This is the peer-reviewed version of the following article:


Available online:  https://doi.org/10.1109/JSEN.2020.3012511
Smart Sensing-enabled Decision Support System for Water Scheduling in Orange Orchard

Rahim Khan, Mian Ahmad Jan (Senior Member, IEEE), Muhammad Zakarya (Member, IEEE), Venki Balasubramaniam (Senior Member, IEEE), Varun G Menon (Senior Member, IEEE)

Abstract—The scarcity of water resources throughout the world demands its optimum utilization in various sectors. Smart Sensing-enabled irrigation management systems are the ideal solutions to ensure the optimum utilization of water resources in the agriculture sector. This paper presents a wireless sensor network-based Decision Support System (DSS) for developing a need-based irrigation schedule for the orange orchard. For efficient monitoring of various in-field parameters, our proposed approach uses the latest smart sensing technology such as soil moisture, leaf-wetness, temperature and humidity. The proposed smart sensing-enabled test-bed was deployed in the orange orchard of our institute for approximately one year and successfully adjusted its irrigation schedule according to the needs and demands of the plants. Moreover, a modified Longest Common SubSequence (LCSS) mechanism is integrated with the proposed DSS for distinguishing multi-valued noise from the abrupt changing scenarios. To resolve the concurrent communication problem of two or more wasp-mote sensor boards with a common receiver, an enhanced RTS/CTS handshake mechanism is presented. Our proposed DSS compares the most recently refined data with pre-defined threshold values for efficient water management in the orchard. Irrigation activity is scheduled if water deficit criterion is met and the farmer is informed accordingly. We have developed a user-friendly Graphical User Interface (GUI) for the proposed smart sensing-enabled test-bed. Both the experimental and simulation results show that the proposed scheme performs better in comparison to the existing schemes.

Index Terms—Wireless Sensor Network, Precision Agriculture, Irrigation Management Systems, DSS, RTS/CTS, LCSS

I. INTRODUCTION

WORLDWIDE, water is a scarce resource and it requires considerable attention from the research community and industry to ensure its maximum utilization. Agriculture sector is one the water’s main consumer because it consumes approximately 70% of the available water to fulfill the food requirements of the world fast growing population [1]. Generally, irrigation schedules are based on farmers experience, crop requirements, environmental conditions, and soil properties. However, these traditional irrigation procedures are not efficient from the resource utilization perspective as a considerable amount of water is wasted. Due to the recent technological advancements, particularly sensors and actuators, it is possible to develop a smart sensing-enabled automated Decision Support System (DSS) that has the ability to identify water-deficit locations and irrigate those areas on priority basis if needed [2].

A network of smart sensing devices has the potentials to collect the real-time data for developing an automated Irrigation Management System (IMS) or DSS, that is also known as precision agriculture [3]. The DSS aims to provide the right resources at the right time, which has a direct correlation with the yield improvement of various crops. To realize this objective, smart sensing devices are deployed in agricultural fields where specialized sensors probe their surrounding phenomena such as soil moisture, soil temperature, pH, humidity, and leaf wetness, etc. These gathered phenomena are thoroughly observed by the DSS on a centralized device. Various agriculture-related activities are subject to these observations. There exist numerous studies in this context. In [4], a WSN-based remote water management system for agriculture sector was implemented in Thailand. The proposed approach was deployed for five months in agricultural fields and various data mining techniques were used to analyze the captured data. However, the proposed approach completely neglected outliers, i.e., noisy data. A DSS-enabled irrigation prediction system was presented in [5] for optimizing the water scheduling of orange orchard. This system has the ability to predict the soil moisture of a particular region within the orchard by applying a hybrid of Support Vector Regression (SVR) and K-mean clustering algorithm on the gathered data. The proposed DSS emphasized on the refinement of captured data before it is being processed by the DSS. However, this system fails to distinguish multivariate noise from abrupt changing scenarios [6]. In [7], a WSN-enabled water management system was deployed in adjacent agricultural fields where different crops were grown. The proposed test-bed utilized both the historical data and variations in the climate values to devise an effective irrigation schedule. An Internet of Things (IoT) and neural network-based DSS was developed in [8] to predict the water-deficit locations. The proposed DSS is capable to precisely detect the required amount of water for irrigation. However, this approach does not consider the outliers in the gathered data.

In precision agriculture, outliers or noise is defined as unwanted data that severely affect the performance of an operational DSS. Usually, this problem occurs due to malfunctioning sensor nodes, interference, collision of packets, circuit failure, extreme pressure, high temperature, and other
environmental conditions. As a result, refinement of sensed data prior to its processing by the concern DSS is a challenging issue. Moreover, in the case of shared media, collision of Request To Send / Clear To Send (RTS/CTS) packets and data packets is another challenging issue. Therefore, the development of a precise and accurate technology-assisted DSS is desperately needed to ensure maximum utilization of water in agriculture sector.

In this paper, a smart sensing-enabled DSS for the orange orchard is presented to resolve the aforementioned issues. In our proposed approach, the sensed data by the various smart sensing devices/nodes is processed using a refinement module to ensure accuracy and integrity of data at the destination. Moreover, every node is bounded to transmit its data only if the medium of communication is free. The main contributions of this paper are:

1) A smart sensing-enabled DSS is presented for proper management of the irrigation activities in agriculture sector. The proposed agricultural DSS is a need-based system that provides water to a particular area only if it is identified as water-deficit.

2) A modified LCSS mechanism is proposed that enables the proposed DSS to differentiate multivariate noise from the abrupt changing scenario.

3) An enhanced RTS/CTS mechanism is proposed to resolve the collision issue associated with concurrent communications. Before any transmission activity, every sensor node is bounded to use the classifier-based mechanism that ensures a collision-free communication between two or more operational devices.

4) A simplified Graphical User Interface (GUI) is developed for the proposed smart sensing-enabled DSS that presents the captured and refined data of sensor nodes in both textual and graphical formats.

The rest of the paper is organized as follows. In Section II, an overview of the literature is presented. In Section III, a detailed description of the proposed smart sensing-enabled DSS module and its deployment for orange orchard are presented. In Section IV, both experimental and simulation results are discussed in detail. Finally, concluding remarks are provided in Section V.

II. LITERATURE REVIEW

Precision agriculture is the technology-assisted farming, which is based on sensor-enabled monitoring, measurement and response generation via the DSS. The responses are generated based on the varying conditions of crops [9]. It enables the farmer to provide the right resources at the right time and right place to any crop [10]. In agriculture, water is an essential resource that is needed to bring forth the maximum potential of the agricultural fields. Moreover, it enables crops to make full use of other yield enhancing production factors [11]. In this section, a brief overview of various test-beds which are related to the proposed work is presented.

B. Keswani et al. [8] have presented an optimal Internet of things (IoTs) and neural network-based irrigation management system that has a one-hour prior prediction capability of water-deficit location(s). For this purpose, various sensors were deployed such as soil moisture, temperature, CO₂, light intensity, and humidity sensors. A hybrid DSS was proposed by Viani et al. [12] which was based on the fuzzy logic and farmer's experiences. Likewise, WSNs and General Packet Radio Services (GPRS) were used to form an optimal irrigation management system. Soil moisture sensors with controller modules were deployed in agricultural field(s) [13]. A machine learning and agronomist's encysted knowledge-based irrigation prediction system was proposed that concluded that Gradient Boosted Regression Trees (GBRT) was the best regression model with approximately 93% irrigation prediction accuracy [14]. Dursun et al. [15] presented an automatic drip irrigation management system for the cherry trees. A low-cost IoT system for smart irrigation was proposed by Nakawandy et al. [16]. The system capabilities include the estimation of irrigation schedule, neural-based decisions and remote monitoring. Similarly, a WSN-based irrigation management platform was presented that has the capacity to calculate the quantity of water needed for irrigating a specific area [17]. A novel watering management system, which is based on low-cost IoT components was presented by TA Khoo et al. [18]. Additionally, LoRa LPWAN technology was used for the transmission to ensure the best performance of the proposed system. In FLOW-AID project, WSNs were used to identify water-deficit locations, a situation where plants need water desperately [19]. In 2011, the Information and Communication Technology unit of Commonwealth Scientific and Industrial Research Organization (CSIRO) used various sensor nodes to recover the ecological integrity of Queensland’s National Park [20]. Pardossi et al. [21] described a methodology of integrating rote zone sensors with WSNs which is used to identify water deficit locations in agricultural fields. An Harun et al. [22] described WSNs as an efficient tool to resolve both the decision-making and resource optimization issues associated with technology assisted farming. A need based irrigation practice was presented by O Abrishambaf et al. [23]. This system has the capacity to schedule the irrigation activity for lowest cost period by using various parameters to temperature, soil moisture, wind, precipitation forecast, and soil evapotranspiration calculation. IoTs based irrigation system was developed to automate the irrigation activity of crops using soil and environmental data [24]. X. Dong et al. [25] presented a pivot based irrigation procedure to optimize the irrigation activity via wireless underground sensor networks.

III. PROPOSED LCSS-BASED DATA REFINEMENT MECHANISM

Data fusion or refinement of the WSNs capture data has become a dominant research area; as the majority of our daily-life activities are either partially or completely dependent on these networks, based on DSS. In this section, a space free LCSS-based data fusion scheme is presented to enhance accuracy and precision level of the proposed agricultural DSS. The captured data of every device $C_i$, that is waspmote agricultural board in the proposed test-bed, is passed through the LCSS based noise detection module which ensures
accuracy of the refined data. Moreover, the proposed fusion scheme has the capacity to distinguish outliers or noisy data from the abrupt change scenarios, i.e., abrupt change occurs if water directly interacts with soil moisture or leaf wetness sensors.

A. Sequence Matching: Definitions and Preliminaries

A sequence \( SQ_a \) is defined as a collection of related values, \( a_1, a_2, ..., a_n \in SQ_a \) where \( n \) represents length of the data set. The longest common subsequence (LCSS) is represented by \( LCSS(k) \) where \( k \) defines length of the LCSS. Concatenation process in LCSS is defined as appending any two symbols \( X \) and \( Y \) to form a subsequence \( XY \) such that \( X \& Y \in SQ_a \).

**definition-I** Any two values \( X \in SQ_a \) and \( Y \in SQ_b \) are considered as equal iff distance \((X,Y)\) \(\leq 0.05 \) or \( X \equiv Y \).

**definitions** LCSS(0,0) is used to describe an empty LCSS.

**definition-II** A value \( a_1 \in SQ_a \) is considered as a predecessor of another value \( a_2 \in SQ_a \) in LCSS iff index \((a_2 > a_1)\) and for both values \( \exists \) (a value \( b_m \in SQ_b \) such that \( a_1 \equiv b_m \) and \( a_2 \equiv b_m \) ).

**definition-III** A value \( a_1 \in SQ_a \) is not considered as a predecessor of another value \( a_2 \in SQ_a \), that is \( a_1 \notin LCSS_{matched} \), iff index \((a_2 < a_1)\) that is \( 1 > 2 \). Although, for both values \( a_1, a_2 \in SQ_a \) \( \exists (b, b_2 \in SQ_b \) such that \( a_1 \equiv b, a_2 \equiv b \ \forall \) where \( y \) and \( z \) represent indexes information.

**definition-IV** Similarity index of any two sequences or data sets \( SQ_a \& SQ_b \) are higher iff length of their \( LCSS_{matched} > length(SQ_{a_{min}}/SQ_{b_{min}}) \)

**definition-V** The \( LCSS(k) \) represents the LCSS of \( SQ_a \) and \( SQ_b \) iff \( V(a_n \in SQ_a \) where \( \exists a_m \in SQ_b \) such that \( a_n \equiv b_m \) and \( \exists (n' < n \ and \ m' < m \) such that \( LCSS(k-n', l-m') \) is generated by \( n' \) and \( m' \).

B. Computation of the LCSS

The proposed approach uses two different sequences of the same size \( n \) that is 10 in this case i.e., \( SQ_a \) for storing current data values and \( SQ_b \) for previously transmitted data where \( a = 1, 2, 3...n \) and \( b = 1, 2, 3...m \). Every device \( C_i \in WSN \) is bounded to store the collect data in \( SQ_a \) until \( a = n \). Initial values for \( SQ_a \) is defined manually, once at the deployment stage of WSNs, and are updated according to collected data of sensor(s). For example, soil moisture sensor values are set according to the average values of three different soil moisture sensors which were deployed in dried soil i.e., 250Hz to 260Hz. Once, the network becomes fully operational i.e., sensors begin to probe the phenomena after the defined interval of time, that is 30 second in the deployed WSN’s infrastructure.

In the proposed test-bed, every wasp-mote board \( C_i \) is bounded to store their captured data temporarily in sequence \( SQ_a \) until value of \( a = 10 \) and then send it to the gateway.

To refine this data, the proposed test-bed uses a modified form of LCSS and gap-free LCSS. LCSS is used to find the similarity indexes of the currently received data \( SQ_a \) and existing data \( SQ_b \). Initially, a matching window control parameter \( \delta \) is defined that is used to limit the matching window of a value in sequence \( SQ_a \), i.e., \( a = 1 \), with another sequence \( SQ_b \). In the proposed test-bed, the value of \( \delta \) is set to three (3) which means that the first element of \( SQ_a \) is matched with at-most three elements of \( SQ_b \) if these elements are not matched. In phase-I, the first element of \( SQ_a \), i.e., \( SQ_1 \in SQ_a \), is matched with element(s) of \( SQ_b \), i.e., \( b_1, b_2, ..., b_\delta \in SQ_b \), such that either a match is found or maximum limit \( \delta \) is reached. If first element \( a_1 \in SQ_a \) matches with any element \( b_1 \& \delta \in SQ_b \) then \( b_m \) is stored in class \( LCSS_{matched} \) with its position information. However, if \( a_1 \) does not match with any element of \( SQ_b \) within the defined window \( \delta \) then \( a_1 \) is ignored and subsequent element \( a_2 \in SQ_a \) is processed. Likewise, second value \( a_2 \in SQ_a \) is matched using similar approach with a slight modification that is its matching criteria with the \( SQ_b \) is subjected to the following conditions.

1) \( a_2 \in SQ_a \) is matched with the first value of \( SQ_b \) iff \( a_1 \in SQ_a \) does not have a matching value in the defined window, i.e., \( \delta \).

2) \( a_2 \in SQ_a \) is matched with value of \( SQ_b \) that is stored after previously matched value i.e., \( b_1 \in SQ_b \) \( a_1 \in SQ_a \equiv b_1 \in SQ_b \).

If \( a_2 \in SQ_a \) has a match in \( SQ_b \) then matching value \( b_{2-\delta} \in SQ_b \) is stored in class \( LCSS_{matched} \) with its position information. For the remaining values of \( SQ_a \), this process is repeatedly applied to compute their LCSS.

In phase-II, first value of \( SQ_a \), i.e., \( a_1 \in SQ_a \), is ignored iff \( a_1 \in LCSS_{matched} \) and the required LCSS is not computed yet. The remaining values, i.e., \( a_2, a_3, ..., a_n \in SQ_a \), is considered as a refined data set which has nine values. Then, the aforementioned process, i.e., finding \( LCSS_{matched} \) of \( SQ_a \) and \( SQ_b \), is repeated.

Both LCSSs, i.e., current \( LCSS_{matched} \) and previous \( LCSS_{matched} \), are compared and LCSS with the maximum length is selected whereas other is discarded. This process is repeated until the required LCSS.

To understand this idea, consider two sequences \( SQ_a \) and \( SQ_b \) which contain data generated by the temperature sensor(s) i.e., \( SS_n = 30 \ 34 \ 31 \ 30 \ 33 \ 35 \ 34 \ 30 \ 34 \ 32 \) and \( SQ_m = 33 \ 30 \ 32 \ 34 \ 30 \ 33 \ 34 \ 30 \ 34 \ 32 \) where \( n = m = 10 \) and \( \delta = 3 \). First value \( 30 \in SQ_a \) is matched with every value of \( SQ_b \) within the defined window \( \delta = 3 \); starting with the first, i.e., 32 \( \in SQ_b \). A match is encountered at the \( 2^{nd} \) position in \( SQ_b \), i.e., \( 30 = 30 \). Value 30 is stored in \( LCSS_{matched} \) with its position information. Second value 34 \( \in SQ_a \) is then matched with every value of \( SQ_b \) starting from the position \( 3^{rd} \) i.e., 31 in this case. However, 31 does not have a matching value in \( SQ_b \) within \( \delta \). Therefore, it is neglected and the subsequent value \( 31 \in SQ_b \) is processed which is matched with \( 3^{rd} \) value in \( SQ_b \). 31 is stored with its location information in \( LCSS_{matched} \). For the remaining values of \( SQ_a \), this process is repeatedly applied until their LCSS is found. It is to be noted that phase-II is applicable only if the computed LCSS length is less than the length of \( SQ_a/2 \).

**Lemma 1.** \( LCSS(p) \) represents the longest common subsequence of \( SQ_a,SQ_b \) of length \( n \) and \( m \) respectively iff \( k = 1 \) and \( \exists (a \cdot p \equiv b_{1-p} \cdot a_{1-p} \in SQ_a \ and \ b_{1-p} \in SQ_b) \) where \( p \leq n \) and \( m \).

*Proof.* Applying mathematical induction i.e., for \( k = 1 \) The \( length(LCSS(1)) \) is equal to 1 iff \( a_1 \equiv b_1 \) where
\[ a_1 \in S_{Q_a} \text{ and } b_1 \in S_{Q_b}. \text{ (According to Definition-VI).} \]

Hence, the lemma is true for \( k = 1 \).

Suppose that the lemma is true for \( k - 1 \) values. We need to prove that the lemma is true for \( k \) values. If \( \text{LCSS}(n,m) \) represents the \( \text{LCSS} \) of \( S_{Q_a} \& S_{Q_b} \) then \( \exists (n < n \text{ and } m < m \text{ such that } \text{LCSS}(n,m) = \text{LCSS}(n',m') \text{ for } k - 1 \text{ value}). \)

According to our assumption,
\[ \text{LCSS}(n,m) = p_1, p_2, p_3, \ldots, p_k \text{ such that } p' < p \text{ and } p_1, p_2, p_3, \ldots, p_k \in S_{Q_a} \& S_{Q_b} : a_n = b_m : n < n \text{ and } m < m. \]

Therefore, \( \text{LCSS}(S_{Q_a} \& S_{Q_b}) \) is of length \( k \).
\[ \therefore \text{length}(\text{LCSS}(n,m)) + \text{length}(\text{LCSS}(n,m)) = k. \]

Hence, it proves that \( k \) is the length of required (\( \text{LCSS}(p) \)).

Conversely, if \( \text{length}(\text{LCSS}(n,m)) \geq k \) and \( a_n = b_m \), where \( a_n \in S_{Q_a} \) and \( b_m \in S_{Q_b} \), then \( \exists (n' < n \text{ and } m' < m \text{ such that } a_n' = b_m'). \)

Moreover, \( \text{length}(\text{LCSS}(n,m)) = \text{length}(\text{LCSS}(n,m)) - 1 \geq k - 1. \) Therefore, \( \text{LCSS}(n,m) \) is the \( \text{LCSS} \) of \( k - 1 \) length data sets (By inductive Hypothesis).

Therefore, the proof i.e., \( \text{LCSS}(p) \) represents the longest common subsequence of \( S_{Q_a} \& S_{Q_b}. \)

\[ \square \]

C. Proposed Methodology: Classifier-based based RTS/CTS Handshake

To resolve one of the aforementioned issue, i.e., collision of RTS/CTS packets, a classifier-based scheduled RTS/CTS mechanism is presented. Every device \( C_i \in \text{IoT}s \) shares its communication schedule \( T_s \) with the neighboring devices via a smaller scheduled-frame preferably after the deployment phase. The proposed communication scheme consists of two phases i.e., hop-count and classifier-based optimal neighbors discovery phases.

1) Hop-count Discovery Phase: The base station module \( S_j \) broadcasts a scheduled-frame which contains a transmission schedule \( (T_s) \), hop-count \( (H_c) \) and back-off timer \( T_b \). Moreover, the hop-count value is set to zero as base station is the ultimate destination for every device \( C_i \in \text{IoT}s \) and \( T_b \) value is set to infinity which distinguishes \( S_j \) from the ordinary devices. Active devices \( C_i \) which reside in the closed proximity of \( S_j \) receive this frame and update it according to their stored information, i.e., \( H_c = 1 \), \( T_b \) and \( T_s \) are set according to the equation 2 & 3 respectively. Moreover, every device \( C_i \) maintains a schedule table where valuable information about neighboring nodes is stored i.e., \( H_c \), \( T_s \), residual energy \( E_r \) that is calculated using equation 1.

\[ E_r = E_i - E_c \]  \hspace{1cm} (1)

where \( E_i \) and \( E_c \) represent the initial and consumed energies respectively.

Back-off timer \( T_b \) is computed using equation 2. The idea of adding an \( H_c \) value or \( \delta \) with the generated random number is to minimize the collision probability of neighboring nodes as usually, these nodes have different \( H_c \) values. However, if back-off timer \( T_b \) of the two neighboring devices are similar then these devices should recompute their \( T_b \), \( \delta \) is an infrastructure dependent parameter i.e., for flat networks its value ranges from 5-15 whereas in hierarchical networks its value ranges from 2-5.

\[ T_b(C_i) = \text{rand}(0 - 1000) + \min(H_c(C_i), \delta) \]  \hspace{1cm} (2)

Transmission schedule \( T_s \) is computed using equation 3.

\[ T_s(C_i) = T_b + T_r + \gamma \]  \hspace{1cm} (3)

where \( T_r \) is the average propagation time of \( C_i \)'s first hop neighbors which includes both the transmission and processing delays. \( \gamma \) represents the sampling rate of a particular device which will be similar for every \( C_i \in \text{WSNs} \).

Once a first hop neighboring node \( C_i \) updates the scheduled-frame, it doesn’t broadcast the frame immediately rather it waits for \( T_b \). When \( T_b \) expires, \( C_i \) broadcasts an updated version of the scheduled-frame which is received by devices reside in vicinity. \( C_i \)'s neighboring devices are divided into two groups i.e., Group-I which consists of devices such that their \( H_c <= H_c(C_i) \) whereas Group-II has devices with \( H_c > H_c(C_i) \). When a device \( C_{i+1} \in \text{Group - I} \) receives a scheduled-frame from a neighboring device \( C_i \) it updates the schedule table entries according to the message contents and discard it. \( C_{i+1} \) discards the received scheduled-frame because it has either transmitted a scheduled-frame or waiting for its \( T_b \) to expires as it has already received a similar message from the base station module \( S_j \). Conversely, if the scheduled-frame is received by a device \( C_{i+2} \in \text{group - II} \) then it updates the scheduled table information particularly about \( C_i \) such as \( H_c \), \( T_b \) and \( T_s \). Moreover, \( C_{i+2} \) computes its back-off timer using equation 2 and updates the scheduled-frame by replacing \( H_c \), \( T_b \) and schedule time \( T_s \) with its own. When \( T_b \) of \( C_{i+2} \) expires it broadcasts the updated scheduled-frame. This process is repeatedly applied until every device \( C_i \in \text{WSNs} \) in an operational network has a defined \( H_c \) value and information about neighboring node’s transmission schedules \( T_s \). Additionally, \( C_i \)'s transmission schedule is not affected even if it serves as a relaying device, that is forwarding packets of neighboring devices, in addition to its own duties.

2) Classifier-based Mechanism to Mitigate the Collisions Ratio of RTS/CTS and Data: In the proposed scheme, every device \( C_i \) maintains a schedule table which contains information about neighboring devices. This information is very useful in both scenarios i.e., minimizing the collision probability and finding an optimal device.

1) Where multiple devices initiate a request-to-send message (RTS) at the same time and are interested to start a communication process with a shared device i.e., base station or cluster head (CH) or neighboring node.

2) Where a single device \( C_i \) has multiple recipient and needs to start communication with a reliable and optimal device.

In scenario-I, without a proper schedule information of neighboring devices, particularly first hop neighbors, collisions will occur and retransmission is mandatory which is not only time-consuming but power consuming too. However, if these devices are bounded to store sufficient information about neighboring devices such as \( T_s \), \( T_b \) and \( H_c \) then packets collisions are minimized or even avoided. The proposed scheme
uses a classifier-based mechanism to resolve the collision issue associated with devices interested in communication with a shared base station or other entity. Since, every neighboring device \( C_i \) has a unique back-off timer \( T_b \), hence, the collision probability is zero even if two neighboring devices initiate the RTS process simultaneously.

In scenario-II, if a device \( C_i \) is interested to initiate a communication session with a reliable and optimal neighboring device or CH or base station then this device needs a simplified classification mechanism which identifies an optimal device. The proposed classifier-based mechanism uses various parameters such as \( T_s, T_b, E_r \) and \( H_c \) values to find an optimal neighbor. Neighboring devices are classified using equation 4.

\[
C_{opt} = (W_1 * T_b + w_2 * T_s)C_i
\]  

where \( W_1 = 50\% \), \( W_2 = 50\% \) represent different weight-ages assigned to these parameters. A neighboring device \( C_i \) with minimum value of \( C_{opt} \) is an ideal and reliable candidate. However, if \( H_c \) and \( E_r \) of neighboring devices are not considered by our classifier then it is possible that either the transmitted packets may propagate in opposite directions or forwards to a device with minimum residual energy. In both cases the results are not favorable specifically in resource limited infrastructures, therefore, once the classifier described in equation 4 identifies the optimal neighbors then the two most optimal devices are passed to another classifier as described in equation 5.

\[
C_{reliable} = W_3 * H_c + W_4 * E_r(C_i)
\]  

where \( W_3 = 40\% \), \( W_4 = 60\% \) are weight-ages assigned to the residual energy and hop-count parameters. A neighboring device \( C_i \) with maximum value of \( C_{reliable} \) is considered as optimal and reliable device.

D. Implementation of the Proposed Scheme in Agricultural Environment: A Case Study

A precise and accurate DSS (preferably technology-assisted) is subjected to the selection of appropriate devices or sensors \( C_i \), parameters to be sensed, data refinement and communication mechanisms. To accomplish this, wasp-mote agricultural boards with a gateway were deployed in the orange orchard of our institute for approximately one year to form an automatic irrigation management system as shown in Fig. 1, Fig. 2 and Fig. 3, respectively. These boards were equipped with soil moisture, temperature, humidity and leaf wetness sensors to collect real-time data continuously after a defined interval of time i.e., 30 seconds.

In the proposed DSS, soil moisture parameter is considered due to its vital role in the development of a precise watering schedule. For example, if the sensed value is below the threshold value, then that particular area is needed to be irrigated on priority basis. To further precise the proposed DSS, soil moisture sensors were deployed at three different levels in the agricultural field, as shown in Fig. 3. Likewise, atmospheric moisture exerts drastic effects on the watering schedules of various crops. Therefore, leaf wetness sensors were deployed in closed proximity to the orange leaves as shown in Fig. 2.
Moreover, temperature and humidity sensors were integrated with the wasp-mote boards to further enhance accuracy of the proposed DSS as shown in Fig. 1. The gateway module is directly connected to a computer via a USB serial port (port-6 in this case) to receive data.

In our previous data collection and communication infrastructure [6], a simplified approach was used to resolve the collision issue associated with simultaneous transmission of two or more devices $C_i$ to a common destination. However, the system enters to deadlock if more than two devices are simultaneously transmitting to a common destination. In this paper, a modified version of the RTS/CTS handshake approach is used to resolve this issue. A detailed description of the proposed WSNs-based DSS for agriculture sector is presented in Fig. 4.

Every wasp-mote board collects real time data from various sensors i.e., temperature, humidity, soil moisture and leaf wetness. This data, say packet-x, is sent to the gateway either directly or through the relaying nodes. In both cases, the transceiver module uses the RTS/CTS handshake approach to avoid collision of packet(s). In the proposed experimental setup, the gateway module is directly connected to a central computer via USB cable specifically through port-6 and the received data is (temporarily) stored automatically using Cool Term software. Before DSS, packet-x is passed through the noise detection module to get the refined data let say packet-y. The DSS module of the proposed system checks packet-y against the threshold value, that is 250Hz for soil moisture sensor, and if the threshold value is crossed then the alarming unit is activated along with a text message to the farmer on his mobile or LAN. If data is within the defined threshold then it is stored permanently.

**IV. EXPERIMENTAL AND SIMULATION RESULTS**

In this section, a detail description of both experimental and simulation results are presented to verify the exceptional performance of the proposed system against the existing schemes in terms of computational time, decisions accuracy, packet collision ratio and packet loss ratio. These algorithms were implemented in OMNET++, which is an open source simulation tool specifically designed for the resource limited networks. Initially, a static topological infrastructure, which was later on changed to the random, with a fixed propagation delay was used to mimic the real deployment process of WSNs in general and our deployment infrastructure in particular. Additionally, other networks related parameters such as interference and path-loss ratio were kept constant. A detail description of various simulation related parameters are presented in table I.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>values</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSN Deployment Area</td>
<td>1000m * 1000m</td>
</tr>
<tr>
<td>Sensor Node</td>
<td>50, 100, 500, 1000</td>
</tr>
<tr>
<td>Base Station</td>
<td>One</td>
</tr>
<tr>
<td>Initial Energy ($E_{i}$)</td>
<td>52000 mAh</td>
</tr>
<tr>
<td>Residual Energy ($E_{r}$)</td>
<td>$E_{r} - E_{c}$</td>
</tr>
<tr>
<td>Packet Transmission Power Consumption ($P_{tx}$)</td>
<td>9.14 mW</td>
</tr>
<tr>
<td>Channel Delay ($C_{delay}$)</td>
<td>10 milliseconds</td>
</tr>
<tr>
<td>Packet Receiving Power Consumption ($P_{rx}$)</td>
<td>59.1 mW</td>
</tr>
<tr>
<td>Idle Mode Power Consumption</td>
<td>1.27 mW</td>
</tr>
<tr>
<td>Sleep Mode Power Consumption</td>
<td>15.4 µW</td>
</tr>
<tr>
<td>Transceiver Energy ($T_{c}$)</td>
<td>1 mW</td>
</tr>
<tr>
<td>Transmission Range ($T_{r}$)</td>
<td>500m</td>
</tr>
<tr>
<td>Receiving Power Threshold ($RT_{sn}$)</td>
<td>1024 bits</td>
</tr>
<tr>
<td>Packet Size ($P_{size}$)</td>
<td>128 bytes</td>
</tr>
<tr>
<td>Initial Hop Count ($H_{c}$) of Sensor Nodes</td>
<td>$\infty$</td>
</tr>
<tr>
<td>Maximum Distance between Nodes</td>
<td>300-450m</td>
</tr>
<tr>
<td>Sampling Rate of sensor nodes</td>
<td>10 to 30 seconds</td>
</tr>
<tr>
<td>Topological Infrastructure</td>
<td>Static and Random</td>
</tr>
<tr>
<td>Traffic Type</td>
<td>CBR and UDP</td>
</tr>
</tbody>
</table>

Initially, the real-time data set, that is collected through the deployment of various wasp-mote boards based tested in the orange orchard, is used as a testing tool to check the performance of these algorithms particularly in terms of computational time and decision accuracy of the underlined DSS. In terms of computational cost, the performance of these algorithms is presented in Fig. 5, which clearly depicts that the proposed algorithm performance is better than existing algorithms except the noise evading algorithm. However, a common problem associated with NE algorithm is its vulnerability to multi-valued noise, which is quite common in WSNs. Moreover, NE does not differentiate multi-valued noise from an abrupt change scenario. Similarly, the proposed scheme performance is not affected by changing data set size either statically or dynamically because it always uses a fixed sliding window. Therefore, the proposed algorithm is suitable for both scenarios, i.e., static and dynamic datasets. Additionally, these algorithms were evaluated on different static versions of the real-time dataset obtained through our deployed test bed, that was approximately collected in a month or two. The proposed scheme performance is intact as shown in Fig.5.

In agriculture sector, the farmer’s attraction to the technology...
based infrastructures or DSS will be increased iff majority of their decisions or predictions are accurate. Therefore, the proposed algorithm is eager to improve accuracy of the agricultural DSS with the available computational resources and minimum cost. The decision accuracy of the proposed and existing algorithms based DSS is depicted in Fig. 6 which shows the exceptional performance of the proposed algorithm based DSS than existing algorithms. Moreover, it is evident from Fig. 6 that the NEA based DSS has a high probability of errors or wrong decision(s).

The claims of an algorithm is considered as authentic iff it is tested on publicly available benchmark datasets. Therefore, these algorithms were tested on various publicly available benchmark datasets as shown in Table II. The computational time of the proposed algorithm is less than that of existing algorithms except NE which has other issues as described above. We have observed that the computational time of the proposed scheme is inversely proportional to the similarity indexes of the datasets or matching windows i.e., if similarity index is high the computational time will be small and vice versa. Due to the high similarity indexes of the benchmark datasets B-Cancer and Two Patterns, the computational time of the proposed algorithms is approximately equal to that of NE algorithm as shown in Table II.

Accuracy of the proposed and existing algorithms based DSS on various publicly available benchmark datasets are depicted in Table III. It is evident from Table III that the proposed mechanism is an ideal candidate for the design of an accurate and precise technology based DSS for the agriculture sector.

Packet delivery ratio is the ratio of successfully delivered packets, particularly at the destination module, to the transmitted one in an operational network. We have observed that the proposed scheme has the maximum packet delivery ratio for both real-time and simulated data against its rival schemes as shown in Fig. 7 and Fig. 8; as packet delivery ratio is inversely proportional to the packet loss ratio which is mostly due to the packet collision. In proposed scheme, the collision issue is resolved by utilizing the RTS/CTS handshaking scheme.
automating various activities such as irrigation and monitoring. To attract farmers, the graphical user interface (GUI) of the proposed system were designed in such a way that a naive user can easily use it.

V. CONCLUSION

Smart sensing-enabled networks, such as WSNs, have the ability to predict when and where the irrigation activities need to be performed. These networks enable the farmers to evaluate the required amount of water for irrigation purposes based on the data sensed by various nodes. In this paper, a real-time smart sensing-enabled Decision Support System (DSS) was presented for optimizing the water schedules for orange orchard. Smart sensing-enabled devices were deployed in different regions for approximately one year to collect soil moisture, temperature, humidity and leaf-wetness of the orchard. The gathered raw data were refined by passing it through a noise module for outliers detection. The DSS module matches the refined data against the threshold values using a modified LCSS mechanism. If these data are below the threshold value, e.g., less than 250Hz for the soil moisture sensor, then irrigation activity is scheduled in that region and the farmer is notified via a text message. Moreover, a modified version of the RTS/CTS handshake mechanism was presented to ensure the successful delivery of packets and collision avoidance. Both the experimental and simulation results showed the exceptional performance of our proposed scheme against the existing schemes for outliers detection and results showed the exceptional performance of our proposed collision avoidance. Both the experimental and simulation results showed the exceptional performance of our proposed scheme against the existing schemes for outliers detection and results showed the exceptional performance of our proposed collision avoidance.

To attract farmers, the graphical user interface (GUI) of the proposed system were designed in such a way that a naive user can easily use it.

Fig. 8. Average Packet Delivery Ratio (Experimental Results)

VI. REFERENCES


