International Journal of Innovative Computing, Information and Control
Volume **17**, Number **4**, August **2021**
 ISSN 1349-4198
pp. 1225–1241

SMCSIS: AN IOT BASED SECURE MULTI-CROP IRRIGATION SYSTEM FOR SMART FARMING

Venus Wazeer Samawi

College of Information Technology
Isra University
Al Hezam Road, Amman 11622, Jordan
Venus.samawi@iu.edu.jo

Received February 2021; revised May 2021

Abstract. *A large amount of freshwater is consumed to irrigate crops. IoT-based automatic irrigation systems are needed to reduce water consumption. Security and privacy issues, factors concerning climate conditions, and real-time soil and air features that affect irrigation control make the development of such system challenging. This study introduces a cloud-connected, secure multi-crop smart irrigation system (SMCSIS) to address the excessive irrigation problem caused by precipitation after irrigation and to reduce water consumption. SMCSIS makes real-time watering decision on the basis of soil moisture predicted at the time of precipitation. The prediction depends on the data captured from soil moisture sensor and climate prediction-based estimated evaporation. An artificial neural network is developed and trained using five factors (air temperature, wind speed and direction, UV, and humidity) captured from online weather forecasting to obtain the estimate evaporation. A database containing the characteristics and irrigation information for each crop is also developed for multi-crop irrigation. Access control and blockchain technologies are used to maintain privacy and data integrity in the system. A prototype has been developed to simulate the system for small farms. Experimental results indicate that SMCSIS is functional, and it provides an effective solution to overcome excessive irrigation.*
Keywords: Smart multi-crop irrigation system, Blockchain, Soil moisture prediction, Artificial neural network, Evaporation estimation, IoT, Automatic irrigation systems

1. **Introduction.** Agriculture is considered the backbone of food security and the major factor that affects the economy of many countries [1]. The most important element required in agriculture sector is water. In developing countries, the agriculture sector consumes a substantial portion of freshwater compared with that in developed countries. Most irrigation systems in developing countries are ineffective because they either cause excess or scarcity of irrigation [1,2]. The large number of factors (such as climate, soil type, season, crop type and age, soil moisture and temperature, and light) that affect the process of irrigation management makes the process of developing an effective automatic irrigation system a challenging problem. Most of these factors and agricultural data are recently monitored by wireless sensor networks using Internet of Things (IoT) technologies and managed by cloud computing to gain the needed information and make proper irrigation decisions [3-5].

The environmental parameters of soil moisture, soil and water PH, weather conditions (temperature, humidity, ultraviolet [UV] light, wind speed, wind direction, and precipitation), and crop height need to be monitored when developing automatic irrigation systems.

DOI: 10.24507/ijicic.17.04.1225

Most studies monitored no more than three parameters, such as soil, humidity and weather parameters. Few studies utilized crop parameter when developing automatic irrigation systems [3]. In terms of weather parameters, most studies monitored temperature and humidity, whereas very few considered wind (speed and direction) and precipitation. In addition, weather forecasting (collected by sensors or from meteorological station) was used in some studies, especially those who considered machine learning and regression methods when developing the irrigation schedule [2,7-9]. Soil moisture is an important real-time parameter used in most irrigation decision-making systems to determine when to start or stop the irrigation process. By contrast, soil PH and nutrients are rarely used [3]. Various wireless sensor network technologies are utilized to implement IoT-based irrigation systems. Most studies used Raspberry Pi (P2 or P3) and/or Arduino boards, especially Arduino UNO and Arduino Mega, as IoT nodes for irrigation systems [10-13]. WiFi is the most popular communication technology used with automatic irrigation systems due to its low cost. Global System for Mobile Communications (GSM) is used for long-range communication. ZigBee is also employed to reduce energy consumption although it has low data rates compared with other technologies. For prototypes and irrigation systems for small gardens, Bluetooth is used due to its low cost and low energy consumption. LoRa technology is recommended to be used to cover long communication range (up to 20 miles) [3,4].

IoT-based irrigation systems require a large storage area to keep the data generated by sensors during the monitoring process and the information regarding certain crops (such as crop type, season, soil parameters, required fertilizers, and watering schedule based on agronomist opinions). Therefore, databases are needed to keep all the necessary information that could be used to analyze these data for the control of irrigation process and the reduction in water consumption [10,12,14]. Machine learning [2] and artificial intelligent approaches, such as fuzzy logic [15-17], artificial neural networks (ANNs), and regression models [3], are recently used to optimize IoT-based irrigation systems utilizing environmental parameters and weather conditions.

Like most IoT systems, IoT-based irrigation systems need to be secured, because such systems could be a victim of various threats, such as privacy threats, software vulnerability, malwares, denial of service, false data injection, in addition to data integrity problems and confidentiality of end-to-end communication. Therefore, in IoT systems, a privacy and data integrity model is crucially needed [18,19]. Access control polices need to be defined to preserve the privacy and security in IoT systems. A trust model is important to protect the communication in IoT and maintain authentication. Encryption algorithms and authentication are used to secure data and prevent data altering by unauthorized access. Blockchain is recently applied to secure IoT-based irrigation systems through tracing the irrigation decisions made by the IoT system [20-22]. In IoTs, system utilization could be improved by applying scheduling policy (such as rate monotonic policy) to schedule tasks (workload generated by sensors) on the processor [23].

Most of the previous research developed smart irrigation system on the basis of irrigation factors monitored in real time (mainly soil moisture, temperature, and humidity) and online weather forecasting. Few studies considered wind speed and direction [24]. Apart from real-time factors, some researchers used irrigation information on the basis of agronomist opinions. A few research developed smart irrigation system for various crops while considering the crop type and height [10,14]. To the best knowledge of the authors, very few studies considered precipitation in near-future when developing an automatic irrigation system [24]. Such systems could cause excessive irrigation problem. The projected evaporation must be calculated over the time period between the irrigation process and the expected precipitation time to determine the appropriate amount of irrigation and address the problem of excessive irrigation, which could be caused by irrigation of crops shortly before precipitation. In some research, the expected evaporation was calculated (using evaporation equations) to develop a decision support system (DSS) for irrigation scheduling [2].

The present study mainly focused on developing a secured, cloud-connected, multi-crop smart irrigation system (SMCSIS) to overcome the excessive irrigation problem caused by precipitation after irrigation. Employing precipitation forecasting could also reduce water consumption. The contributions of this study are detailed as follows. First, the feasibility of using ANN approach in determining the estimated evaporation (E) on the basis of five factors (that is, temperature, humidity, wind speed and direction, and UV) is investigated. Subsequently, the estimated evaporation (E) and crop characteristic K_c coefficient are used to calculate the estimated soil moisture (ESM) at time T. ESM is used to specify the next irrigation time. Second, two new algorithms are proposed to schedule the next irrigation time on the basis of the calculated ESM at time T, and the expected time of precipitation (after irrigation) to reduce water consumption and overcome the problem of excessive irrigation when rainfall is expected in the near future. The performance of the two suggested algorithms was evaluated from the point of view of water consumption. Third, the developed irrigation system was designed to serve multi-crop fields that utilized a database to determine the proper amount of watering for a certain crop. The database contains information about various crops (e.g., crop type, crop age, day or night irrigation, min/max soil moisture based on agronomist opinions, season, soil type, and K_c coefficient). Finally, the developed system was secured by defining a chain of trustees and creating audit log units to prove user activities and maintain data integrity in real time. The irrigation decision in the developed system is based on some factors (such as soil moisture and light to specify day or night) monitored in real time in addition to online weather forecasting (temperature, humidity, wind speed and direction, UV, and precipitation expectation). Two algorithms were proposed to control the irrigation process. In the proposed system, a set of rules was applied on the factors monitored in real time, weather forecasting (rainfall or not), and E to decide about the irrigation time and amount of water required by a crop in accordance with the database content for that crop. Access control policies were defined to preserve privacy and data security, and a proper ciphering mechanism was applied on the crop database to preventing intruders from altering stored data. A blockchain-based technology was also used to create audit log modules to prove user activities and preserve data integrity. A prototype was developed to control the irrigation process of a 100 m^2 potato farm and simulate and evaluate the performance of SMCSIS. The prototype is an Android mobile application that could control the irrigation process on the basis of the decision made by the irrigation system rules in accordance with the crop information (crop type, crop age, and season) inputted by the user. Access control rules were specified to preserve security and data integrity, and a simple notification system based on blockchain was applied. The developed system is an IoT autonomous system (i.e., autonomous sensor nodes) connected to a wireless sensor network (WSN) with Zigbee.

The rest of this paper was organized as follows. Literature was reviewed in Section two. In Section three, evaporation estimation utilizing ANN was demonstrated, and the performance of the developed ANN was measured and assessed. Section four illustrated the developed irrigation system, at which all the system modules and decision-making algorithms were introduced and demonstrated. The developed prototype, along with the testing and analysis phase, was discussed in Section five. Section six provided the conclusion.

2. **Literature Review.** Much research recently focused on utilizing IoT and WSNs to develop automatic irrigations systems and reduce water consumption. Several studies were also conducted to predict the watering schedule that reduces water consumption [24]. Researchers used machine learning methods to predict soil moisture and determine the estimated schedule for irrigation. In 2018, Goap et al. developed an IoT and machine learning (ML)-based smart irrigation system. The developed system utilizes sensed data and weather forecast data (from the Internet) to predict the irrigation requirement for a certain field. The sensed data are soil moisture, soil temperature, air temperature, ultraviolet (UV) light radiation, and relative humidity of the crop field. The used forecast attributes are precipitation, air temperature, humidity, and UV for the near future. Machine learning algorithms were proposed on the basis of applying a support vector regression (SVR) model and k-means clustering algorithm to predict the soil moisture in the crop field. The accuracy of the predicted soil moisture through SVR and k-means outperformed the predicted soil moisture by using SVR only [2]. Goldstein et al. [7] applied an ML model on a dataset constructed by collecting data from various sources (soil sensors, a meteorological station, and irrigation records defined by an agronomist) to predicting the weekly irrigation schedule. Three ML approaches were applied on the dataset to predicting the irrigation schedule: linear regression (LR), gradient boosted regression tree (GBRT), and boosted tree classifier (BTC). Comparison of the two regression approaches showed that GBRT outperformed LR, with accuracy reaching 92%. In 2020, Torres-Sanchez et al. studied three learning techniques, namely, LR, random forest regression (RFR), and SVR to specify their efficiency in developing a robust model expert decision system for irrigation management. The developed system utilized climatic and soil data for nine citric crops in Southeast Spain to determine the system accuracy. The authors found that RFR was the best in emulating the agronomist in specifying the irrigation schedule for various crops [3]. Shalini and Aravinda [25] introduced a study and predictive analytic model to determine the expected amount of precipitation for irrigation. In this study, four ML techniques (multiple LR, k-nearest neighbor, decision tree, and random forest techniques) were used to construct the predictive model. The performance of each of the four techniques was evaluated on the basis of root mean squared error (RMSE) for choosing the best technique. The results showed that random forest outperformed the other three techniques. Artificial neural network is used to estimate soil moisture and develop IoT-based irrigation systems. Adeyemi et al. [26] utilized an ANN to develop a dynamic model for one-day soil moisture flux prediction on the basis of previous information concerning soil moisture, precipitation, and climate. The crop-water productivity model, AquaCrop, was used to explain the developed ANN model in predicting the irrigation schedule for potato-growing season. The predictive irrigation schedule was compared with predefined irrigation schedule. The experimental results showed that the predictive irrigation schedule is comparable to the predefined irrigation schedule and saves approximately 20%-46% of irrigation water. Nawandar and Satpute [27] proposed an IoT-based low-cost smart irrigation system utilizing ANN to enhance decision making and improve irrigation efficiency. The developed system provides remote crop monitoring using MQTT and HTTP to update the user about crop state. The experimental result of irrigation schedule, decision making based on NN, and remote data monitoring for sample crop was illustrated to demonstrate the system performance. The developed model suits greenhouse and farms. Some of the developed irrigation systems focus on security, privacy, and data integrity. An irrigation system was developed to organize and manage the use of water in rural areas [21]. The developed system utilizes blockchain technologies to maintain trust among system users (community members) and resource devices. Munir et al. [22] developed an intelligent smart watering system (SWS) based on fuzzy

logic approach, and supported by an Android application to control water consumption in small crop fields. Blockchain technology was also used to preserve security and privacy. The experimental results indicated that SWS is effective and it secures irrigation application. In this study, access control policies were defined to preserve privacy. Moreover, a blockchain-based technology was also used to create audit log modules to prove user activities and preserve data integrity.

Most studies consider soil moisture prediction significant in developing automatic irrigation system. In [2], five environmental parameters (soil moisture (read by sensor), air temperature and humidity, soil temperature, and UV) are utilized to estimate soil moisture. In [26], past soil moisture, precipitation, and climatic measurements were utilized to estimate soil moisture. Few studies utilized crop parameter and wind speed and direction when developing automatic irrigation systems [3,24]. In this study, five environmental parameters (air temperature, humidity, wind speed and direction, and UV) are utilized to estimate soil moisture in addition to K_c coefficient. In terms of soil moisture prediction, various approaches have been applied to predicting soil moisture. In [2], ML approaches (SVM and k-means) have been applied, whereas [26] applied ANN. In the present study, ANN has been applied to estimating evaporation. Subsequently, soil moisture is estimated on the basis of soil sensor read, K_c coefficient, and estimated evaporation. In previous studies, different approaches are developed to schedule irrigation based on on-field sensor reads, regression models, machine learning, and ANN. Most studies did not consider rainfall in the near future when developing the irrigation schedule [2,3,7]. Predictive analytic model has been developed in [25] to predict the expected amount of precipitation for irrigation based on machine learning. In this study, two algorithms are suggested to schedule irrigation in rainy and non-rainy days. Finally, the suggested irrigation system serves multi-crop fields that utilize a database (crop type, crop age, day or night irrigation, min/max soil moisture based on agronomist opinions, season, soil type, and K_c coefficient) to determine the proper amount of watering for a certain crop.

3. **Evaporation Estimation Based on ANN: Proposed Technique.** The feed-forward NN (FFNN) is a machine learning approach that has the potential to approximate the function represented by the dataset [28]. Therefore, we applied the FFNN method for estimating evaporation, which in turn is used to estimate soil moisture. In this study, a supervised feed-forward NN (FFNN) was proposed to estimate the evaporation. First, an evaporation dataset containing data collected every hour for 4 years to test the feasibility of using ANN for evaporation estimation. The dataset contained 26280 records. It needed to be cleaned up as some records contain null values. After the dataset was cleaned, the total valid records became 24890. Each record consists of six fields: air temperature, relative humidity of air, wind speed, wind direction, Net radiation, and evaporation field. An FFNN, which consists of three layers (input nodes, one hidden layer, and one output node), was developed and trained using the dataset. The suggested FFNN consists of five input nodes (air temperature, relative humidity of the air, wind speed, wind direction, and Net radiation), one hidden layer with 10 nodes, and one output node (evaporation). It was trained utilizing Levenberg-Marquardt optimization with learning rate $\alpha = 0.45$. The average performance of the five folds, as measured using RMSE, was 0.0523. Five-fold cross validation was utilized to evaluate the performance of the developed FFNN, at which the dataset was partitioned five times into five sets, each set of size 4978 records. The suggested FFNN was trained five times, where four sets were used for training and one set for testing. On the basis of the promising results of the testing dataset (evaporation dataset), actual data were collected (using online weather forecasting) to obtain the five features (air temperature, relative humidity of air, wind speed, wind direction, and

UV), in addition to a soil moisture sensor, which was used to read soil moisture (SM_t) every 30 minutes. The actual evaporation was calculated using Equation (1) to act as the actual output field. The training dataset was constructed by collecting data for 60 days from five different regions in non-rainy days.

$$E_{avr} = abs(SM_t - SM_{t-1}) \tag{1}$$

After the dataset was collected and cleaned, FFNN was developed with three layers (input nodes, hidden layer, and one output node). The number of nodes in the hidden layer was specified using validation data, in which various numbers of nodes were tested (between 1 and 14 nodes). Ten nodes in the hidden layer showed the best performance. The developed FFNN was trained utilizing Levenberg-Marquardt optimization with learning rate $\alpha = 0.45$. The estimated evaporation (EE_{avr}) could then be obtained for a certain hour in the next day by testing the FFNN with five attributes (air temperature, relative humidity of air, wind speed, wind direction, and UV) through accessing the online weather forecasting. The EE_{avr} (at a certain hour) that resulted from applying the trained FFNN on the five features specified by the online weather forecasting was compared with E_{avr}, which was calculated using Equation (1), to evaluate the behavior of the developed FFNN in real time. Monitoring of the system behavior for 6 days (test data) revealed that the average RMSE = 0.089041919. Compared with the RMSE when utilizing the testing dataset above, the RMSE increased (from 0.0523 to 0.089041919) because UV was used instead of Net radiation in addition to margin of error that resulted from using online weather forecasting. A notable detail that selects the proper feature set highly affects the system accuracy. Therefore, FFNN was trained by excluding one feature at a time (i.e., ANN was trained with four features). The results showed that the feature that could be excluded without majorly affecting the system performance was when excluding "wind direction". Table 1 illustrates the performance of the FFNN in obtaining EE_{avr} compared with E_{avr}. The results showed that using the five features to obtain EE_{avr} is recommendable.

TABLE 1. The performance of the FFNN in obtaining EE_{avr} compared with E_{avr}

NN architecture (input × hidden × output)	The performance of FFNN (RMSE)	
	Utilize TestWing dataset	Utilize Actual data
$5 \times 10 \times 1$	0.077141919	0.089041919
$4 \times 10 \times 1$	0.080663447	0.091963447

4. **Secured Multi-Crop Smart Irrigation System (SMCSIS): The Proposed Model.** In most automatic irrigation systems, farmers specify irrigation schedule to control the irrigation process. Nowadays, IoT-based irrigation systems (ubiquity applications) are developed to the control irrigation process to reduce water consumption. Various types of sensors (soil moisture, WL, soil temperature, pH of water and soil, and leaf wetness) are used in smart irrigation systems [25,29]. In most smart irrigation systems, the irrigation decision and the amount of used water are based on the sensors read at time t. However, rainfall after irrigation causes crop withering due to excessive irrigation. Therefore, SMCSIS, which is capable of avoiding excessive irrigation in case of precipitation in near future, was developed in the present work. SMCSIS is provided with a database (multi-crop database) that contains information about the amount of water needed by each crop (irrigation information) on the basis of its type and age (growth stage), soil type, season, and K_c coefficient. The irrigation information in the database is populated in accordance with the agriculture prescription provided by agronomists. The developed SMCSIS model

mainly consists of three layers, namely, application layer, data processing, and decision-making layer, and data collection and control layer. In addition to maintaining privacy data through defining trustees and specifying the access control policies for them, the application layer serves as a user-interface layer that provides user services concerning user log in, user authentication, message passing between user, and data processing, and a decision-making layer. The second layer is the data processing and decision-making layer (located in cloud), which controls the irrigation process on the basis of the attribute values (factors) entered by the user and the irrigation records in the multi-crop database. The data processing and decision-making layer mainly consists of three modules: multi-crop database module, engine module, and audit log module. Multi-crop data, along with the irrigation information of each crop, are saved in the database module, which is updated by authorized users. Data are processed and decisions are made in the engine module. The audit log module, which tracks user events and actions, is utilized to assist in preserving transparency and data integrity. The third layer is the data collection and control layer, which collects data from a real field utilizing on-field sensors (soil moisture sensor and UV light sensor). Moreover, data are collected from a WL sensor in the water tank to control the actuators of tank-fill water pumps. This layer also controls the water pump actuator for field irrigation on the basis of the decision made by the engine of the irrigation decision. Figure 1 illustrates the main model of SMCSIS.

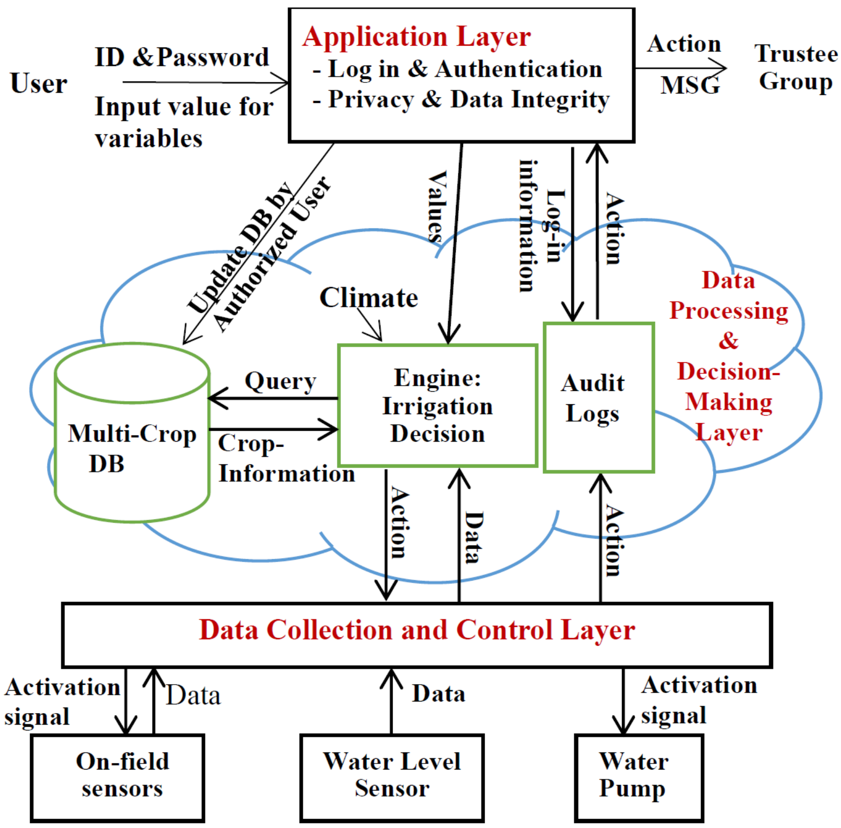

FIGURE 1. SMCSIS: The proposed model

4.1. **Application layer.** The first layer is the application layer, which is a mobile application accessed by authorized users (based on login with username, password, and authentication code sent to the user). This layer acts as an interface between the users (trustee group) and the data processing and decision-making layer. The main tasks of the application layer are data collection and transmission to the data processing and decision-making layer, and notifying the trustee group about all the activities accomplished by the system and the users. The data entered by the user concerning crop field information (crop type, age, season, and soil type) are passed to the *Engine of Irrigation Decision* in the data processing and decision-making layer to be processed. The end user is also able to update the multi-crop database after approving end-user authorization. The log-in information of the end-user is passed to the audit logs to preserve security through monitoring and documenting all user activities. Finally, all messages concerning user and system activities are posted to the trustee group through the application layer.

4.2. **Data processing and decision-making layer.** A data processing and decision-making layer was developed to process data and then makes proper irrigation decision to manage the irrigation process. This layer is located in cloud.

4.2.1. *Database module: Multi-crop database.* Irrigation scheduling decision is needed to determine the amount of water required for irrigation and when to irrigate. The irrigation method and scheduling vary in accordance with crop type, growth stage, soil type (sand or clay), season, needed soil moisture, light (day or night), humidity, and temperature. Real-time data (such as current soil moisture, light, temperature, and humidity) are collected by the sensors, while the amount of water required by a crop is determined by agronomists on the basis of crop features. Therefore, a database was developed to retain the crop features (*crop type, crop age, day or night irrigation, min and max soil moisture based on agronomist opinions, season, soil type, and K_c coefficient*), the amount of water required for irrigation (*min, max, and average soil moisture threshold*), and the number of times to irrigate (per day or week). The information is used to set the initial irrigation schedule and when to activate the sensors. The crop characteristic K_c coefficient of each crop is also kept in the database to be used in calculating the estimated evaporation (as explained in the engine model). The K_c coefficient varies between 0 and 1 according to the periods of growth of the crop (as defined by experts). Therefore, for each crop, three K_c values are kept in the database (initial stage, mid-stage, and final stage) [30]. In the present study, the database was developed using MySQL, and information was specified on the basis of agronomist opinions and irrigation guides for farmers. The database could be updated by authorized user(s), as specified by the system access control policy.

4.2.2. *Engine module: Irrigation decision.* This module is responsible for making irrigation decision that is based on crop features (factors) stored in the multi-crop database, field features entered by the user, and factors monitored in real time from the field (soil moisture and UV light radiation to specify day or night). Online climate forecasting is used to obtain wind speed, wind direction, UV, and precipitation expectation on the basis of the geographical position of the field (as specified by the user). The developed engine makes the irrigation decision in accordance with real-time soil-moisture sensing, precipitation expectation, and the average amount of soil moisture required by a crop (as specified in the database). One of two algorithms could be used depending on the season: *Irrigation Algorithm without Precipitation (NPI)* and *Irrigation Algorithm considering Precipitation (PI).* Both algorithms predict E_{avr}/hour for 5 cm-shallow soil obtained by the trained NN by using climate expectation from online climate systems. For both algorithms, the time of the next irrigation (T_{EI}) must be estimated using the T_{EI} algorithm on the basis of EE_{avr} calculated by FFNN/hour (utilizing the weather forecasting read). First, the real-time soil moisture (SM_t) must be read and treated as ESM at time t. The next step is to find (H), which indicates the hours needed to reach the minimum soil moisture (min) required by the crop (as specified in the database). The expected soil moisture (ESM)/hour (ESM_{t+1}) must be calculated on the basis of crop characteristic K_c coefficient and the EE_{avr} calculated at time t by utilizing FFNN (as illustrated in Section 3 above). Equations (2) and (3) illustrate how to obtain ESM at time $t+1$.

$$E_{avr(t)} = EE_{avr(t)} \times K_c \tag{2}$$

$$ESM_{t+1} = ESM_t - E_{avr(t)} \tag{3}$$

T_{EI} Algorithm: The estimate time for the next irrigation

Input and variables:

- T is the current time scheduled for irrigation
- ESM_{t+1} is the expected soil moisture at time $t+1$ (*using Equations (2) and (3)*)
- SM_T is the soil moisture sensor read at time t
- Min: minimum soil moisture for a crop
- h is number of hours until the next irrigation (*the duration between current SM_T until reaching minimum soil moisture for the crop "Min"*)

Output

- T_{EI} is the approximated time of the next irrigation (*initially equals the current date and time of irrigation*)

Step 1: $ESM_t = SM_t$

Step 2: Find h, accumulate hours by repeatedly applying Equations (2) and (3) until $ESM_{t+1} \leq Min$

$$h = \sum_{t=T}^{ESM_{t+1} \leq Min} 1$$

Step 3: T_{EI} = current time and date + h

NPI Algorithm: Irrigation algorithm without considering precipitation

Input and variables:

- T is the current time
- T_{EI} is the expected time of the next irrigation (*algorithm T_{EI}*)
- ESM_{TEI} is the expected soil moisture at time T_{EI} (*using Equations (2) and (3)*)
- SM_T is the soil moisture sensor read at time T
- Min, Max, Av are the minimum, maximum, and average amount of soil-moisture required by a crop (*as specified in the multi-crop database*)
- E_{avr} is the average expected evaporation/hour for 5 cm shallow soil (obtained by the trained FFNN) *using climate expectation from online climate.*
- TE: the total expected evaporation between T and TRF, where $TE = \sum_{i=T}^{TRF} E_{avr_i}$
- H: time of next irrigation

Output: IR (irrigation status)

Step 1: If current time (T) and date match the schedule time and date of the target crop (H), then

$$SM_T = \text{Read Sensor ()}$$

Step 2: Find time schedule for the next read

If ($SM_T \leq Min$) then IR = Irrigate (SM_T, Max)

Else No irrigation

SM_T = Read Sensor ()

$H = T_{EI}$ (SM_T, current time and date) // *call T_{EI} to calculate time and date of next read*

Goto **Step 1**

Irrigate Function (SM_T, Max)***:***

Input and variables:

- NT is the time till the next read. (*It is specified by the user based on water pumping amount*)

While $(SM_T < Max)$
  Send 1 for relay to carry on with irrigation
  $T = T + NT$
  SM_T = Read Sensor ()
    Send 0 for relay to stop irrigation

PI Algorithm: Irregation algorithm considering precipitation

Input and variables:

- T is the current time
- TRF is the time of rainfall expectation $(TRF > T)$
- SRF: the time of stop raining (*based on online climate forecasting*)
- RFP_{TRF}: the percent of rainfall at time TRF (*based on online climate forecasting*)
- ESM_{TRF}: the expected soil moisture at time TRF (*using Equations (2) and (3)*)
- SM_T: the soil moisture sensor read at time T
- Min, Max, Av are the minimum, maximum, and average $((Max-Min)/2)$ amount of soil-moisture required by certain crop respectively (*as specified in the multi-crop DB*)
- E_{avr}: the average expected evaporation/hour for 5 cm shallow soil (*obtained by the trained FFNN*)
- TE: the total expected evaporation between T and TRF, where $TE = \sum_{i=T}^{TRF} E_{avr_i}$
- Climate expectation (*obtained from the online climate forecasting*)
- H: time of next irrigation

Output: IR (irrigation status),

If time (T) match time of next irrigation (H)

Step 1: SM_T = Read Sensor ()

Step 2: Find time schedule for the next read
  If $(SM_T < Min)$ then IR = Irrigate (SM_T, Average)
    SM_T = Read Sensor ()
  $H = T_{EI}$ (current date and time, SM_T) // find time of next irrigation

Step 3: Based on climate expectation for the next hours (H), get TRF, SRF, and RFP_{TRF}
  Case 1: No rainfall within time duration till H
    If $(SM_T \leq Min)$
      IR = Irrigate (SM_T, Max)
    Else $IR = 0$ //no irrigation
  Goto **Step 1**
  Case 2: Rainfall at time TRF & $TRF < (H)$
    $IR = 0$
    At time SRF Goto **Step 1**
  Case 3: Rainfall at time TRF & $TRF \approx (H)$
    $TE = \sum_{i=T}^{TRF} E_{avr_i}$ // based on Equation (2)
    $ESM_{TRF} = (SM_T - TE)$
    Case 3.1: $(5\% < RFP_{TRF} \leq 20\%)$
      IR = Irrigate (SM_T, Average)// irrigate until soil moisture reaches average moisture

At time SRF Goto **Step 1**
Case 3.2: $(RFP_{TRF} > 20\%)$
If $(ESM_{TRF} \leq Min)$ then IR = Irrigate (SM_T, Min)// irrigate
Else $IR = 0$ // no irrigation
At time SRF Goto **Step 1**
Case 3.3: $(RFP_{TRF} \leq 5\%)$
IR = Irrigate (SM_T, Max)// irrigate
Goto **Step 1**

4.2.3. *Audit logs module.* Audit logs are used to track user events in a sharing environment to verify compliance with privacy policies. Therefore, user actions should be listed in the audit logs [31,32]. In the present study, audit trail (audit logs) was employed to preserve privacy and data security through monitoring and documenting user access and their activities. Audit logs list user access, activities, data modification attempts, and timestamp of each access and activity to ensure data integrity. User activities, such as user request for irrigation decision for certain crop-field, data values entered by the user (crop type, crop age, soil type, and season), and database updating, are documented with the timestamp of user access and activities (log record indicates what event is performed by whom). Each user access and activity will be reported to the trustees (through the application layer) utilizing blockchain technique to preserve data integrity and enhance information security.

4.3. **Data collection and control layer.** In the IoT system, the data collected by sensors are periodically sent to the Internet or Intranet to be processed by the server and interpreted on a front-end interface. In WSN, the sensors connect to a router, where data are passed to their final destination via the router. Thus, the IoT system could be connected to WSN routers to collect the data from the sensors [33]. Either an IoT autonomous system or WSN could be used on the basis of the requirements of a developed system. In the present study, the developed irrigation system (SMCSIS) uses the IoT autonomous system (i.e., autonomous sensor nodes), which is connected to WSN with Zigbee. Figure 2 illustrates the framework of the WSN for SMCSIS, whereas Figure 3 illustrates the schematic diagram of one-set of on-field sensors. In SMCSIS, three on-field sensors (soil moisture, UV light, and WL in tank) and a water pump actuator are connected to Arduino-Uno (microcontroller), which is connected to Zigbee. Arduino-Uno reads the output of these sensors. Data are collected by the sensors in a periodic or non-periodic time (i.e., the user could run the sensor readings when needed). The periodic sensor reading (e.g., hourly) is determined by the user on the basis of soil type, season, crop type, and age.

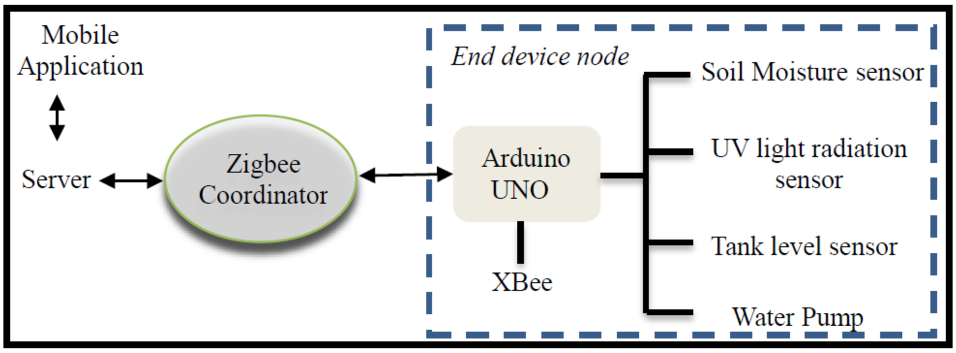

Figure 2. SMCSIS: Framework of the IoT system connected to WSN

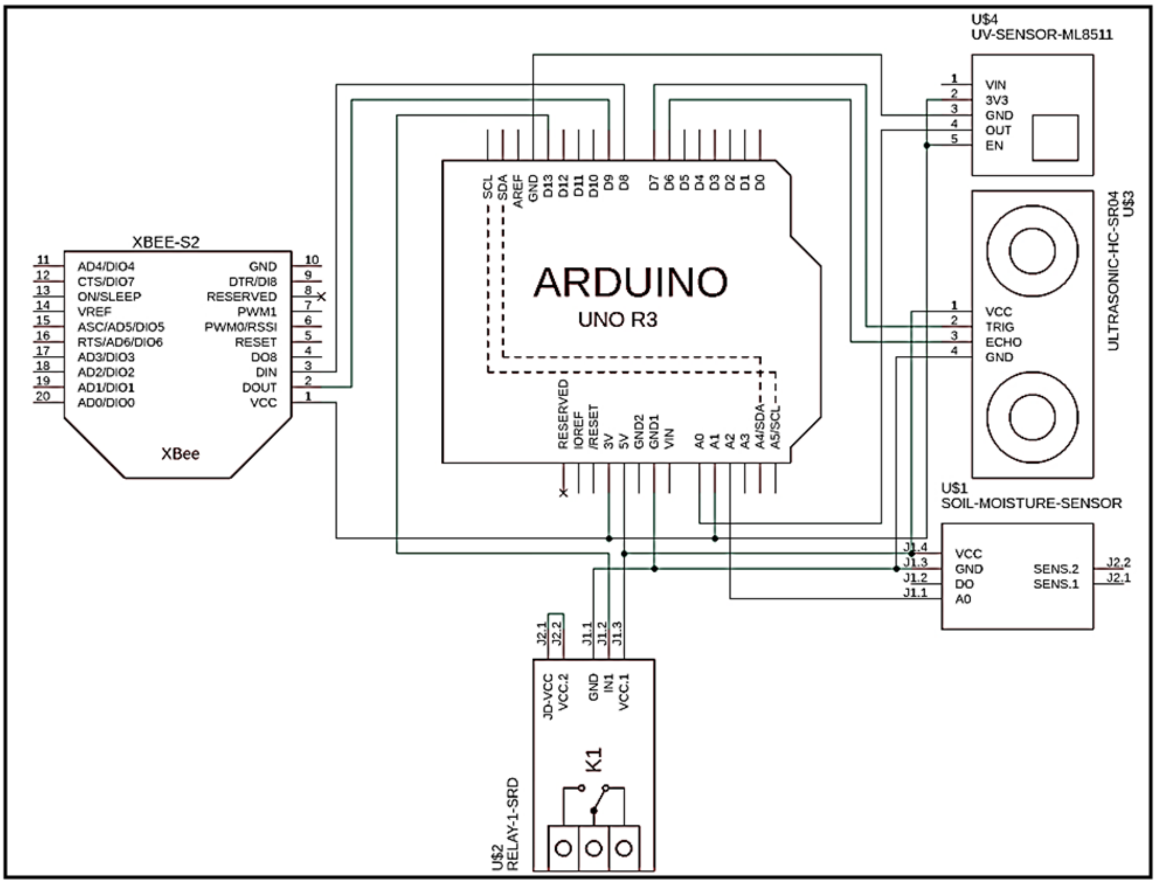

FIGURE 3. The schematic diagram on one set on-field sensors connected to WSN

The data collection and control layer also activates/deactivates the water pump. For instance, water pump is activated to pump water from the water tank to the farm when needed. The amount of water required for irrigation is specified on the basis of sensor reading and crop information in the database to control the irrigation process. The designed irrigation system is a mobile application that helps in dynamically scheduling the irrigation process in accordance with the crop information in the database, sensor reads, and climate prediction (from online climate forecasting). A sensor is used to check the WL in the tank to fill it when WL < WL-threshold in the tank.

WL Algorithm: Fill the Water-Tank

Input and variables:

- Water-Level-Threshold: Minimum water level that should be in the tank before starting irrigation

Tank-Water-Level = Read Sensor ()
    If Tank-Water-Level < Water-Level-threshold, then
        Send alarm (water shortage).
        Start to fill the tank
    Else If (Tan-Water-Level < Water-Min-level) and the pump is pumping water, then
        Stop pumping
        Send alarm (tank empty).

From the database, the system could determine the quantity of water (in liters) needed for irrigation and specify the irrigation schedule, and the irrigation duration is determined by the *Engine Model* by utilizing soil moisture (sensor reads), light (day/night), and the required irrigation for a certain crop (as specified by the database).

5. **Testing and Analysis.** A prototype of SMCSIS was developed and tested on a small farm area to observe its performance. The prototype was designed and implemented to control the crop irrigation on scientific bases. A small part of a farm (100 m^2) was dedicated for testing the behavior of the prototype.

5.1. **Experimental results.** As well known, the accuracy of the ESM is a significant factor for the success of the proposed method. Consequently, the accuracy of ESM must be found and compared with the actual reads. Thus, the actual soil moisture was read using on-field soil sensor and ESM was calculated for 7 days (three times/day). The ESM is obtained based on the EE_{avr}, which is calculated utilizing FFNN/hour (as illustrated in Section 3 above) and crop characteristic K_c coefficient (as shown in Equations (2) and (3)), which illustrates how to obtain ESM at time $t+1$. Table 2 illustrates the sample of evaporation based on actual evaporation for 7 hours and EE_{avr} obtained after applying the developed FFNN.

TABLE 2. Sample of evaporation based on actual data and EE_{avr}

Evaporation in mm/hours	EE_{avr} (obtained by FFNN)	Absolute error
0.030112	0.0629	0.032788
0.048703	0.07261	0.023907
0.047237	0.070818	0.02358
0.041905	0.073469	0.031564
0.036698	0.075245	0.038547
0.014115	0.075136	0.06102
0.005692	0.068293	0.062601

Figure 4 illustrates a comparison between actual soil moisture and ESM for a 7-day duration. The RMSE between the actual soil moisture and ESM was approximately 0.39, whereas the MSE was approximately 0.154.

The prototype of SMCSIS was applied on small part of a 100 m^2 farm for growing potato. A set of on-field sensors was deployed to collect data from the farm. For evaluation of the system performance, the system prototype was used to control the irrigation process for the period between September and January. Three parts of the farm (all had the same area of 100 m^2) were irrigated under different scenarios to test and validate the performance of the irrigation system. The first scenario was performing manual irrigation under the supervision of an agronomist. In the second scenario, the irrigation process was performed automatically on the basis of on-field sensor reads. In this scenario, the on-field soil moisture was read three times per day in case of hot-dry weather. Otherwise, it was read two-times/day. The watering decision was made on the basis of the degree of soil moisture compared with the minimum permissible moisture for the crop. This scenario did not consider the rainfall within the next hours. In the third scenario, the prototype of SMCSIS was applied to controling the irrigation process. Table 3 illustrates the ratio of water consumption of scenarios 2 and 3 with respect to scenario 1. The ratio of water consumption was calculated for two cases (non-rainy and rainy days). In the non-rainy days, scenario 2 and scenario 3 showed comparable improvement in water consumption because scenario 3 applied the scenario 2 algorithm. The small increase in water consumption in scenario 3 was due to the time estimation of the next read that was based on weather forecasting. In the rainy days, scenarios 1 and 3 outperformed scenario

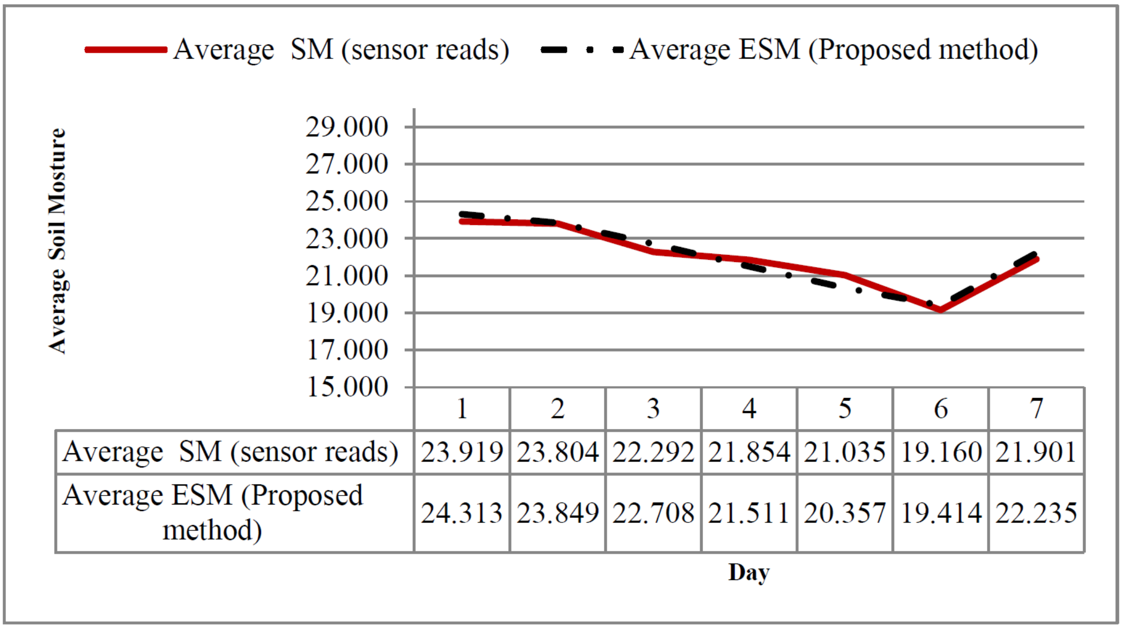

Figure 4. Actual and estimated soil moisture for 7 days duration

Table 3. Water consumption (WCon) scenarios 2 and 3 with respect to scenario 1

Cases	Scenario 2 (S2)		Scenario 3 (S3)	
Non-rainy days	$WCon_{S2}/WCon_{S1}$	Water saving	$WCon_{S3}/WCon_{S1}$	Water saving
	0.83612	0.16388	0.841173	0.158827
Rainy days	1.044383 (excessive irrigation)		0.909847	0.090153

2 because they considered the rainfall within the next 24 hours before making irrigation decision.

The significant problem faced by potato cultivation is rot, which is caused by excessive irrigation. The problem of excessive irrigation appears when the watering process is carried out shortly before the rain. The experimental results showed an increase in rotting ratios of the potato crop in the second scenario (approximately 6%) compared with that in the other two scenarios (very low rotting ratios of ∼1%). The experimental results proved the effectiveness of the proposed system in saving water and avoiding excessive irrigation. Finally, the suggested FFNN proved to be effective in estimating evaporation based on the five factors (air temperature, humidity, wind speed and direction, and UV).

5.2. **Flow of control.** The sensors are initially activated on the basis of the irrigation schedule for the crop under normal conditions of the season (or manually by the farmer). The amount of water for irrigation is specified on the basis of climate forecasting, soil-moisture sensor reads, and crop information. The time of the next sensor activation is also specified, and the irrigation schedule will be updated correspondingly. The designed system is a mobile application. User could use the mobile application and set information about the crop to the Arduino board. The irrigation process is illustrated in the irrigation algorithm of the prototype.

The prototype of the proposed irrigation algorithm

Setup stage

1) Register trustee group, and specify users authority

2) Construct crops database

Processing stage

1) User log-in and authentication

2) Enter information about the crop (crop-type, crop-age, soil-type), and the season

3) The system will search the database to specify the initial irrigation schedule, K_c factor, and the Min/Max required soil moisture

4) Send the commands to Arduino to activate soil moisture sensor (SM), and UV (Light) sensor.

5) Receive the sensors readings (SM and UV) from sensors

6) Mobile application will access online climate to check for rainfall for the period between two successive irrigations

7) If no rainfall within the specified period, call NPI algorithm, else call PI algorithm

8) Ask for user approval on the irrigation decision (optional in the prototype)

9) Register all activities in audit trail file

10) Send messages to all trustee group in the chain (utilize blockchain to preserve data integrity)

The prototype provides friendly interface, where user could choose one of four different ways to send command to Arduino (Terminal, Arrow keys, Button and Slider, and Voice Control).

6. Conclusions.

In this study, SMCSIS was developed to overcome the excessive irrigation problem (caused by precipitation) and to improve water consumption. In SMCSIS, estimated evaporation was calculated using FFNN, which was trained in using five features (temperature, humidity, wind speed, direction, and UV) collected from online weather forecasting. EE_{avr} was then used to calculate ESM at time T. The performance of FFNN was measured using RMSE (∼0.0890). Despite the weather forecast error, FFNN was proven to be efficient in estimating evaporation on the basis of weather prediction. ESM was then calculated in accordance with EE_{avr} and K_c coefficient. The experimental results showed that the estimation was comparable to the actual soil moisture, where RMSE was ∼0.39. Real-time watering decision was made on the basis of ESM at the time of precipitation in rainy days and the time table of soil moisture sensor read in non-rainy days.

In this study, real-time watering decision was made on the basis of an estimate of soil moisture and probability of precipitation for the coming hours. Two algorithms were proposed to control the irrigation process and make the irrigation decision. An algorithm was initially proposed to determine the next watering schedule (time) in accordance with ESM, which was then used to make the watering decision on non-rainy days. Another algorithm was proposed to control the irrigation process, taking account of rainfall during the coming hours to overcome excessive irrigation. A prototype for controlling the irrigation process of the 100 m^2 potato farm was developed to test and evaluate the performance of the proposed irrigation system. The experimental results proved the effectiveness of the proposed system in saving water and avoiding excessive irrigation.

Acknowledgment. This work is partially supported by Isra University. We would like also to acknowledge the technical help provided by Mr. Khader Saket and Engineer Muath Khaled.

REFERENCES

[1] J. R. dela Cruz, R. G. Baldovino, A. A. Bandala and E. P. Dadios, Water usage optimization of smart farm automated irrigation system using artificial neural network, *The 5th International Conference on Information and Communication Technology*, Malacca, pp.1-5, 2017.

[2] A. Goap, D. Sharma, A. K. Shukla and R. Krishna, An IoT based smart irrigation management system using machine learning and open source technologies, *Computers and Electronics in Agriculture*, vol.155, pp.41-49, 2018.

[3] R. Torres-Sanchez, H. Navarro-Hellin, A. Guillamon-Frutos, R. San-Segundo, M. C. Ruiz-Abellón and R. Domingo-Miguel, A decision support system for irrigation management: Analysis and implementation of different learning, *Water (MDPI)*, vol.12, no.2, p.548, 2020.

[4] M. S. Mekala and P. Viswanathan, A survey: Smart agriculture IoT with cloud computing, *International Conference on Microelectronic Devices, Circuits & Systems (ICMDCS)*, Vellore, pp.1-7, 2017.

[5] S. Salvi et al., Cloud based data analysis and monitoring of smart multi-level irrigation system using IoT, *International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, Palladam, pp.752-757, 2017.

[6] F. F. Montesano, M. W. van Iersel, F. Boari, V. Cantore, G. D'Amato and A. Parente, Sensor-based irrigation management of soilless basil using a new smart irrigation system: Effects of set-point on plant physiological responses and crop performance, *Agricultural Water Management*, vol.203, pp.20-29, 2018.

[7] A. Goldstein, L. Fink, A. Meitin, S. Bohadana, O. Lutenberg and G. Ravid, Applying machine learning on sensor data for irrigation recommendations: Revealing the agronomist's tacit knowledge, *Precision Agriculture*, vol.19, pp.421-444, 2017.

[8] A. Vij, S. Vijendra, A. Jain, S. Bajaj, A. Bassi and A. Sharma, IoT and machine learning approaches for automation of farm irrigation system, *Procedia Computer Science*, vol.167, pp.1250-1257, 2020.

[9] C. Patil, S. Aghav, S. Sangale, S. Patil and J. Aher, Smart irrigation using decision tree, in *International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications. Advances in Intelligent Systems and Computing*, V. K. Gunjan and J. M. Zurada (eds.), Singapore, Springer, 2021.

[10] K. Singh, S. Jain, V. Andhra and S. Sharma, IoT based approach for smart irrigation system suited to multiple crop cultivation, *International Journal of Engineering Research and Technology*, vol.12, no.3, pp.357-363, 2019.

[11] N. A. M. Leh, M. S. A. M. Kamaldin, Z. Muhammad and N. A. Kamarzaman, Smart irrigation system using Internet of Things, *International Conference on System Engineering and Technology (ICSET)*, Shah Alam, Malaysia, pp.96-101, 2019.

[12] P. N. Crisnapati, I. N. K. Wardana, I. K. A. A. Aryanto and A. Hermawan, Hommons: Hydroponic management and monitoring system for an IoT based NFT farm using web technology, *International Conference on Cyber and IT Service Management*, Denpasar, pp.1-6, 2017.

[13] S. Rawal, IOT based smart irrigation system, *International Journal of Computer Applications*, vol.159, no.8, pp.7-11, 2017.

[14] C. Kamienski et al., SWAMP: An IoT-based smart water management platform for precision irrigation in agriculture, *Global Internet of Things Summit (GIoTS)*, Bilbao, pp.1-6, 2018.

[15] M. Li, R. Sui, Y. Meng and H. Yan, A real-time fuzzy decision support system for alfalfa irrigation, *Computers and Electronics in Agriculture*, vol.163, 2019.

[16] B. Alomar and A. Alazzam, A smart irrigation system using IoT and fuzzy logic controller, *HCT Information Technology Trends (ITT)*, Dubai, pp.175-179, 2018.

[17] R. S. Krishnan, E. G. Julie, Y. H. Robinson, S. Raja, R. Kumar, P. H. Thong and L. H. Son, Fuzzy logic based smart irrigation system using Internet of Things, *Journal of Cleaner Production*, vol.252, 2020.

[18] M. A. Khan and K. Salah, IoT security: Review, blockchain solutions, and open challenges, *Future Generation Computer Systems*, vol.82, pp.395-411, 2018.

[19] H. F. Atlam and G. Wills, IoT security, privacy, safety and ethics, in *Digital Twin Technologies and Smart Cities. Internet of Things (Technology, Communications and Computing)*, M. Farsi, A. Daneshkhah, A. Hosseinian-Far and H. Jahankhani (eds.), Springer, 2020.

[20] H. F. Atlam, A. Alenezi, M. O. Alassafi and G. Wills, Blockchain of things: Benefits, challenges and future directions, *International Journal of Intelligent Systems and Applications*, vol.10, pp.40-48, 2018.

[21] B. Bordel, D. Martin, R. Alcarria and T. Robles, A blockchain-based water control system for the automatic management of irrigation communities, *International Conference on Consumer Electronics (ICCE)*, Las Vegas, USA, pp.1-2, 2019.
[22] M. S. Munir, I. S. Bajwa and S. M. Cheema, An intelligent and secure smart watering system using fuzzy logic and blockchain, *Computers & Electrical Engineering*, vol.77, pp.109-119, 2019.
[23] N. Min-Allah, Energy efficient scheduling of tasks in IoT architecture, *ICIC Express Letters, Part B: Applications*, vol.10, no.9, pp.833-840, 2019.
[24] L. García, L. Parra, J. M. Jimenez, J. Lloret and P. Lorenz, IoT-based smart irrigation systems: An overview on the recent trends on sensors and IoT systems for irrigation in precision agriculture, *Sensors*, vol.20, no.2, 2020.
[25] H. Shalini and C. V. Aravinda, An IoT-based predictive analytics for estimation of rainfall for irrigation, in *Advances in Artificial Intelligence and Data Engineering. Advances in Intelligent Systems and Computing*, N. Chiplunkar and T. Fukao (eds.), Springer, 2021.
[26] Q. Adeyemi, I. Grove, S. Peets, Y. Domun and T. Norton, Dynamic neural network modeling of soil moisture content for predictive irrigation scheduling, *Sensors*, vol.18, no.10, 2018.
[27] N. K. Nawandar and V. R. Satpute, IoT based low cost and intelligent module for smart irrigation system volume, *Computers and Electronics in Agriculture*, vol.162, pp.979-990, 2019.
[28] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, The MIT Press, 2016.
[29] S. A. Jaishetty and R. Patil, IoT sensor network based approach for agricultural field monitoring and control, *IJRET: International Journal of Research in Engineering and Technology*, vol.5, no.6, pp.45-48, 2016.
[30] R. G. Allen, L. S. Pereira, D. Raes and M. Smith, Crop evapotranspiration – Guidelines for computing crop water requirements, *FAO – Food and Agriculture Organization of the United Nations*, Rome, Italy, 1998.
[31] B. Duncan and M. Whittington, Cloud cyber-security: Empowering the audit trail, *International Journal on Advances in Security*, vol.9, nos.3-4, pp.169-183, 2016.
[32] A. Sutton and R. Samavi, Blockchain enabled privacy audit logs, *The Semantic Web – ISWC 2017. ISWC 2017. Lecture Notes in Computer Science*, C. d'Amato et al. (eds.), Cham, Springer, 2017.
[33] M. Kocakulak and I. Butun, An overview of wireless sensor networks towards Internet of Things, *Annual Computing and Communication Workshop and Conference (CCWC)*, pp.1-6, 2017.