Research Article

S. Premkumar* and AN. Sigappi

IoT-enabled edge computing model for smart irrigation system

https://doi.org/10.1515/jisys-2022-0046
received January 10, 2022; accepted March 16, 2022

Abstract: Precision agriculture is a breakthrough in digital farming technology, which facilitates the application of precise and exact amount of input level of water and fertilizer to the crop at the required time for increasing the yield. Since agriculture relies on direct rainfall than irrigation and the prediction of rainfall date is easily available from web source, the integration of rainfall prediction with precision agriculture helps to regulate the water consumption in farms. In this work, an edge computing model is developed for predicting soil moisture in real time and managing the water usage in accordance with rain prediction. A soil moisture prediction hybrid algorithm (SMPHA) has been developed that revolves around the decision-making techniques with live environmental parameters including weather parameters for the prediction of soil moisture through the impact of precipitation. Numerous algorithms with the combination of regression + clustering are estimated, and it is inferred that XGBoost + k-means outperforms other algorithmic combinations that is deployed in edge model. This model is used as an intermediary between the end IoT devices and cloud that results in the saving of computationally intensive processing performed on cloud servers. The servers located on a local edge network perform the developed algorithmic computations. Avoiding transmission over the cloud results in significant latency, response time, and computation power savings and therefore increases the efficiency of data transfer. The proposed edge computing model is implemented in Raspberry Pi as an edge, Heroku as cloud, and edge nodes as the combination of Pi with actuators and sensors. The monitored data from Pi are stored in MongoDB webserver that is controlled by Web dashboard. Finally, the developed model is implemented in cloud and edge where the edge server implementation performs better in terms of latency, bandwidth, throughput, response time, and CPU memory usage.

Keywords: smart irrigation, edge-based irrigation, edge computing, precision agriculture, soil moisture prediction, irrigation management system, IoT, offloading mechanism

1 Introduction

It is evident that agriculture always has a specialized role in the anthrophonic evolution and has been serving as an important economic factor for the growth of a country [1]. Around 58% of the population depend on agriculture as the chief source of livelihood in India. The quality and productivity of agricultural products have declined over these years as several factors have influenced the crop productivity both directly and indirectly. Some major factors that affect the crop production are climatic changes, global warming, and water scarcity [2]. The agricultural land's productivity is affected by the direct and indirect

*** Corresponding author: S. Premkumar,** Department of Computer Science and Engineering, Faculty of Engineering and Technology, Annamalai University, Chidambaram - 608002, Tamilnadu, India, e-mail: premambal@gmail.com
AN. Sigappi: Department of Computer Science and Engineering, Faculty of Engineering and Technology, Annamalai University, Chidambaram - 608002, Tamilnadu, India, e-mail: an.sigappi@gmail.com

changes in climate [3,4]. The crop growth has been already affected by the changes in climate incurred by global warming. The nutrition quality of soil, ground water level, sea, and ocean are affected by the modifications in average temperature, rainfall, and extreme weather conditions such as hail storms, dust storms, heatwaves, etc. due to global warming [5,6]. Degradation of soil is primarily created by various methods including 93.7% by water erosion, 9.5% by wind erosion, 5.9% by salinity and alkalinity, etc. Further changes in climate would influence adversely the crop production [7]. Since water is an indispensable requirement for plants and cultivation, the high level of soil is eroded and thereby the fertility is also declined. Due to the ever-changing climate, water scarcity has become a huge problem. Drought-like conditions is already formed in several areas and thereby the present and conventional farming practices are not suitable. New and unique environment preserving techniques are the need of the hour [8].

The conventional approaches in agriculture are enhanced by the advent of several advancements in technology [9]. These new improved methodologies ensure optimized utilization of resources, accurate forecast of water needs and environmental parameters, reduction of human intervention, etc. [10]. Consequently, the outcomes of crops in terms of yield and quality are higher with cost-effective methods. One such booming technology is the Internet of things (IoT) [11].

IoT is the collection of components embedded in the sensor for measuring and transferring data via network devices as sensed from pumps and tractors to weather stations. Primarily, IoT deals with the transmission and reception of data related to farms through devices using the Internet for prediction and providing decisions to the farmers. IoT-based methodologies has brought a changeover in agricultural patterns and farming approaches [12]. IoT devices can gather information about soil moisture, chemical properties, dam levels, livestock health, and weather details in real time. The information acquired from IoT devices facilitates the farmers in tracking farms periodically. Farmers can save time and money by responding faster to farm conditions. Cloud computing models integrated with on-field agricultural sensors need to be incorporated for tackling the issue of processing huge voluminous data.

One of the major challenges of IoT is the processing of huge datasets in a sequential way. Some of the key factors that need to be focused on this process are as follows: information about the type and nature of data, the way of acquiring the data, etc. The preliminary stage comprises acquiring the data and ingesting the data to the system. Substantial cognizance of data are achieved as the data pass through all the gateways where it is cleansed and transformed before entering into the system. In the near future, dynamic prediction of soil moisture and precipitation techniques are to be developed for smart irrigation systems. Therefore, a system is developed for efficient and optimal utilization of fresh water in irrigation along with drip irrigation system. It aids in finding which one of the plants fails to get sufficient water. When the water supply is provided the next day, this delay should not disturb the system. It becomes important for the farmers to understand the optimal usage of water and fertilizers to bring out sustenance in the agricultural industry. Therefore, processing must be done for analyzing the data, so that patterns can be analyzed and planning can be done for the long term, accordingly. Hence, it gives a broad vision in deciding where the processing is to be done exactly. Therefore, it is obvious that not all data are crucial, and it provides a clear view of which data need to be stored, discarded, and retained for both long-term and short-term purposes. Thus, all these challenging issues require to be addressed and that is where storage technologies are actually highlighted. The poor quality of Internet access in developing nations makes the implementation quite challenging. An applicable solution to solve this problem is through edge computing where the essential data could be offloaded from the cloud over the edge of the cloud, and this is the exact point where the approach of smart sensing with edge computing gets in.

With the purpose of broadening the potential of edge computing and using it in the agriculture domain, a novel approach using machine learning (ML) methods is proposed for analyzing the data acquired by the IoT devices deployed at the farm. Here, the data acquired from IoT components undergo preprocessing and ML models on the edge nodes to analyze and assess the appropriate results for providing the best instructions for controlling the actuators (e.g., light, pumps at different locations) in the farms.

This article presents an automated system, as shown in Figure 1, to predict the soil moisture using the field information acquired from the self-designed sensor node deployed at the field and the forecast information of weather via Internet. A unique algorithm has been developed that revolves around the

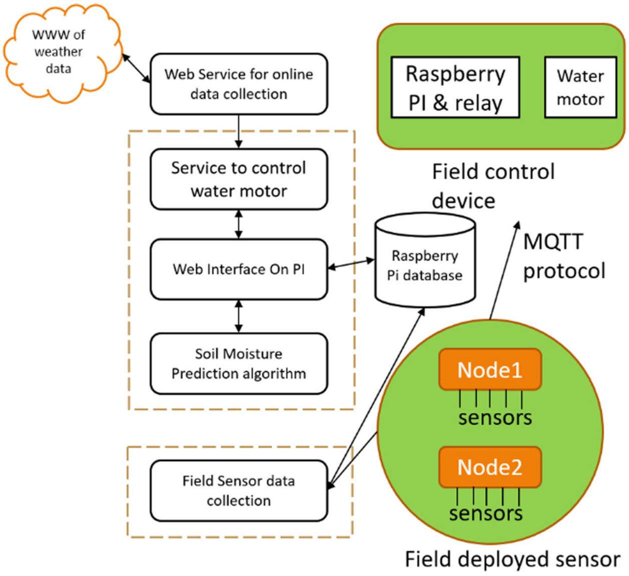


Figure 1: Architecture of the proposed system.

machine learning techniques for the prediction of soil moisture. Here, many algorithms with the combination of regression + clustering was estimated, and it is inferred that XGBoost + k-means outperforms other algorithmic combinations, and therefore, it is deployed for the prediction of soil moisture in the proposed work. The proposed algorithm makes effective irrigation decisions with optimized usage of water in a more accurate and reliable manner. The effective decision-making refers to the process of predicting the rainfall, thereby reducing the water usage in advance by the proposed algorithm in accordance with the predicted rainy days. Through this automatic decision-making, over watering is avoided by saving the soil. The server-side software is developed with node-side connectivity using the information for visualization and decision support features. This proposed algorithm is implemented in edge to prove the efficiency of the edge server handling the automated system better than the cloud control. The performance of the decentralized edge-based architecture has been evaluated for downloading the hybrid algorithm from cloud in real time execution. The performance can be enhanced by adopting edge computing architecture and measured with the help of network parameters like latency, bandwidth, and response time. Edge computing capacity is also estimated using the CPU processor and memory consumption while executing the proposed algorithm with irrigation scheduling.

2 Related work

In ref. [13], a smart irrigation system not affected by communication disconnection and delay is developed using edge nodes deployed at the farms. Environmental parameters have an intricate impact on the plant growth. It becomes necessary for evaluating multiple AI models simultaneously in an actual cultivation environment for comparing AI models under the same conditions. Due to the working of existing irrigation systems on the cloud, communication is instable in the concurrent evaluation of AI models. However, the instability does not induce an edge node in its performance.

The factors such as type of plant, soil, climate, humidity, temperature, and soil moisture need to be considered for the irrigation system packed with potential smart decisions. The nature and type of plant, soil, and climate are queried by ontology (branch of metaphysics dealing with the nature of being), whereas other factors such as temperature, humidity, and soil moisture are sensed by the sensor network. The trained ML model predicts the watering decisions based on ontology and other factors as mentioned earlier.

Smart irrigation has three modules: (i) sensor network modules that sense the parameters impacting the water requirement by using sensors DHT22, light sensor BH1750, and HL-69 hygrometer for sensing the temperature, soil moisture, light, and humidity in air. (ii) Edge and IoT server's module to send and receive data through HTTP requests. (iii) Training module in which KNN is applied on the sample dataset for training and decision-making regarding the water needs. Based on the input values, the trained model categorizes the input into five possible classes: highly not needed, not needed, average, needed, and highly needed [14].

A decentralized smart irrigation approach is proposed for strawberry greenhouses in contrast to conventional cloud-based solutions for keeping the agricultural data at the edge of the network. A full-scale smart irrigation system in an actual strawberry greenhouse environment is developed after a small-scale smart irrigation networking prototype system and a reference architecture targeting edge data distribution for strawberry greenhouse applications are framed. A three-step industrial approach is formed for designing, implementing, and validating a solution for smart strawberry irrigation in greenhouses and keeping the corresponding data at the edge of the network at the same time: (i) A small-scale smart irrigation prototype solution with off-the-shelf hardware and software equipment is tested and evaluated on various types of plants for gaining useful insights for deployments on a large scale. (ii) A reference network architecture is designed for targeting smart irrigation and edge data distribution specifically for strawberry greenhouses. (iii) A large-scale system in an actual strawberry greenhouse environment is developed in Greece, incorporating the proposed reference architecture [15].

Edge computing is proposed for addressing the issues by taking advantage of computing resources in the edge of the network. The issues such as an edge mobile device make it easier to achieve low end-to-end latency, high bandwidth, and low jitter to services located on the edge network. An edge can enforce the privacy policies of its owner prior to the release of the data to the cloud through edge analytics. If a cloud service becomes unavailable due to network failure, cloud failure, or a denial-of-service attack, a fallback service on a nearby edge can temporarily mask the failure. Cloud services, partial analysis, and control functions are extended to the edge nodes from the cloud data center. Edge nodes facilitate the timely monitoring of sensors in smart farming by the reduced latency and enhanced data transmission. Due to these factors, edge computing is applied through farming [16,17]. A three-tier open-source software platform we proposed by authors, and the platform enhanced the precision agriculture by introducing edge computing and fog computing. An network functions virtualization (NFV)-based approach is deployed for performing the local operational decisions at the edge level for mitigating the influence of network failures while using cloud data centers [18]. For control processing in smart farming, a platform enabling cost-effective sensor/actuator network based on IoT, utilizes edge computing [19,20].

The authors in ref. [21] predicted the soil moisture using a mathematical model that measures the values given by a sensor matrix on the ground. Due to the huge interval in measurements (10 minutes), the model presented estimated error by more than 10%. This methodology has incorporated the online approach by making the sensors to send data every minute to edge devices without time-based interruption. The authors of ref. [22] applied a combinative approach of using field sensor network's data along with weather forecast station's data for the management of optimality in water conditions for the enhanced growth of grapes. The generated data are forwarded to a web server, which displays graphics without statistical analysis of such data. The analysis must be performed a posteriori by the user.

The watering mechanism for a plant via IoT methodology is devised by the proposed smart irrigation model without acquiring any pre-processed data. A prototype application is developed, which gets adapted to the parameters needed in irrigation after a couple of human-made irrigations. With the usage of various ML algorithms, several tests are devised for manual and automated irrigations for the performance evaluation. After the evaluation using four different ML algorithms such as logistic regression (LR), K-nearest neighbors (KNN), Gaussian naive Bayes (GNB), and gradient boosting regression trees (GBRT), it is found that GBRT outperforms other algorithms. To analyze the overall performance, a test bed for the sensor edge, mobile client, and the decision service on the cloud is established. Two different indoor species are selected as test items for the prototype, namely, Peace lily and Sardinia. The outcomes were quite good, and it is inferred that the prototype has learned the patterns of irrigation and making decisions automatically with a high rate of accuracy [23].

The authors from ref. [24] adopted the deep learning methodology for detecting the type and the category of the plant using an automated plant irrigation system. The water necessity of the plant is determined using the recognition of predefined set of plant images and data set acquired from farm. It utilizes the database for fetching the irrigation information after the recognition process is completed. Modeling the training processes are time consuming as voluminous set of images needs to be stored. The authors in refs [25,26] incorporated ML methods in the irrigation decision support model using a pre-processed irrigation data set. A model is developed for learning the irrigation needs of any plants progressively rather than using a readily available dataset. Several ML algorithms are evaluated with their precision for concluding the irrigation decisions. Manual irrigations are performed two times before making precise decisions. Due to the dynamicity in model, data processing is done progressively, and it can be applied to several plants having varying irrigation conditions. There is a need for the learning model that can be trained by itself using a comparatively lighter learning process using environmental parameters that do not need larger storage in the system but need higher computation. From the aforementioned survey for making a precise decision with instant computation locally, edge computing needs to be integrated into the irrigation system. This article is directed towards presenting a platform that implies IoTs and edge computing in monitoring soil moisture via sensors, data communication between sensors and edge devices, and an Analytics-as-a-Service cloud. It analyzes the collected data in the form of a density map of soil moisture for denoting the areas in need of greater or lesser frequency of irrigation. Here, density map does not refer to the geographical point data by satellite mapping, and it actually denotes the point of dry area and watery area through soil moisture detection point. This point is averaged among areas of irrigation to be done and the irrigation process is controlled with prediction of rainfall using the proposed system.

3 The proposed system

The proposed learning model for irrigation is implemented in a prototype IoT system that has four components: (i) Edge node layer – This layer consists of sensors, actuator, and two microcontrollers. In this layer, edge node acquires the sensor data from the surroundings and controls the actuator for actuating water pumps to start irrigation. (ii) Edge server layer – This layer consists of Raspberry Pi that act as edge server and capable of multitask processing. Here, edge server controls the edge nodes for sending signal and receiving data at regular interval of time. It is also connected to the cloud server for receiving developed and trained machine learning model to be deployed and make irrigation decision for controlling edge nodes. (iii) Edge service layer – This layer is deployed in the edge server and it is responsible for controlling the whole system through a developed web dashboard. The dashboard has live feed data, control of edge nodes, and cloud services access. This service layer also has the control access of the proposed machine learning model. (iv) Cloud server layer – This layer composed of cloud services and cloud storage where its role is to train the machine learning model and store the data in database. It sends the trained proposed model to the edge server for decision-making regarding irrigation scheduling. The comprehensive interconnections in the system are shown in Figure 2. The proposed IoT-based smart irrigation system includes five major components: field deployed module, Web-based interface, Web API weather input, soil moisture prediction mechanism, and edge communication model.

3.1 Field deployed module

In the field requirements, a wireless sensor network of the sensor nodes needs to be deployed as shown in Figure 3. Here, field data collection device accommodates four different sensors: Capacitive Soil Moisture Sensor V2.0, DS18B20 Water Proof Temperature Sensor Probe for soil temperature, ultraviolet (UV) Light Radiation, DHT11 – Temperature and Humidity Sensor Module, and GYML8511 Analog Output Ultra-Violet

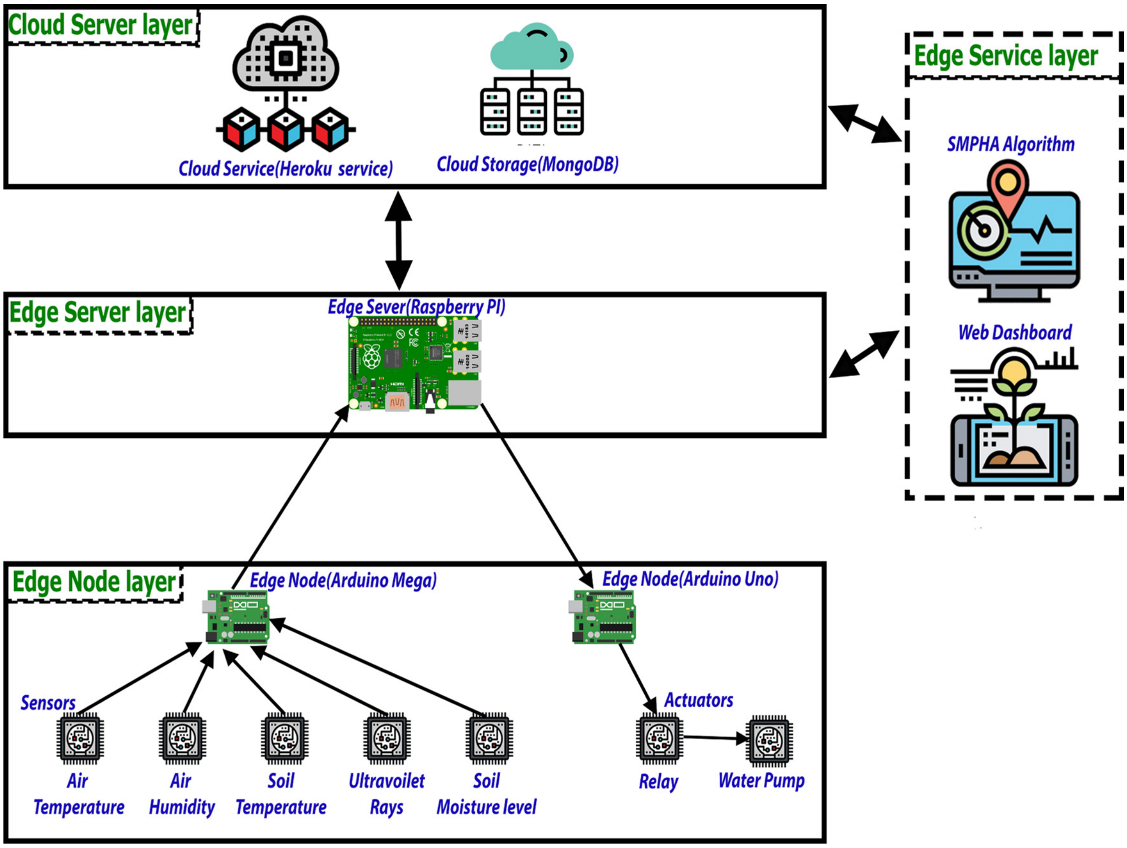


Figure 2: Components of the proposed system.

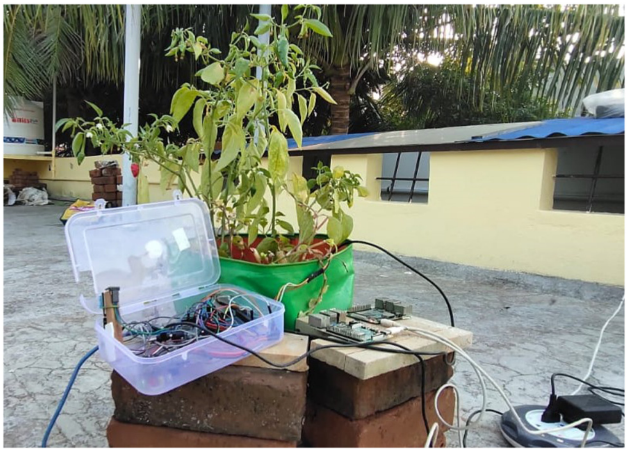


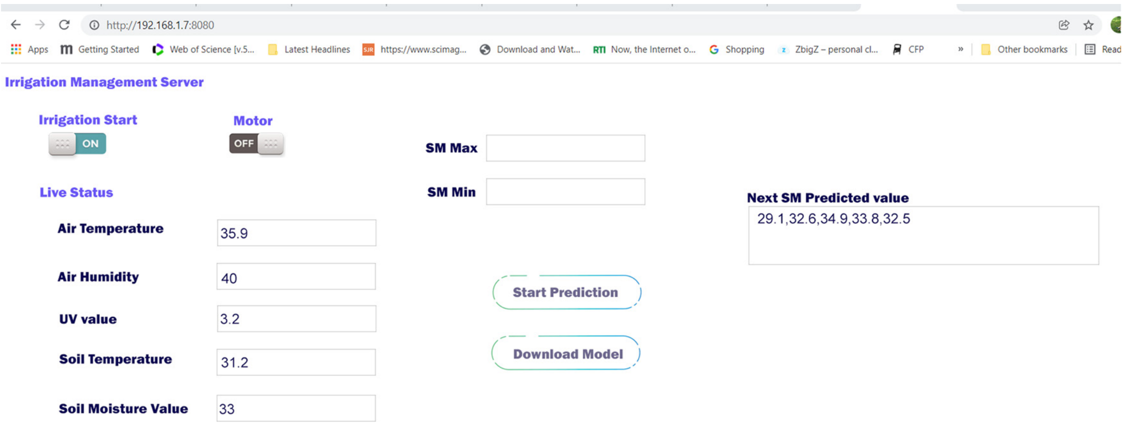
Figure 3: Real-time prototype of the proposed edge model.

Light Sensor Module. An Arduino Mega connected to Raspberry Pi 4 Model-B read, the output of these sensors where the program is developed in Python for the Pi model to fetch the hourly data from sensors and store the data in MongoDB [27] database. It is then synchronized with the server database using the developed web service. A Wi-Fi-enabled Arduino controls the water pump connected to a relay switch.

For the real time monitoring, a trigger is made for controlling the web service from the responsive web-based interface. The irrigation decisions are checked periodically by the proposed model performed in the server. The water pump is actuated, and irrigation process is started only if the server makes any irrigation decision. A wireless sensor network (WSN) [28] scenario with ZigBee [29] technology can be implemented for a large farming area in which several sensor nodes can be affixed in the specified area and every sensor node possesses sensors similar to a standalone device. Then, the Arduino Mega reads the sensor output connected to ZigBee for transferring data to Gateway Node for aggregating the received data and storing it in MongoDB locally and also for transferring the data via web service to the edge server.

3.2 Web-based interface

The proposed framework consists of a web-based application to allow farmers visualize the growing data and interacting with the garden in real time. In addition, users can also be able to examine and analyze the historical growing data, if needed, through functionalities such as irrigation control, motor control prediction model deployment, and manual data entry implemented in this web application. Here, Node.js was chosen for developing the web application [30,31], while MongoDB [27] was utilized as the database system. Data stored in the database, which is deployed in the cloud, will be used for further data analysis in the future. The web application's functions are designed following a software design pattern called model-view-controller (MVC) as shown in Figure 4. In the frontend, ChartJS is used to represent data through dynamic charts. The web application is also used as an interface to manage all the physical devices/actuators in the garden. To deploy the web-server to the cloud, a cloud platform as a service (PaaS), namely, Heroku, had been utilized. Heroku is a cloud platform that provides platform as a service (PaaS), facilitates the creation of applications and deploying these online rapidly [32,33]. It also enhances scalability and functionality by integrating several add-on services. The field data are sent to the server by Raspberry Pi using this web service. This web service manages the network outage/fluctuation during data synchronization from the field device to the server by taking the help of flag settings at the database level. The interface facilitates the scheduling of irrigation along with visualizing real time sensors and predicted soil moisture for upcoming days and precipitation information. By using the denoted threshold value of soil moisture suggested by agronomists, the irrigation can be scheduled by the user. The system maintains the threshold value depending on the predicted pattern of soil moisture and precipitation information. The process of irrigation is initiated automatically and stopped after the specified threshold value generated from the proposed algorithm of soil moisture when it is reached.



Irrigation Management Server

Irrigation Start: ON

Motor: OFF

SM Max

SM Min

Live Status

Air Temperature: 35.9

Air Humidity: 40

UV value: 3.2

Soil Temperature: 31.2

Soil Moisture Value: 33

Next SM Predicted value: 29.1,32.6,34.9,33.8,32.5

Figure 4: Web interface for the irrigation system.

3.3 Web API weather input

The weather prediction data are collected by a web service developed in Python. The forecast data such as humidity, temperature, ultra violet index, precipitation, and cloudiness of web forecasting portals like Open Weather API are aggregated by the developed web service [34]. These portals provide the forecasted information in HTML, XML, or JSON format. The predicted data with JSON format are read by the developed web and stored in database at the edge server, which is concerned in the prediction algorithm. Also, these data are utilized as testing dataset in the ML model for predicting the soil moisture.

3.4 Soil moisture prediction mechanism

An algorithm for predicting the soil moisture based on data derived from field sensors and weather forecasting using the combination of supervised and unsupervised machine learning techniques has been developed underpinned by regression algorithms and k-means clustering for estimating the difference/change in soil moisture owing to weather conditions. Many regression algorithms are compared against each other and infusing each of them with k-means to check the preciseness in mean square error (MSE), R2, accuracy and mean absolute percentage error (MAPE) for prediction of soil moisture of upcoming days with the help of sensor data and weather forecasting days. The information about soil moisture for the upcoming days and suggestions for irrigation in accordance with the prescribed levels of soil moisture and predicted precipitation values, thereby saving energy and water, is presented by the algorithm. The information generated from the device and the predicted values from the algorithm soil moisture prediction hybrid algorithm (SMPHA) are stored in the server.

3.5 Edge communication model

The communication protocols in the proposed framework are flexible and transparent in nature for accepting both wired and wireless methodologies. For the maximum utilization of potentiality in edge computing components, the communication among various components in the edge-IoT system requires intense probing by using the versatility among the devices in network edges. For transferring the data gathered from pivot sensors, a communication technology such as Zigbee [35] is needed for the irrigation systems. Therefore, the communication component in the proposed work is classified into three main areas as shown in Figure 5. The Message Query Telemetry Transport (MQTT) protocol is used for the

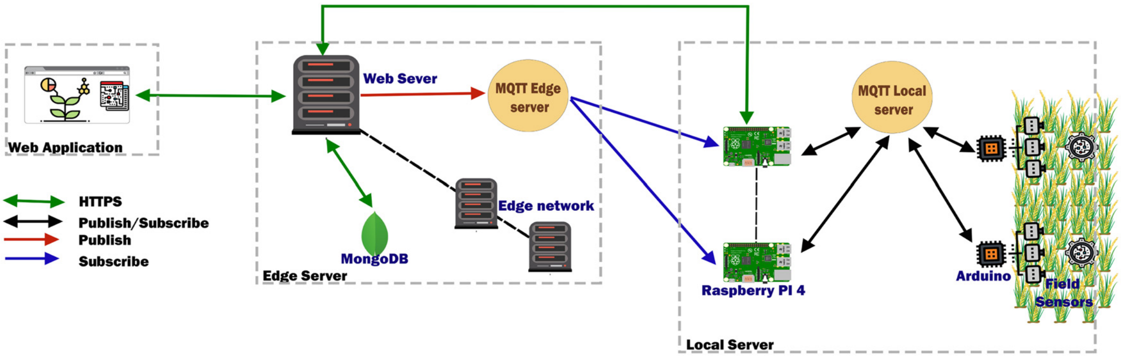


Figure 5: Proposed edge communication model.

communication in the proposed system. The analysis in ref. [36] presented seven IoT messaging protocols (MQTT, CoAP, XMPP, AMQP, DDS, REST-HTTP, and WebSocket) as communication protocols that play a major role in smart farming. The authors have concluded that MQTT proved to be the most secure protocol after probing all the protocols with respect to latency, energy and bandwidth requirements, throughput, reliability, and security. Moreover, MQTT is secure in both end-to-end architecture and gateway server architecture. In an MQTT setup, a MQTT server termed as MQTT broker executes on the IoT solution [37]. Under a common identifier, a "publisher" and a "subscriber" link among themselves to this broker. In the IoT solution, publishers and subscribers are the IoT devices and IoT hubs or control devices, respectively. When the publishers have new data for recording, the data are published to the broker. The broker then flags that it has new publisher data, and the corresponding data are read by the subscriber. Then, the subscriber analyzes the data and reacts accordingly.

The first level accomplishes with connecting the end users to system with the help of mobile or web-based applications through the Internet. The next level (cloud computing server) deals with the connection of web server and MQTT broker for directing the user requests and other components at the edge landscape or from the farms to the right cloud-based services like displaying the real time status of the farm for the users, triggering a new deployment of the updated ML model to the corresponding edge node. The third level (farming area) is directed toward the deployment of sensors and IoT devices (actuators) for communicating with other components in the entire system.

4 Deployment of soil moisture prediction hybrid algorithm

The watering mechanism of the plant has different approaches in the proposed model. Primarily, the system is trained with manual irrigations datasets during the process of learning with respect to suggestions defined by agronomists. The model is trained to learn the needs of irrigation in the first level of deployment in cloud without the inclusion of pre-processed data. After acquiring the required data and training, the proposed system is initiated to grasp the plant's watering needs by undergoing plenty of manual irrigations. Thereafter, manual irrigation is not required and the system makes automated decisions in watering using the gathered data and the application of ML methods. The proposed model then decides the irrigation strategies automatically using ML methods without the need including collected datasets in the automatic irrigation process. The proposed model can be improved through the learning process when the number of precise irrigation inputs is provided to the model at each stage of training.

The decision-making procedure is developed with two modules for irrigation strategies according to the soil moisture prediction for upcoming days. The first module deals with training the model in cloud with manual irrigation datasets through steps such as data collection, data preprocessing, training, and model development. The system acquires values of air temperature (TH), soil temperature (SMT), soil moisture (SM), humidity (HU), and ultraviolet rays (UV) periodically from the physical environment in the data collection stage, which is essentially required for arriving at the watering decisions. Also, the time of performing the manual irrigation is recorded in the database. These data are timestamped and stored in as datasets to aid in making decisions for knowing the time of irrigation. In the next step of pre-processing, inconsistencies are eliminated and outliers caused by sensor errors are detected from the irrigation dataset, thereby helping in the removal of broken data. The training stage involves the application of supervised machine learning (ML) algorithms. Here the regression algorithms such as support vector regression (SVR), multiple linear regression (MLR), lasso regression (LR), decision tree regressor (DTR), random forest regressor (RF), and XG-boost regressor (XB) techniques are used for the deployment. The regression algorithms are trained using the collected datasets. Finally, through training, regression models are created, namely, SVR model, MLR model, LR model, DTR model, RF model, and XB model that are been combined with the second module for decision-making.

The second module caters to the prediction of irrigation for upcoming days by infusing the weather data as an input to the regression trained models. The live datasets from the weather API for future prediction of soil moisture variable are used. The dependent variables from weather forecast data like temperature (TH), humidity (HU), ultraviolet (UV), and precipitation (PC) are tested in the aforementioned model for soil moisture prediction. Then, the regression trained model is evaluated and deployed using the weather testing data for the prediction of soil moisture in accordance with the precipitation. After the prediction of data for the upcoming days, these developed regression models are combined with unsupervised ML algorithm named k-means clustering for estimating the changes incurred in soil moisture prediction due to the impact of weather conditions. Further, each regression models with k-means algorithm are evaluated for performances in terms of irrigation decision-making process as shown in Table 1. The combined algorithms are estimated through MAPE, MSE, R2, execution speed, power consumption, and accuracy. The estimation and computation of these parameters are detailed by the authors in ref. [38].

Table 1: Comparison of performance metrices obtained from various ML algorithms

Algorithms used	Accuracy	R^2	MSE	MAPE (%)	Execution time	Power (J)
SVR + k-means	0.96	0.96	0.25	1.98	0.06078	1164.85
MLR + k-means	0.94	0.88	0.31	2.15	0.02075	429.30
LR + k-means	0.95	0.94	0.32	2.23	0.02482	351.35
DTR + k-means	0.93	0.95	0.29	1.62	0.15687	914.70
RF + k-means	0.95	0.91	0.27	1.57	0.16745	1475.13
XB + k-means	0.97	0.98	0.20	1.08	0.03547	537.87

XGBoost + k-means (XB+k-means) approach provides more accuracy with less MSE comparatively and also the R2 with 98% in soil moisture prediction using combined approach is given in Table 1. It is evident that the proposed combination performs better when compared to other regression + k-means-based approaches. XB + k-means-based hybrid machine learning algorithm is applied in irrigation planning module on account of aforementioned performance metrices of ML. Although it performs moderately in terms of execution time and power usage, it is selected for the deployment in edge computing as it has better performed in terms of accuracy, R2, MSE, and MAPE metrices. It is observed that the prediction of soil moisture for the upcoming days from the proposed algorithm (XB+k-means) is nearer to the actual value as shown in Table 2, and hence, XB+k-means is selected for the implementation of SMPHA in edge-based irrigation scheduling.

Table 2: Comparison of predicted SM value with actual SM value

Date	Average SM value from sensor	Average predicted SM value (XB+k-means)
28-09-2021	35.23	34.04
29-09-2021	36.41	37.20
30-09-2021	31.57	30.46
01-10-2021	34.66	33.15
02-10-2021	36.73	37.12
03-10-2021	32.88	33.01

4.1 Hardware setup

IoT system is crucial to handle, collect, and transfer the data to the computing nodes at the edge or in the cloud. These devices are connected to the edge nodes through wireless communication protocols like ZigBee. It is used in reducing the latency and loss of data. An Arduino micro-control unit controls the combined IoT sensors and actuators at the same part of a field into a cluster, each connected to a Raspberry

Pi that acts as an edge node in processing the gathered data and controlling the actuators. For example, Figure 2 shows an edge architecture with a Raspberry Pi connected to two components: Arduino Uno and Arduino Mega units via ZigBee connection. The first Arduino Mega node is responsible for collecting data from sensors and the second one is for controlling the actuators in the field. Depending on the sensor type with collecting Arduino unit, the sensors are connected via analog or digital PWM pins while controlling Arduino uno joins with actuators in the field and controls (turn on/off) them in accordance with upper layers (from the edge web server). The trained (cloud) and deployed ML model in edge nodes provides the necessary instructions to the edge nodes.

4.2 Web layer setup

The deployment of web server assists the user in planning and managing the irrigation system. It visualizes the crucial information of factors like temperature of air and soil, UV, humidity, and soil moisture in live irrigation with real time updates in the form of various charts. In accordance with the selected field, the web application redirects the user to the field's dashboard as shown in Figure 4. The dashboard consists of field parameters as well as control signals for activating all the physical devices/actuators at the garden layer. These signals are denoted as switch buttons, and each switch controls (turn on/off) a particular kind of actuator (for instance, water pump to start and stop the irrigation). The user interface facilitates remote controlling of the field by just clicking on the buttons as shown in Figure 4.

4.3 Edge layer setup

The edge node acts as a computing center where incoming data are analyzed and fed as the input vector to the ML model for processing and to return the control signals for activating or deactivating the actuators placed at the farm. Edge node processes the physical data (real time) at every end device such as the collected and processed data via the Raspberry Pi nodes presented in the proposed scheme. The prediction model is designed using TensorFlow API and trained, tested on Google Colab in this work. Amazon Web Service (AWS) offers a library named Boto3 having many APIs to upload and download objects. After the development of model, it is transferred to Amazon S3, a service provided by AWS. The edge node utilizes the trained model from S3 for analyzing the sensed data acquired from garden's sensors. The decision is delivered based on real time data analysis at the edge node and transmitted to Arduino nodes in the fields landscape immediately for controlling the actuators. In another flow, the data collected from sensors are filtered so as to keep only the modified data at the edge node before being sent back for mitigating the communication cost to the database in the cloud. These data are used in the updation of the ML model to enhance its efficiency.

4.4 Analytics setup

The main goal of this experiment lies in gathering the various physical parameters of a farming land via sensors and utilizing the fetched data along with weather forecast information for developing an algorithm using hybrid machine learning approach to infuse higher accuracy in predicting the soil moisture for the upcoming days. As discussed in Section 4, for the proper planning and provisioning of optimal irrigation, the algorithm provides a predictable estimate of soil moisture with the assistance of various statistical measures as shown in Table 1. The measures are adopted for estimating the appropriateness and error rate of the proposed algorithm. It is inferred from the experiment that, optimal irrigation is feasible using a good estimation (close to the actual value) of the soil moisture (Table 2), with the support of field data and forecast information, thereby utilizing the natural rain efficiently.

The SMPHA ML model is interdependent on dynamic changes in weather environment where the models deployed on edge nodes need to change the controls accordingly after model gets trained continuously. For the process of retraining, the trained model needs to be updated. The parameters such as TM, HU, ST, UV, SM about grown plants are logged for the training purpose, and these generated datasets are recorded from the already developed manual mode system [39]. The growth of the Indian Mundu Chilli [40] is taken for the observation from the first stage to the last grown stage for 95 days. While retraining the model, the training is carried in cloud without causing effect to the functionalities at edge nodes. A signal is transferred to the corresponding edge server for triggering the task of updating the SMPHA model from the web server. At that time, the newly trained model is downloaded to replace the existing one at the considering edge server. From then, the ML model at the edge server is called to be updated with the real-world knowledge and is ready for its garden controlling tasks (to apply in the next farming season).

4.5 Work flow

The flowchart in Figure 6 depicts the working of the proposed system based on the decision support system that is beneficial for irrigation needed for the growth of vegetables. The chilli plant is grown in a growbag attached with sensors and Pi as shown in Figure 3 and monitored for 95 days of data collection. To bring out optimality in the irrigation system, features relating to climate, soil, crop, and field infrastructure are to be considered. To provide several recommendations in the production of vegetables, decision support systems (DSSs) are designed, which process voluminous information [39]. This proposed work is the extension of soil moisture differences (SMD) model [41] developed for soil moisture prediction. The threshold values of soil moisture are used in the SMD model where the system schedules the irrigation date based on the predicted soil moisture and weather forecast (precipitation) information automatically using SVR+ k-means modeling. Therefore, in the extension of the aforementioned work, further more number of sensors are used to log soil moisture value, which is averaged in the proposed model. This model is developed in two divisions of flowchart as shown in Figure 7, where both are interconnected. It is observed that the prediction of XB + k-mean approach provides better results as presented in Table 2.

The first phase of the flowchart describes the hybrid algorithm for the soil moisture prediction (SMPHA) using the combination of XB + k-means algorithm. During the data collection step, the sensor data for the parameters, namely, TM, HU, ST, UV, and SM, are collected. During preprocessing, null values and outliers are removed and the preprocessed data are used to train the XG-Boost model. The developed model is then trained with variables of live weather features (TM, HU, UV, PC) obtained from Weather API for the prediction of SM data. These data are given as input to k-means clustering algorithm to predict the soil moisture, which is defined as SMPHA value to be infused in the next phase of the flowchart. The second phase of the flowchart defines the automatic irrigation planning setup. The setup starts obtaining the soil moisture maximum (SMMax) and soil moisture minimum (SMMin) values in the dashboard for setting the maximum and minimum level of soil moisture. Then, the current soil moisture (CuSM) is sensed and compared against the threshold SMMin. If the resulting value is less than SMMin, the process proceeds with SMPHA. On the contrary, it stops the irrigation process by sending 0 to the relay. In SMPHA, the nearest precipitation date is selected and it is assigned to the predicted soil moisture (PSM). The SMMax is decided by finding the minimum of (PSM + SMMin, SMMax), and the predicted SMMax is further checked against CuSM with a condition if SMMax is greater the CuSM then it sends 1 to the relay as a signal to start irrigation. If the condition fails, then it sends 0 to stop irrigation. The process of automatic irrigation ends by forecasting the irrigation schedule in accordance with the live weather parameters.

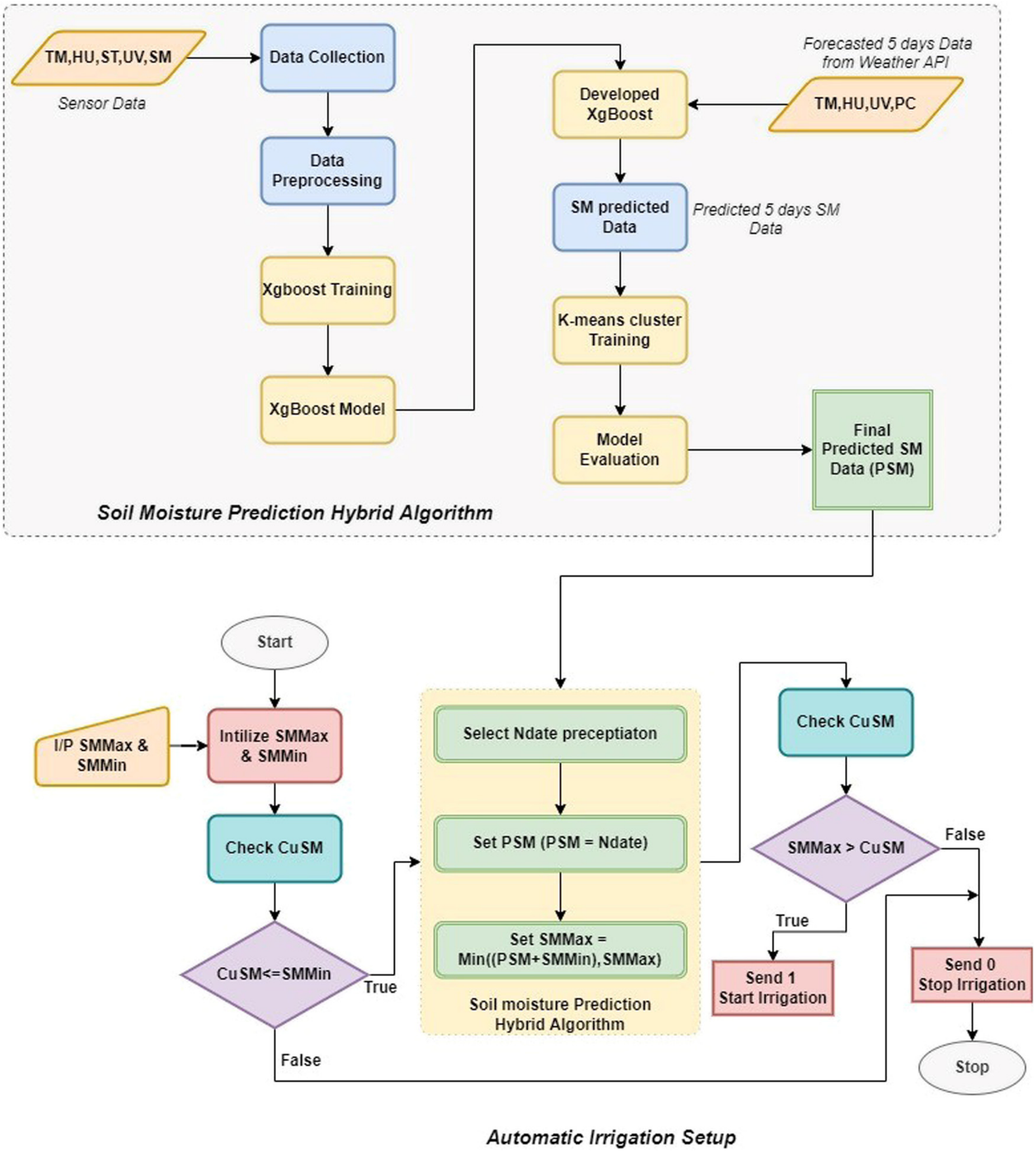


Figure 6: Flow chart of the proposed edge model.

5 Experimental setup and evaluation

The test bed is developed and deployed, and the data are collected for the analysis in irrigation management. Here, Heroku cloud platform is used to deploy the cloud web server. The same cloud is also installed at a local edge that is at two Raspberry Pi units equipped with Wi-Fi 802.11n connections to denote the edge nodes. JMeter application is used to get sequential accesses to the web page from various users for evaluating the network parameters. The specification of these servers is given in Tables 3 and 4.

We evaluated the performance of the proposed IoT-based smart farm on two different platforms, namely, in the cloud and on the local computer to show the feasibility and the benefit of the edge computing scheme. Further many parameters are considered for evaluation and discussed in the next section to show that edge deployment is better than cloud.

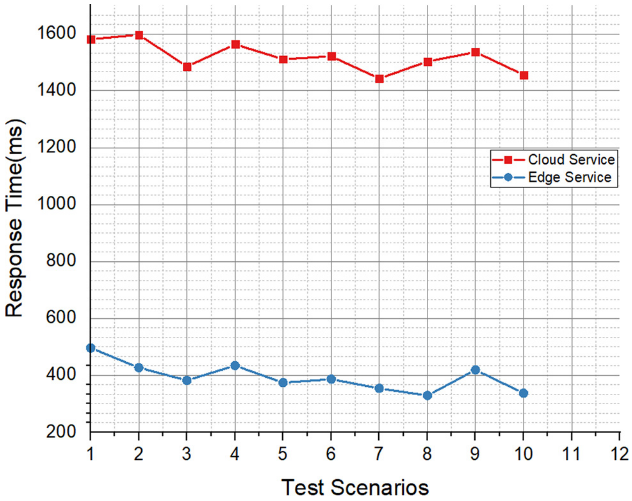


Figure 7: Average response time with 10 test scenarios.

Table 3: Configuration of raspberry Pi

CPU	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5 GHz
RAM	8 GB LPDDR4-3200 SDRAM
Network	2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE, Gigabit Ethernet
Pinboard	8 GB LPDDR4-3200 SDRAM
Operating system, language	Raspbian , Python 3

Table 4: Configuration of Heroku cloud

Country	United states
Service	Amazon web service S3
Processor	2.4 GHz Intel Xeon E5-2676 v3 Processor
CPU Power	8 GB
Virtual CPUs	3–5

5.1 Evaluation

A hybrid machine learning methodology is used in evaluating the first stage of the proposed model. The predicted value of the soil moisture is better in terms of their accuracy and error rate. From the comparison of the other ML algorithms as shown in Table 2, XB + k-means performs better and taken further to be deployed in edge and cloud to check its efficiency with each other. Therefore, for analyzing the efficiency of the edge server in accordance with the proposed hybrid algorithm SMPHA is evaluated in terms of the time taken to train the ML model in edge and cloud. In this experiment Raspberry Pi is used to train the SMPHA model with 196,400 rows, that is, input data sample size and takes around 1,710,000 ms (approximately 28.5 min). The same model when it is trained in Google Colab cloud environment, it takes 204,000 ms (approximately 3.4 min) as depicted in Table 5. The main purpose is to run the trained model on edge not to train the model at edge. So due to the lack of computing capability at the edge, it takes more time to train the model, but it can be ignored as it does not affect the purpose of the proposed model. Here, edge is introduced to obtain the task of computing from the cloud (i.e., offloading the task) by making the system more edge-oriented deployment. It can be accomplished rapidly as it requires only 14 s to download a trained SMPHA model from the cloud to the edge node with a size of 3,101 kb as given in Table 5. The time to

Table 5: Comparison of model training time

	Edge	Cloud
Model training time	28.4 min	3.4 min
Downloading time	Not applicable	14 s

download varies according to the size of the trained model. So, from this process it can be inferred that downloading the trained model saves time when compared to training the model at the edge. Through this in real time, deployment of the trained SMPHA model in edge is better compared to deployment in cloud services. Furthermore, network parameters like latency, throughput, bandwidth, and response time are adopted to measure the performance improvements in edge computing.

The performance metrices taken into account are latency, bandwidth, and response time [42]. The latency of an application is the product of two factors: computing latency and transmission latency. The time spent on data processing and transmission between end devices to cloud servers is termed as computing latency and transmission latency, respectively. The computational capacity of the system decides the computing latency as the network servers possess a considerable amount of capacity to make the data processing faster, whereas the sensors come with limited computing capacity. The latency in transmission is increased by the end devices and cloud servers. Bandwidth: As large number of sensors are deployed in IoT, data generated would be huge that consumes an intense range of bandwidth and leads to several problems such as delay in transmission and loss of packets. It becomes unacceptable for the data to be transferred directly to cloud servers without applying compression. Therefore, data preprocessing and aggregation are needed for IoT gateways before redirecting them to remote cloud servers. Then, the issue to be confronted is to control the traffic flow by migrating data processing and aggregation tasks optimally to decrease the bandwidth needs of the end users while maintaining the data quality. Response time: The total response time is calculated by adding up transmission and processing time. The local deployment of the proposed model for controlling IoT-based irrigation are deployed on two modes: (i) Cloud mode: The developed SMPHA model is implemented in the cloud communicating with IoT sensors nodes directly to manage the irrigation process. The data are stored and processed at the cloud server itself where it uses Heroku platform. (ii) Edge mode – Raspberry Pi is deployed as an edge server that involves in processing of the SMPHA model controlling the IoT sensor nodes. Here, the data are stored and processed locally within the edge servers. This SMPHA model from both the edge and cloud does the job of controlling the actuators to initiate and quit the working of water flow motors. Through this deployment in both the environments, performance of edge server and cloud server can be checked in terms of latency, throughput, bandwidth, and response time is shown in aforementioned graphs in Figures 6, 8, and 9. This performance metrices is not feasible to calculate while deploying in real time, so the aforementioned scenarios of two modes are virtually created by generating many request and response threads between the servers. This sampling, load test, and distributed testing are conducted through JMeter application [43] and also verified with Wireshark [44] in cloud servers. The test scenario is created here by data of sending and receiving sampling data between cloud to IoT sensors and between Edge to IoT sensors. The sampling data considered in this work refer to the approximate number of requests generated by Arduino to cloud and Arduino to Raspberry Pi that are calculated in real time. The test scenario is divided into 10 days of sampling data collected for each day. The evaluation results are depicted for latency and response times in 10 days perspective. In latency parameter, edge service has decreased by an average of 77.85% time compared to the with cloud. In the same manner, the response time of edge service is also decreased by 74.09% time compared to cloud service. In throughput calculation, sampling data are calculated for an hourly basis for the 10 hours data in a day. From the hourly comparisons of throughput value, edge outperforms with 67.17% high Mbps usage. Through this analysis as shown in Table 6, it is evident that the proposed edge computing methodology deployed in Raspberry Pi or in local computers outperforms the cloud-oriented approach.

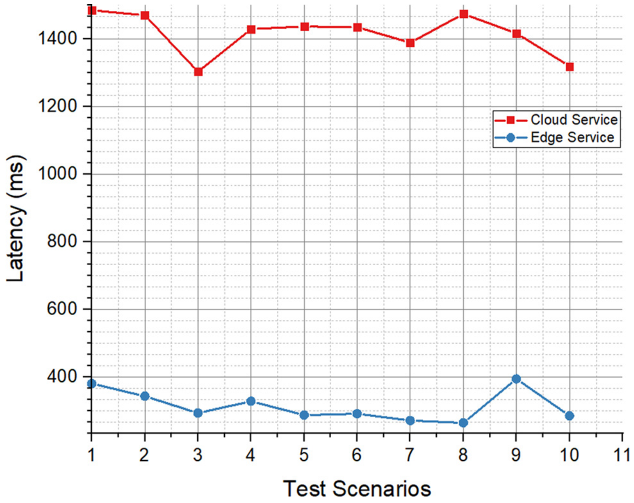


Figure 8: Average latency with 10 test scenarios.

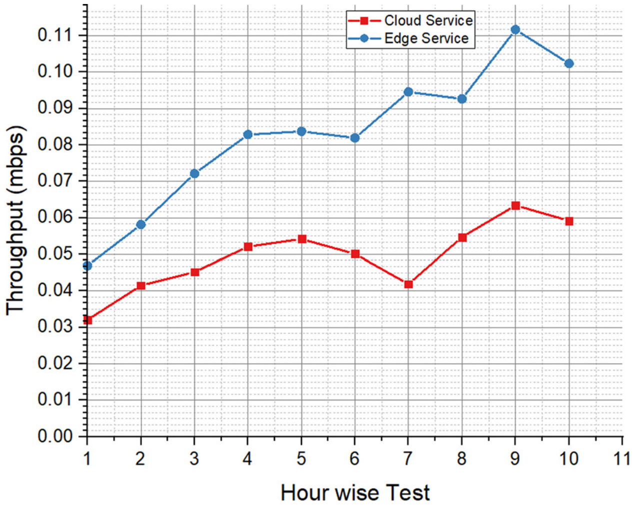


Figure 9: Average throughput value with 10 h test scenarios.

Finally, to illustrate the efficiency of resource management in edge computing, CPU and memory utilization are considered for the analysis as both factors rely on the service execution model and the computational needs of the services being fired from off-loaders. Figure 10 depicts the utilization of CPU and RAM on the Raspberry Pi acting as an edge node in two cases: with and without the deployment of SMPHA model on it. As shown in Figure 10, the SMPHA model affects the CPU of the Raspberry Pi node significantly as it consumed around 41.2% of the CPU compared to only 3.5% when it does not host the

Table 6: Performance metrices for cloud and edge services

Performance metrices	Cloud service	Edge service
Throughput (Mbps)	0.04944	0.08265
Latency (ms)	1415.8	313.6
Response time (ms)	1519.6	393.8
Bandwidth (bps)	86	1,365

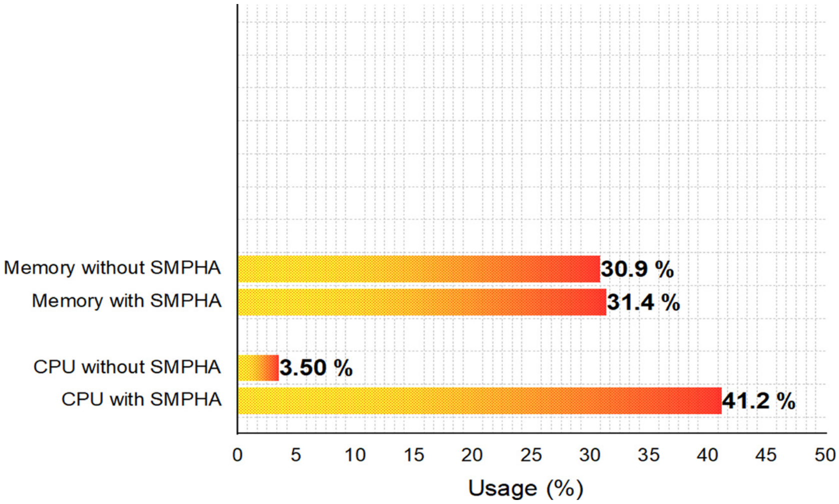


Figure 10: CPU and memory utilization with and without SMPHA.

SMPHA model. However, the memory (RAM) utilization in both the cases (with and without deployment of an SMPHA model) is nearly the same which is around 31%. Comparatively RAM utilization does not have much difference in with and without SMPHA. It is worthwhile to note that, the CPU utilization is still much lower than the 50% of total CPU capacity in Raspberry Pi. Therefore, it becomes feasible for adopting edge server implementation in the proposed irrigation system.

6 Conclusion

This article proposed a novel approach to edge-based irrigation system to facilitate decision-making on watering the plants on scheduled time. The proposed approach applying IoT with an edge computing framework enables the farming system to adapt to the changes in environmental conditions automatically and efficiently. The process of automatic irrigation regulates irrigation according to the live weather parameters for forecasting the irrigation process. Soil moisture prediction was performed using major regression algorithms that are again combined with k-means clustering for estimating the changes incurred in soil moisture prediction. These techniques were compared through metrics such as MAPE, MSE, speed, and power consumption from which XB + k-means was found to perform better. The XB + k-means algorithm was further used for the implementation of decision mechanism on the developed edge computing model. The proposed edge model saves the data communication cost and reduces the response time of IoT services. It can be deployed on existing devices on the network edges serving as edge nodes, thereby reducing the overall implementation cost of a large-scale IoT system. The edge-based approach was found to perform better than the cloud-based approach in terms of response time, latency, throughput, and bandwidth usage. Finally, the edge model was analyzed through CPU and memory usage while running with and without the algorithm. In both cases, the memory utilization is almost lower to total available resource of the edge device. From this, edge device can allocate its remaining resource for other computing services, which increases the efficiency of edge computing device. The number of end edge nodes can be increased according to the field area and then to check the potency of the system.

Conflict of interest: The authors declare no conflict of interest.

Data availability statement: All data that support the findings of this study are included within the article.

References

[1] India: Issues and Priorities for Agriculture, The World Bank, May 17, 2012. https://www.worldbank.org/en/news/feature/2012/05/17/india-agriculture-issues-priorities.

[2] India at a glance in Agriculture, FAO in India. https://www.fao.org/india/fao-in-india/india-at-a-glance/en/.

[3] Cavicchioli R, Ripple WJ, Timmis KN, Azam F, Bakken LR, Baylis M, et al. Scientists' warning to humanity: Microorganisms and climate change. Nature Rev Microbiol. 2019;17(9):569–86. doi: 10.1038/s41579-019-0222-5.

[4] Huong NTL, Bo YS, Fahad S. Economic impact of climate change on agriculture using Ricardian approach: A case of Northwest Vietnam. J Saudi Society Agricult Sci. 2019;18(4):449–457. doi: 10.1016/j.jssas.2018.02.006.

[5] Fagodiya RK, Pathak H, Bhatia A, Jain N, Kumar A, Malyan SK. Global warming impacts of nitrogen use in agriculture: An assessment for India since 1960. Carbon Management. 2020;11(3):291–301. doi: 10.1080/17583004.2020.1752061.

[6] Sarkar S, Chatterjee S, Misra S. Assessment of the suitability of fog computing in the context of internet of things. IEEE Trans Cloud Comput. 2018;6(1):46–59. doi: 10.1109/TCC.2015.2485206.

[7] Porter JR, Xie L, Challinor AJ, Cochrane K, Howden SM, Iqbal MM, et al. Food security and food production systems. In: Field CB, Barros VR, Dokken DJ, Mach KJ, Mastrandrea MD, Bilir TE, et al., editors. Climate Change 2014: Impacts, Adaptation, and Vulnerability. Part A: Global and Sectoral Aspects. Contribution of Working Group II to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change Cambridge, United Kingdom: Cambridge University Press and New York, NY, USA; 2014. p. 485–533.

[8] Lal R. Adaptation and mitigation of climate change by improving agriculture in India. In: S. SherazMahdi (Ed.), Climate Change and Agriculture in India: Impact and Adaptation. Cham: Springer International Publishing; 2019. p. 217–27.

[9] Saravanan K, Julie G, Robinson H. (Eds.), Handbook of research on implementation and deployment of IoT projects in smart cities. Hershey: IGI global, 2019.

[10] Baylis A. Advances in precision farming technologies for crop protection. Outlooks Pest Manag. 2017;28(4):158–61.

[11] Mulla D, Khosla R. Historical evolution and recent advances in precision farming. Soil-Specific Farming Precision Agriculture. Boca Raton: CRC Press; 2015.

[12] Dutta L, and Basu TK. Extraction and optimization of leaves images of mango tree and classification using ANN. IJRAET 2013;1(3):46–51.

[13] Kawai T, Mineno H. Evaluation environment using edge computing for artificial intelligence-based irrigation system. 2020 16th International Conference on Mobility, Sensing and Networking (MSN). Tokyo, Japan: IEEE; 2020. p. 214–9.

[14] Munir MS, Bajwa IS, Ashraf A, Anwar W, Rashid R. Intelligent and smart irrigation system using edge computing and IoT. Complexity. 2021;2021:1–16.

[15] Angelopoulos CM, Filios G, Nikoletseas S, Raptis TP. Keeping data at the edge of smart irrigation networks: A case study in strawberry greenhouses. Comput Netw. 2020;167:107039.

[16] Satyanarayanan M. The emergence of edge computing. Computer. 2017;50(1):30–9.

[17] Shi W, Dustdar S. The promise of edge computing. Computer. 2016;49(5):78–81.

[18] Ramirez Izolan PL, Diniz Rossi F, Hohemberger R, Konzen MP, da Cunha Rodrigues G, Saquette LR, et al. Low-cost fog computing platform for soil moisture management. In: 2020 International Conference on Information Networking (ICOIN). Barcelona, Spain: IEEE; 2020. p. 499–504.

[19] Ferrandez-Pastor F, Garcia-Chamizo, J, Nieto-Hidalgo, M, Mora-Pascual, J, Mora-Martínez, J. Developing ubiquitous sensor network platform using internet of things: application in precision agriculture. Sensors. 2016;16(7):1141.

[20] Xu X, Liu X, Xu Z, Dai F, Zhang X, Qi L. Trust-oriented IoT service placement for smart cities in edge computing. IEEE Internet Things J. 2020;7(5):4084–91.

[21] Wu X, Liu M. In-situ soil moisture sensing: Measurement scheduling and estimation using compressive sensing. In: 2012 ACM/IEEE 11th International Conference on Information Processing in Sensor Networks (IPSN). Beijing, China: IEEE; 2012. p. 1–11.

[22] Kameoka T, Nishioka K, Motonaga Y, Kimura Y, Hashimoto A, Watanabe N. Smart sensing in a Vineyard for advanced viticultural management. In: Proceedings of the 2014 International Workshop on Web Intelligence and Smart Sensing. Saint Etienne France; 2014. p. 1–4.

[23] Cagri Serdaroglu K, Onel C, Baydere S. IoT-based smart plant irrigation system with enhanced learning. In: 2020 IEEE Computing, Communications and IoT Applications (ComComAp.) Beijing, China: IEEE; 2020. p. 1–6.

[24] Kwok J, Sun Y. A smart IoT-based irrigation system with automated plant recognition using deep learning. In: Proceedings of the 10th International Conference on Computer Modeling and Simulation - ICCMS2018. Sydney, Australia: ACM Press; 2018. p. 87–91.

[25] Goldstein A, Fink L, Meitin A, Bohadana S, Lutenberg O, Ravid G. Applying machine learning on sensor data for irrigation recommendations: Revealing the agronomist's tacit knowledge. Precision Agricult. 2018;19(3):421–44.

[26] Vij A, Vijendra S, Jain A, Bajaj S, Bassi A, Sharma A. IoT and machine learning approaches for automation of farm irrigation system. Proc Comput Sci. 2020;167:1250–7.

[27] Krishnan H, Scholar R. MongoDB – a comparison with NoSQL databases. Int J Scientific Eng Res. 2016;7(5):1035–7.

[28] Ojha T, Misra S, Raghuwanshi NS. Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. Comput Electr Agricult. 2015;118:66–84.

[29] Gutierrez J, Villa-Medina JF, Nieto-Garibay A, Porta-Gandara MA. Automated irrigation system using a wireless sensor network and GPRS module. IEEE Trans Instrument Measurement. 2014;63(1):166–76.

[30] Chanthakit S, Keeratiwintakorn P, Rattanapoka C. An IoT system design with real time stream processing and data flow integration. In: 2019 Research, Invention, and Innovation Congress (RI2C.) Bangkok, Thailand: IEEE; 2019. p. 1–5.

[31] Lv H, Wang S. Design and application of IoT microservices based on Seneca. USA: DEStech Transactions on Computer Science and Engineering, (icte.). 2016.

[32] Lee B-H, Dewi EK, Wajdi MF. Data security in cloud computing using AES under HEROKU cloud. In: 2018 27th Wireless and Optical Communication Conference (WOCC). Hualien: IEEE; 2018. p. 1–5.

[33] Lopez Pena MA, Munoz Fernandez I. SAT-IoT: An architectural model for a high-performance fog/edge/cloud IoT platform. In: 2019 IEEE 5th world forum on internet of things (WF-IoT.) Limerick, Ireland: IEEE; 2019. p. 633–8.

[34] Weather API. Retrieved from https://openweathermap.org/api.

[35] Drew Gislason. Zigbee wireless networking, 1st ed. Newnes, London: Elsevier Publisher; 2008.

[36] Tanabe K, Tanabe Y, Hagiya M. Model-based testing for MQTT applications. In: Virvou M, Nakagawa H, Jain LC. (Eds.), Knowledge-Based Software Engineering: 2020. Cham: Springer International Publishing; 2020. p. 47–59.

[37] Babun L, Denney K, Celik ZB, McDaniel P, Uluagac AS. A survey on IoT platforms: Communication, security, and privacy perspectives. Comput Netw. 2021;192:108040.

[38] Rastogi K, Lohani D. Edge computing-based internet of things framework for indoor occupancy estimation. Int J Ambient Comput Intell. 2020;11(4):16–37.

[39] Premkumar S, Sigappi AN. Functional framework for edge-based agricultural system. In: AI, Edge and IoT-based Smart Agriculture, 1st ed. USA: Academic Press, Elsevier; 2021. p. 71–100.

[40] Phani Kumar J, Paramaguru P, Arumugam T, Manikanda Boopathi N, Venkatesan K. Genetic divergence among Ramnad mundu chilli (Capsicum annuum L.) genotypes for yield and quality. Electr J Plant Breeding. 2021;12(1):228–34.

[41] Goap A, Sharma D, Shukla AK, Rama Krishna C. An IoT-based smart irrigation management system using Machine learning and open source technologies. Comput Electronic Agricult. 2018;155:41–9.

[42] Aslanpour MS, Gill SS, Toosi AN. Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research. Internet Things. 2020;12:100273.

[43] Sunardi A, Suharjito MVC architecture: a comparative study between Laravel framework and slim framework in freelancer project monitoring system web based. Proc Comput Sci. 2019;157:134–41.

[44] Robert Shimonski. The wireshark field guide, 1st ed. New York: Syngress Press, Elsevier; 2013.