, jumper wires, etc.) | 3.00 | - | 3.00 |
| **TOTAL** | | | | **60.06$** |

corresponding costs. The rates of the components were calculated at the time of purchase, which may vary now due to variation in the market at the global level. The final cost of the prototype (excluding labor charges) turns out to be around 60$, which shows a considerable reduction in the cost when compared to the similar implementation by Abba S et al. (2019) in which an irrigation system was developed using Arduino Uno costing around 80$ approximately.

## CONCLUSION AND FUTURE DIRECTIONS

The main highlight of the developed system is its quick, easy and low-cost deployment nature aiming to solve irrigation issues for farmers globally, particularly belonging to the developing countries. The paper describes how data from the sensors in the agricultural field can be collected and analyzed to provide the farmers with decision making capability. Low- cost nature of the developed system will attract a greater number of low to medium holding farmers to give it a try, who are otherwise not in a position to afford systems having high initial deployment costs. Easy to use *Blynk* Android App with drag and drop functionality was found to be very effective in not only providing irrigation automation but also visualization. The developed FFNN based model justifies the use of agriculture sensor parameters as valid inputs to the system. The manual and automated irrigation controls were added to address the different levels of understanding of the farmers. This research led to some interesting findings while developing the system. The first inference that can be drawn is that ESP32 MCU has got the tremendous potential of being one of the strong contenders for any IoT based application (built-in Wi-Fi and dual-mode Bluetooth, high security, low-cost, low-power, dual-core 32-bit processor more GPIO pins). The second inference that can be drawn is that the use of FC-28+LM393 based soil moisture sensor (resistive) started corroding and turning green at some places after few testings’ in the field, thus, it is recommended to use capacitive soil moisture sensor which is immune to corrosion. Overall, a positive response was obtained from the end-user in terms of ease of usage and reduction in labor involvement and resource conservation. In the developed system, the predictive model was deployed in cloud sever, adding some latency to the message exchanges. Thus, as a future enhancement to the system, the predictive model would be deployed at the edge node rather than the cloud so that latency can be considerably reduced and also power consumption could be reduced as there would be no requirement of the Internet for running the model.

## ACKNOWLEDGMENT

The authors would like to extend gratitude towards a farmer (holds M. Tech degree in VLSI and has an agriculture farm in Tikota, Vijayapur, Karnataka, India) Mr. Bapuray D. Yammenavvar, helping us to understand important agricultural requirements and for providing us space in his farm for testing our prototype and providing us with valuable feedback.

## REFERENCES

Abba, S., Namkusong, J. W., Lee, J. A., & Crespo, M. L. (2019). Design and performance evaluation of a low-cost autonomous sensor interface for a smart iot-based irrigation monitoring and control system. *Sensors (Switzerland)*, *19*(17), 3643. Advance online publication. doi:10.3390/s19173643 PMID:31438597

Abdelwahab, S., Hamdaoui, B., Guizani, M., & Znati, T. (2016). Cloud of Things for Sensing-As-A-Service: Architecture, Algorithms, and Use Case. *IEEE Internet of Things Journal*, *3*(6), 1099–1112. doi:10.1109/JIOT.2016.2557459

Ahmed, N., De, D., & Hussain, I. (2018). Internet of Things (IoT) for Smart Precision Agriculture and Farming in Rural Areas. *IEEE Internet of Things Journal*, *5*(6), 4890–4899. doi:10.1109/JIOT.2018.2879579

Alahi, M. E. E., Pereira-Ishak, N., Mukhopadhyay, S. C., & Burkitt, L. (2018). An Internet-of-Things Enabled Smart Sensing System for Nitrate Monitoring. *IEEE Internet of Things Journal*, *5*(6), 4409–4417. doi:10.1109/JIOT.2018.2809669

Aleotti, J., Amoretti, M., Nicoli, A., & Caselli, S. (2018). A Smart Precision-Agriculture Platform for Linear Irrigation Systems. In *2018 26th International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2018* (pp. 401–406). Institute of Electrical and Electronics Engineers Inc. doi:10.23919/SOFTCOM.2018.8555841

Aryal, J. P., Sapkota, T. B., & Khurana, R. (2019). *Environment, Development and Sustainability*. Advance online publication. doi:10.1007/s10668-019-004144

Blynk Documentation. (n.d.). Retrieved May 6, 2019, from http://BlynkDocumentation_sensors

Chen, W. L., Lin, Y. B., Lin, Y. W., Chen, R., Liao, J. K., Ng, F. L., & Yen, T. H. (2019). AgriTalk: IoT for precision soil farming of turmeric cultivation. *IEEE Internet of Things Journal*, *6*(3), 5209–5223. doi:10.1109/JIOT.2019.2899128

Dharwadkar, N. V., & Harale, V. R. (2019). Automated greenhouse system for tomato crops using deep learning. *International Journal of Sustainable Agricultural Management and Informatics*, *5*(2/3), 131. doi:10.1504/IJSAMI.2019.101674

Dobrescu, R., Merezeanu, D., & Mocanu, S. (2019). Context-aware control and monitoring system with IoT and cloud support. *Computers and Electronics in Agriculture*, *160*, 91–99. doi:10.1016/j.compag.2019.03.005

Elijah, O., Rahman, T. A., Orikumhi, I., Leow, C. Y., & Hindia, M. N. (2018). An Overview of Internet of Things (IoT) and Data Analytics in Agriculture: Benefits and Challenges. *IEEE Internet of Things Journal*, *5*(5), 3758–3773. doi:10.1109/JIOT.2018.2844296

Goldstein, A., Fink, L., Meitin, A., Bohadana, S., Lutenberg, O., & Ravid, G. (2018). Applying machine learning on sensor data for irrigation recommendations: Revealing the agronomist's tacit knowledge. *Precision Agriculture*, *19*(3), 421–444. doi:10.1007/s11119-017-9527-4

Gomez, J., Fernandez, A., & Zúñiga Sánchez, M. (2019). Monitoring of small crops for the measurement of environmental factors through the internet of things (IoT). In *Communications in Computer and Information Science* (Vol. 895, pp. 16–28). Springer Verlag., doi:10.1007/978-3-030-05532-5_2

Guillermo, J. C., García-Cedeño, A., Rivas-Lalaleo, D., Huerta, M., & Clotet, R. (2019). IoT Architecture Based on Wireless Sensor Network Applied to Agricultural Monitoring: A Case of Study of Cacao Crops in Ecuador. In Advances in Intelligent Systems and Computing (Vol. 893, pp. 42–57). Springer Verlag. doi:10.1007/978-3-030-04447-3_3

Jochinke, D. C., Noonon, B. J., Wachsmann, N. G., & Norton, R. M. (2007). The adoption of precision agriculture in an Australian broadacre cropping system-Challenges and opportunities. *Field Crops Research*, *104*(1–3), 68–76. doi:10.1016/j.fcr.2007.05.016

Joshi, V. B., & Goudar, R. H. (2019). IoT-based automated solution to irrigation: An approach to control electric motors through Android phones. In *Advances in Intelligent Systems and Computing* (Vol. 707, pp. 323–330). Springer Verlag., doi:10.1007/978-981-10-8639-7_33

Kale, A. P., & Sonavane, S. P. (2019). IoT based Smart Farming: Feature subset selection for optimized high-dimensional data using improved GA based approach for ELM. *Computers and Electronics in Agriculture*, *161*, 225–232. doi:10.1016/j.compag.2018.04.027

Klein, L. J., Hamann, H. F., Hinds, N., Guha, S., Sanchez, L., Sams, B., & Dokoozlian, N. (2018). Closed-Loop Controlled Precision Irrigation Sensor Network. *IEEE Internet of Things Journal*, *5*(6), 4580–4588. doi:10.1109/JIOT.2018.2865527

Koduru, S., Padala, V. G. D. P. R., & Padala, P. 2019. Smart Irrigation System Using Cloud and Internet of Things. In *Proceedings of 2nd International Conference on Communication, Computing and Networking*. Springer. doi:10.1007/978-981-13-1217-5_20

A. Kumar, & S. Mozar (Eds.). (2019). Design of a Smart Water-Saving Irrigation System for Agriculture Based on a Wireless Sensor Network for Better Crop Yield. In *International Conference on Communications and Cyber-Physical Engineering 2018*. Springer. doi:10.1007/978-981-13-0212-1_11

Lata, S. (2019). Sources of Irrigation: A Theoretical Framework. In *Irrigation Water Management for Agricultural Development in Uttar Pradesh, India.* Springer. doi:10.1007/978-3-030-00952-6_3

Math, R. K. M., & Dharwadkar, N. V. (2019). IoT Based low-cost weather station and monitoring system for precision agriculture in India. In *Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics, and Cloud), I-SMAC 2018* (pp. 81–86). Institute of Electrical and Electronics Engineers Inc. doi:10.1109/I-SMAC.2018.8653749

Mittal, A., Sarangi, S., Ramanath, S., Bhatt, P. V., Sharma, R., & Srinivasu, P. (2019). IoT-Based Precision Monitoring of Horticultural Crops-A Case-Study on Cabbage and Capsicum. In *GHTC 2018 - IEEE Global Humanitarian Technology Conference, Proceedings*. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/GHTC.2018.8601908

Muangprathub, J., Boonnam, N., Kajornkasirat, S., Lekbangpong, N., Wanichsombat, A., & Nillaor, P. (2019). IoT and agriculture data analysis for smart farm. *Computers and Electronics in Agriculture*, *156*, 467–474. doi:10.1016/j.compag.2018.12.011

Nawandar, N. K., & Satpute, V. R. (2019). IoT based low cost and intelligent module for smart irrigation system. *Computers and Electronics in Agriculture*, *162*, 979–990. doi:10.1016/j.compag.2019.05.027

Pavón-Pulido, N., López-Riquelme, J. A., Torres, R., Morais, R., & Pastor, J. A. (2017). New trends in precision agriculture: A novel cloud-based system for enabling data storage and agricultural task planning and automation. *Precision Agriculture*, *18*(6), 1038–1068. doi:10.1007/s11119-017-9532-7

Popović, T., Latinović, N., Pešić, A., Zečević, Ž., Krstajić, B., & Djukanović, S. (2017). Architecting an IoT-enabled platform for precision agriculture and ecological monitoring: A case study. *Computers and Electronics in Agriculture*, *140*, 255–265. doi:10.1016/j.compag.2017.06.008

Raut, R., Varma, H., Mulla, C., & Pawar, V. R. (2018). Soil Monitoring, Fertigation, and Irrigation System Using IoT for Agricultural Application. In *Lecture Notes in Networks and Systems* (Vol. 19, pp. 67–73). Springer., doi:10.1007/978-981-10-5523-2_7

Tripathy, S. (2019). Rooting for Sustainable Agriculture and Food Security Through Improved Regulatory Governance in India. In K. C. Liu & U. Racherla (Eds.), *Innovation, Economic Development, and Intellectual Property in India and China. ARCIALA Series on Intellectual Assets and Law in Asia*. Springer. doi:10.1007/978-981-13-8102-7_17

Tzounis, A., Katsoulas, N., Bartzanas, T., & Kittas, C. (2017, December 1). Internet of Things in agriculture, recent advances and future challenges. In *Biosystems Engineering*. Academic Press. doi: 10.1016/j.biosystemseng.2017.09.007

Vidyashree, L., & Suresha, B. M. (2019). Methodology to Secure Agricultural Data in IoT. In A. Abraham, P. Dutta, J. Mandal, A. Bhattacharya, & S. Dutta (Eds.), *Emerging Technologies in Data Mining and Information Security. Advances in Intelligent Systems and Computing* (Vol. 814). Springer. doi:10.1007/978-981-13-1501-5_11

*Rajinder Kumar M. Math has obtained his Bachelor of Engineering degree from Karnataka University Dharwad, MTech in Digital Communication from Visvesvaraya Technological University and is a research scholar at Visvesvaraya Technological University, Belagavi, Karnataka, India. Currently he holds the position of Assistant Professor in the department of Electronics and Communication Engineering of B.L.D.E. Association's Vachana Pitamaha Dr. P.G. Halakatti College of Engineering and Technology, Vijayapur-Karnataka, India. His areas of interests include Wireless Sensor Networks (WSNs), Internet of Things (IoT) and Machine Learning related to Precision Agriculture Systems,*

*Nagaraj V. Dharwadkar obtained his B.E. in Computer Science and Engineering in 2000, from Karnataka University Dharwad, M.Tech. in Computer Science and Engineering in the year 2006 from VTU, Belgium and Ph.D. in Computer Science and Engineering in 2014 from National Institute of Technology, Warangal . He is Professor and Head of the Computer Science and Engineering department at Rajarambapu Institute of Technology, Islampur. He had 15 years of Teaching Experience at Professional Institutes across India and published 40 papers in various International Journals and Conferences. His area of research interest is Multimedia Security, Image Processing, Big Data Analytics and Machine Learning.*