Computers and Electronics in Agriculture 171 (2020) 105309

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Multilevel data fusion for the internet of things in smart agriculture

Andrei B.B. Torres, Atslands R. da Rocha*, Ticiana L. Coelho da Silva, José N. de Souza, Rubens S. Gondim

Fortaleza - CE, Brazil

ARTICLE INFO

Keywords:
Data fusion
Internet of things
Smart agriculture

ABSTRACT

The Internet of Things (IoT) aims to enable objects to sense, identify, and analyze the world, but to achieve such goal cost-effectively, it should involve low-cost solutions. That implies a series of limitations, such as small battery life, limited storage capabilities, low accuracy, and imprecise sensors. Data fusion is one of the most widely used methods for improving sensor accuracy and providing a more precise decision. Therefore, we propose Hydra, a multilevel data fusion architecture, to improve sensor accuracy, identify application target events, and make more accurate decisions. Hydra is composed of three layers: low-level (sensor data fusion), medium-level (events and decision making), and high-level (decision fusion based on multiple applications). In partnership with Embrapa (Brazilian Agricultural Research Corporation), we instantiated Hydra for the smart agriculture domain, and we also developed two applications aiming smart water management. The first application goal was to determine the need for irrigation based on soil moisture levels, and the second ascertained the adequate irrigation time by estimating the crop's evapotranspiration (rate of water evaporation by the soil and transpiration by plants). We performed a set of experiments to assess Hydra: (i) evaluation of methods to detect and remove outliers; (ii) analyze data resulting from the applications; (iii) the use of machine learning to create a new accurate evapotranspiration model based on the sensors data. The results indicate that a combination of the ESD method (Extreme Studentized Deviate) and WRKF filter (Weighted Outlier-Robust Kalman Filter) was the best method to identify and remove outliers. Moreover, we generated an evapotranspiration model using the SVM (Support Machine Vector) quadratic machine-learning model that produced values close to the evapotranspiration reference model (Penman-Monteith).

1. Introduction

Agribusiness has great importance in the Brazilian economy, representing a large amount of the Gross Domestic Product. Irrigated agriculture plays a vital role in agribusiness by ensuring innovative approaches that lead to increased productivity and provide sustainable solutions. According to National Water Agency (ANA) (2017), irrigation in agriculture is responsible for 67.2% of freshwater consumed in Brazil, and causing a water withdrawal of 969.0 m^3/s. Furthermore, Brazilian producers usually withstand long periods of drought, causing the process of agricultural production to be even more difficult and more expensive. Therefore, using water resources in a planned and controlled manner is vital.

The Internet of Things (IoT) can be a viable approach to address the issue of agricultural monitoring and control. Wireless Sensors and Actuators Networks (WSAN), frequently used as an infrastructure of IoT systems, can be used to monitor crops and determine the optimum irrigation timing. At the same time, intelligent objects can inform harvest storage conditions. Both pieces of information can be merged and help in making crop-related decisions. Thus, it can advise the farms about the exact amount of water to use in the irrigation process, avoiding wasting such a valuable resource due to poorly executed irrigation and lack of control.

A viable and cost-effective IoT solution, especially for outdoor environments like agriculture, involves low-cost objects, which implies a series of limitations such as small batteries, low processing, and storage capabilities. Moreover, it requires dealing with the imprecision of data sensors and low accuracy due to the environment, faulty sensors, communication problems, and noise. Data fusion (also known as information fusion) can help solve those issues, as it is one of the most widely used methods for improving sensor accuracy, representing the big picture, and providing more precise decisions (Adamchuk et al., 2004).

Various terms, such as data fusion, sensor fusion, and information fusion, are used to describe some fusion aspect. Nevertheless, the terms

* Corresponding author.
E-mail addresses: andreibosco@virtual.ufc.br (A.B.B. Torres), atslands@ufc.br (A.R. da Rocha), ticianalc@virtual.ufc.br (T.L. Coelho da Silva), neuman@ufc.br (J.N. de Souza), rubens.gondim@embrapa.br (R.S. Gondim).

https://doi.org/10.1016/j.compag.2020.105309
Received 9 November 2019; Received in revised form 17 February 2020; Accepted 24 February 2020

Table 1
Related work summary.

Work	Structure	Decision fusion	UI management layer	Classification adopted
This proposal	Three layers	Yes	Yes	Dasarathy et al. (1997)
Wichit (2014)	Non-hierarchical	No	No	Not specified[a]
Bish et al. (9831)	Non-hierarchical	No	No	Not specified
Wang et al. (2015)	Two layers	No	No	Not specified
De Paola et al. (2016)	Three layers	No	No	Not specified[b]
André et al. (2017)	Three layers	No	Yes	Dasarathy et al. (1997)
Martins et al. (2018)	Four layers	Yes	No	Luo et al. (2002)

[a] The authors mention the "data level, feature level, and decision level" classification, but they do not specify the authorship of such classification (probably (Dasarathy et al., 1997)).

[b] Although not specified, the authors seem to follow the 'measurement, feature, and decision' data abstraction classification by Dasarathy et al. (1997).

data fusion and information fusion are accepted and used interchangeably (Nakamura et al., 2007). The concept of data fusion, although widely used, may vary depending on the context. According to Nakamura et al. (2007), data fusion can be defined as the combination of multiple sources to obtain improved information (optimized data quality). The word "quality" is a loose term intentionally adopted to denote that the fused data is somehow more appropriate to the application than the original data (Nakamura et al., 2007). From another perspective, the definition proposed by Boström et al. (2007) states that "*data fusion is the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human or automated decision making.*" In this work, we adopted the definition of data fusion, aiming at improving both data quality and decision-making.

In this paper, we propose Hydra, a multilevel data fusion architecture that aims at maximizing sensor accuracy, identifying application target events, and making decisions. Hydra is composed of three layers: (i) low-level, focused in sensor data fusion, including outlier identification and removal; (ii) medium-level, dealing with events and decision making based on sensed data and user-defined rules; and (iii) high-level, responsible for handle decision fusion based on multiple applications.

We instantiated the Hydra fusion architecture for the smart agriculture domain. In partnership with Embrapa, we developed two applications to monitor experimental cultures of precocious-dwarf cashew and coconut trees, aiming at smart water management. The first application intends to determine the irrigation water need based on soil moisture. The second application aims to estimate the crop evapotranspiration (rate of water evaporation by the soil and transpiration by plants Dantas Caminha et al., 2017) to determine an adequate irrigation time for those cultures. We performed a set of experiments to evaluate the methods of Hydra fusion data at each level. Furthermore, we used machine learning to create a new evapotranspiration model that resulted in values close to the Penman-Monteith evapotranspiration reference model (widely used in the agriculture domain).

The key contributions of this paper are: (i) a multilevel fusion data framework able to identify and remove outliers, filter signal and make decisions based on multiple applications, thus improving the quality of data and the decisions; and (ii) offer an accurate and lower cost solution to estimate the reference evapotranspiration for agriculture, because few variables will need to be monitored, thus improving the sensor network resource consumption.

This article is organized as follows: Section 2 discusses related works. Section 3 presents background information about data fusion classification, outlier detection methods, and filtering methods used. Section 4 explores Hydra and its components. Section 5 shows the instantiation of Hydra in the agriculture domain. Section 6 presents the material and methods. Section 7 presents the experiments and results obtained. Moreover, Section 8 draws the research conclusion and future works.

2. Related work

We carried out a systematic literature review to find the most relevant works in the data fusion field, in which they proposed either a multisensory or multilevel fusion model or architecture. A brief comparison between our approach and the related work is presented in Table 1 based on the following aspects:

- Structure: if the proposals follow a structured hierarchy or not. In general, a hierarchy structure allows a better modularity degree than a non-hierarchy structure. Besides, it is a noteworthy trend noticed in the most recent related papers.
- Decision fusion: if the paper proposes or presents any high-level fusion of decisions.
- User interface (UI) management layer: if the paper proposes any management interface for the final user aiming to provide a user-friendly experience.
- Classification: what type of classification scheme was followed (if any). Adhering to a classification makes it easier to understand the paradigms adopted in the paper.

The work in Wichit (2014) presents a multisensory data fusion architecture focused on the recognition of human behavior using a fuzzy logic-based fusion algorithm to improve accuracy and robustness. The authors do not specify a layer structure for the architecture, only presenting a workflow for human activity recognition and a fuzzy logic inference application.

The authors describe in Bish et al. (9831) a fusion engine architecture responsible for combining multimodal and multi-sensor fusion within an Open Standard for Unattended Sensors (OSUS) framework. The architecture goal is to be a modular plug and play system and to allow new fusion methods to be easily integrated. This paper focuses on how the structure proposed by the authors works in the OSUS framework, in which multisensory fusion is one of the model's elements. However, it does not go int detail about how it operates.

In Wang et al. (2015), the authors propose a two-level data fusion structure based on state estimation, which focuses on improving data accuracy. At the first level, sensors with the same data structure are grouped, and their data is merged. The second level receives the merged data from the first level groups and performs its merging using a covariance intersection algorithm. The authors focus on increasing data accuracy without addressing data interpretation, rulemaking, and decision-making.

In De Paola et al. (2016), the authors propose a context-aware, self-optimizing, adaptive system for sensor data fusion based on a three-tier architecture. The lower layer is responsible for sensing and generating raw data. The middle layer performs data fusion, in which it attempts to integrate contextual information. Also, the upper layer seeks to achieve a balance between system performance and execution costs (such as energy consumption). However, they did not specify the rulemaking and management interface, nor if any of the layers merged decisions. As

proof of concept, they used an activity detection scenario in an intelligent environment.

In André et al. (2017), the authors propose a three-layer architecture for low-reliability sensors focused on detecting and treating outliers. The first layer (local fusion layer) handles data collection and apply some local data fusion, such as arithmetic average. The second layer (low-level data fusion layer) is their main contribution and it focuses on calibrating, timestamping, detecting gross and systematic errors, and executing data fusion either online or offline (in which the cluster heads forward the data received). Moreover, the third layer (management layer and the user interface) is responsible for managing the data acquired. The authors focus on accuracy and do not address data interpretation, rulemaking, or decision-making.

Finally, in Martins et al. (2018), the authors present a knowledge fusion algorithm called Athena, which appears to follow the four levels of abstraction described by Luo et al. (2002) (signal, pixel, feature, and symbol/decision), though it is not clearly stated. The proposed algorithm focuses on enhancing accuracy on a multi-application wireless sensor and actuator network (WSAN) using multisensory data fusion supporting knowledge extraction. The algorithm achieved increased precision when compared to a moving average filter (MAF) while consuming less energy. Unfortunately, the authors do not address how it could handle outliers or any management interface for setting application rules.

Compared to the work cited above, the Hydra architecture differentiates by being multilevel, based on Dasarathy et al. (1997) proposal of the organization of data input-output abstraction. Moreover, the data fusion is performed not only to save energy on transmission, but also to identify events and for decision-making, and opening the possibility of decision fusion on multiple applications running on the same platform.

3. Background

This section provides a brief background about the data fusion classification scheme adopted, the outlier identification methods, and the filtering methods tested during the experiments presented in Section 7.

3.1. Data fusion classification

Data fusion can be classified based on several aspects, such as relationships among sources, levels of abstraction, data input-output abstraction levels, among others (Nakamura et al., 2007). On data fusion based on the relationship among sources, there is no direct sensor dependency. The relationship can be divided into complementary (sensors provide different information about the same scenario), redundant (sensors provide the same information about the same scenario), and cooperative (information from different sensors can be merged to generate new information).

Data fusion classification based on levels of abstraction, proposed by Iyengar et al. (2001) and complemented by Nakamura et al. (2007), is divided into four levels: low-level, medium-level, high-level, and multilevel. Low-level (also called signal or measurement level) deals with raw data, which can be combined to generate more accurate data. Medium-level (or feature/attribute level) represents entity features, which can be combined to obtain new features. A high-level (or decision level) represents the union of decisions to attain new decisions or a more precise one. Moreover, multilevel fusion represents the possibility of fusing data from different abstraction levels.

Data fusion classification based on data input-output abstraction can be considered a more granular version of classification based on levels of abstraction. The work of (Dasarathy et al., 1997) expands the concept of data, feature, and decision into five categories:

- *Data In–Data Out* (DAI-DAO): fusion deals with raw data as input and output, resulting in possibly more accurate or reliable data.
- *Data In–Feature Out* (DAI-FEO): fusion uses raw data to extract features or attributes that describe an entity (object, situation, or world abstraction).
- *Feature In–Feature Out* (FEI-FEO): fusion works on a set of features to refine a feature or extract new ones.
- *Feature In–Decision Out* (FEI-DEO): fusion takes a set of features of an entity and generates a decision.
- *Decision In–Decision Out* (DEI-DEO): decisions can be fused to obtain new decisions or enhance an existing one.

3.2. Outlier Detection Methods

3.2.1. Z-Score

Also known as 'standard score' or z-value (NIST/SEMATECH, 2013), it is the number of standard deviations a data point is above or below the mean of the dataset being observed. A common criterion is considering observations with z-score higher than 3 (either positive or negative) as outliers.

3.2.2. Modified z-score

The z-score as one issue in that the mean can be affected by outliers; hence the modified z-score method replaces it with the median and the standard deviation with the median of absolute deviations (MAD) (NIST/SEMATECH, 2013). Observations by Iglewicz and Hoaglin (1993) suggest using a modified z-score value of 3.5 to identify potential outliers.

3.2.3. Adjusted boxplot

The boxplot method represents data using a central line (usually the median), a lower quantile (25th percentile), and upper quantile (75th percentile). Above and below the quantile are located the fences, and data beyond these fences are considered potential outliers. This method has the limitation of becoming skewed with the presence of outliers, and to counter that (Hubert et al., 2008) proposed the adjusted boxplot method, that uses medcouple to counter skewness.

3.2.4. Generalized ESD

The Generalized Extreme Studentized Deviate (ESD) (NIST/SEMATECH, 2013) method checks the dataset for one or more outliers. It is similar to the Grubbs test, but the number of outliers expected does not have to be specified correctly, only an upper boundary.

3.2.5. Chauvenet's criterion

Chauvenet's criterion (Taylor, 1997) works by creating a probability band based on the data samples and considering any data point outside that band as an outlier (over two standard deviations). This method tends to not work very well on small datasets, in which a normal distribution cannot be assumed.

3.2.6. Peirce's criterion

Although being created before Chauvenet's criterion, Peirce's criterion (Ross, 2003) provides a more robust method to identify outliers. This method takes the mean and standard deviation of the dataset, calculates the maximum allowed deviation, and checks if the actual deviation of the potential outlier is greater than the maximum or not. If it is, the outlier gets removed, and it continues to iterate for all possible outliers.

3.3. Filtering methods

3.3.1. Kalman filter

The Kalman filter is a two-step algorithm: first, it predicts (estimates) a state with a degree of uncertainty. Afterward, based on the next observed measurement, it updates the estimates using a weighted average and giving more weight if the estimate was close to the observed measurement. Kalman filter is a recursive algorithm that can run

in real-time and only requires the previous measurement and the uncertainty matrix (that can be an estimate on the first iteration).

3.3.2. Weighted Outlier-Robust Kalman Filter (WRKF)

The Kalman Filter is not robust to outliers, and the Weighted Outlier-Robust Kalman Filter (WRKF) (Ting et al., 2007) proposes the use of weights for each data sample and having a data point with a small weight to have less influence on the estimation step.

3.3.3. Robust locally weighted scatterplot smoothing (rloess and rlowess)

Both methods use weighted linear regression to smooth the dataset (Cleveland et al., 1979), in which the neighboring data points are used to determine the smoothed value, and a robust weight function makes them resistant to outliers. Rloess uses a quadratic polynomial regression model, and rlowess uses a linear polynomial model.

3.3.4. Savitzky-Golay

Savitzky-Golay filtering (Savitzky and Golay, 1964) can be considered as a generalized moving average, and it removes high-frequency noise while preserving the original shape and features of the data. It works by fitting each data point within a high order polynomial using the linear least square method.

3.3.5. Scale-space

Scale-space filtering is a method that uses convolution to expand the data signal with gaussian masks, and then collapse it into a tree providing a concise but still complete description covering all scales of observation (Witkin, 1984).

4. Hydra: a multilevel data fusion architecture

Hydra is a multilevel data fusion architecture based on data classifications defined in Dasarathy et al. (1997) and Iyengar et al. (2001). Fig. 1 shows an overview of the Hydra architecture structure, which is composed of three fusion layers (low-level, medium-level, high-level) and one management layer. The low-level layer focuses on sensor data fusion, including outlier identification, removal, and signal filtering. The medium-level is responsible for dealing with events and decision making based on sensed data and user-defined rules. The high-level is responsible for handling decision fusion based on multiple applications. Finally, the management level is responsible for the user interface, in which the users define the rules of the medium-level and high-level, rule priority, and application execution time.

Regarding the network infrastructure of our proposal, wireless sensor nodes gather data from sensor units (e.g., soil moisture) connected to them and forward it to a sink node. Afterward, the sink node relay that data to the internet.

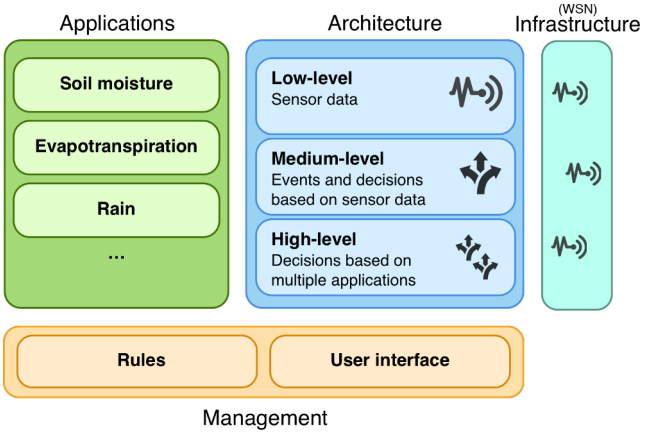

Fig. 1. Hydra architecture structure overview.

Hydra Architecture is independent in terms of domain areas and infrastructure. Hydra architecture levels are applied in a distributed way, and those levels can be applied in-network (to the sensor nodes of the network) or external of the sensor network (a sink node or a cloud service). For example, the sensors nodes can process the low-level and medium-level layers. Otherwise, the sensors can only relay raw data to be processed by a sink node or a cloud service. Finally, high-level processing can be performed by nodes able to fuse decisions of several applications such as a sink node, an edge node, or a cloud service. More details about the fusion layers of the Hydra architecture are shown in Fig. 2.

Specifically, the low-level layer is responsible for receiving raw sensor data, performing preliminary filtering (according to the nominal working scope of each sensor), and generating filtered data. In the low level, performing raw data fusion to obtain a feature (interpreted data) is also possible. Below some examples illustrate the low-level layer in the field of smart agriculture:

- DAI-DAO fusion: raw sensor data results in a new raw data. For example, multiple temperature sensors generate 12-bit integers, which are fused, and the final value can be converted to Celsius or Fahrenheit.
- DAI-FEO fusion: Interpreted data is obtained from raw sensor data. It can be similar data (redundant fusion) or different data (cooperative fusion). For example, a soil water voltage sensor generates a 10-bit integer representing the soil resistance to an electric current (Ohms), and, in conjunction with data from the soil temperature sensor (Celsius degree), an interpreted value is generated (kPa).

In the low-level layer, we can use extreme value analysis, proximity-based models, or statistical modeling for outliers detection such as Chauvenet's Method, Peirce's Criterion, Z-Score algorithms, Generalized Extreme Studentized Deviate, Adjusted boxplot, among others. Besides, we can also use a signal smoothing technique to reduce signal noise such as Kalman filter, Weighted outlier robust Kalman (WRKF), Savitzky-Golay, Robust locally weighted scatterplot smoothing (RLOESS and RLOWESS), Scale-space, among others.

Data generated by the low-level layer are sent to the medium-level layer, responsible for performing complementary sensor fusion and executing decisions based on user-defined rules. In such a level, this type of fusion technique is application-specific, i.e., features representing aspects of the environment can be extracted and used by the application. Below are examples to illustrate the medium-level layer in the field of smart agriculture:

- DAI-FEO fusion: interpreted data obtained from raw filtered sensor data (e.g., multiple sensors data from a meteorological station to obtain new information, such as evapotranspiration).
- FEI-FEO fusion: interpreted data combined to obtain new data (e.g., multiple sensors data fused to obtain more accurate interpreted data).
- FEO-DEO: interpreted data use to achieve a decision (e.g., deciding if the soil needs irrigation based on fused soil moisture data and predefined thresholds).

Data and decisions from the medium-level are sent to the management layer and the high-level layer, when necessary. High-level layer deals with decision fusion (DEI-DEO), performed by multiple applications, to execute a more complex composed decision. This composed decision is taken according to the rules defined in the management level. For example, the high-level decides which sprinklers to activate and for how long based on crop irrigation needs, generated by a soil moisture application, and adequate irrigation time, generated by an evapotranspiration application (Fig. 3). A rule engine is typically employed for composing decisions at medium and high levels. Still, inference methods such as Bayesian theory, Dempster-Shafer reasoning,

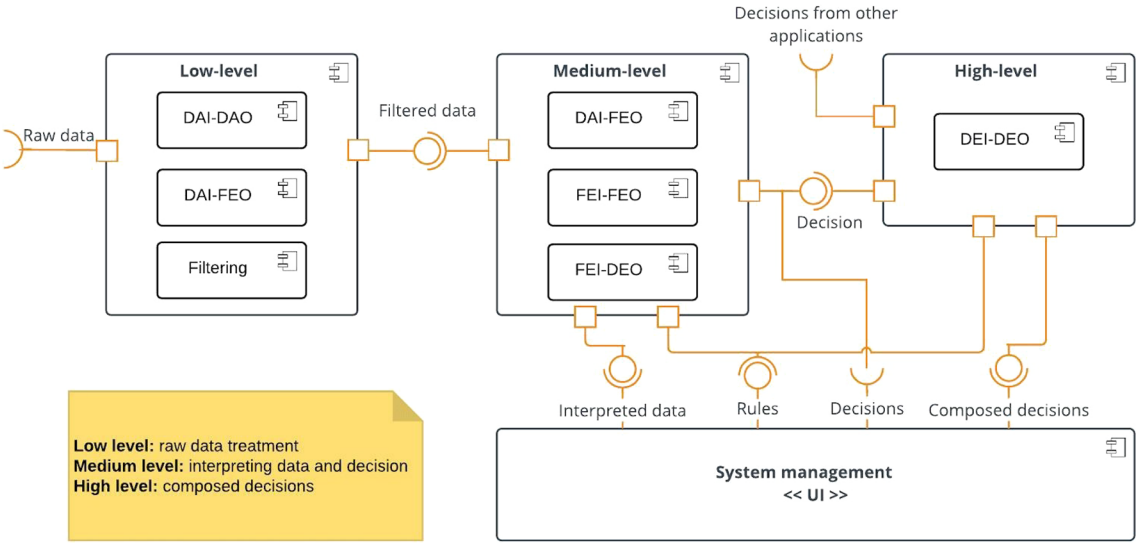

Fig. 2. Hydra architecture layers detail.

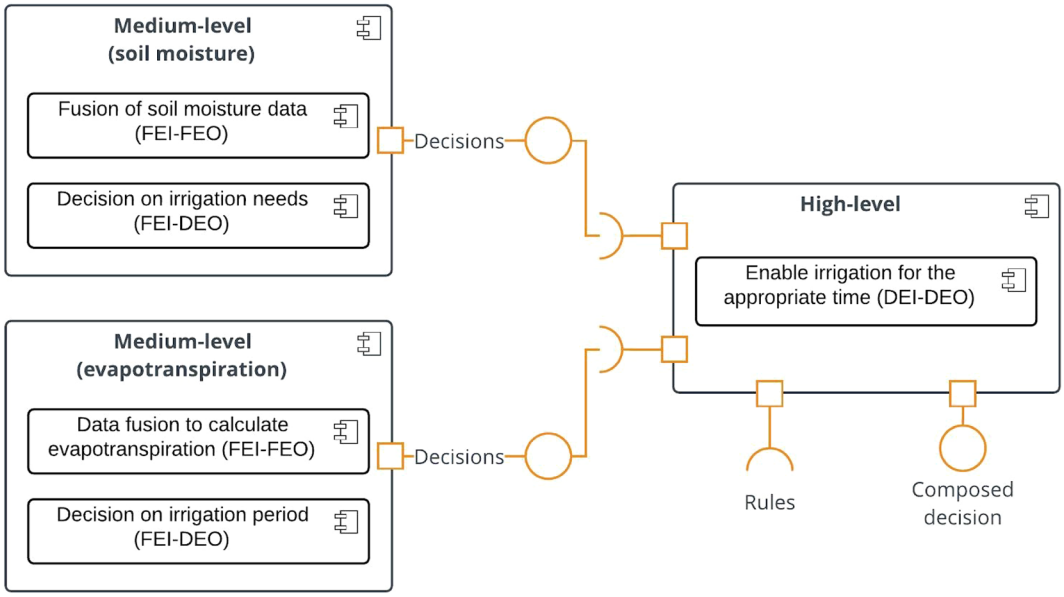

Fig. 3. The high-level layer composed decision example.

and logic fuzzy could also be employed at these levels. In this case, the knowledge base is built based on the rules defined by users.

Moreover, the management layer sits above the fusion layers, receiving data from medium and high-level layers. The management layer is responsible for the user interface, and it defines the rules (knowledge base) that govern the decisions of the fusion layers, the priorities of the rules, and the execution time of the applications. On it, the user can track generated data and decision results.

5. Hydra instantiation in smart agriculture domain

We instantiated Hydra architecture for the smart agriculture domain to demonstrate its operation. For this purpose, we developed two applications focused on smart water management: monitoring soil moisture and estimating evapotranspiration. The first application aims to determine if a crop requires irrigation. The second one aims to help farmers efficiently manage the amount of water needed to irrigate adequately. Developed applications are explained in the following subsections.

5.1. Soil moisture application

This application aims to determine if a crop requires irrigation based on soil moisture measures. For this purpose, it requires soil water tension data from various depths to monitor soil water absorption and determine its humidity. Another critical factor is the soil temperature that also requires monitoring. Fig. 4 shows the detailed application structure according to Hydra architecture. Raw data is treated at the low-level layer, data interpretation and decisions are performed at medium-level, and composed decisions are performed at the high-level layer. The following subsections provide more details about each layer.

5.1.1. Low-level: DAI-FEO fusion

Soil water tension sensors generate raw data in *Ohms*, which must be interpreted into a unit that is useful for agricultural calculations, Kilopascal (kPa). The recommended formula (Eq. 1) (Shock et al., 1998) also uses the soil temperature data to obtain a more accurate conversion. Thus, we have a DAI-FEO fusion, which uses raw data from two types of sensors to generate feature data that is easier to interpret and

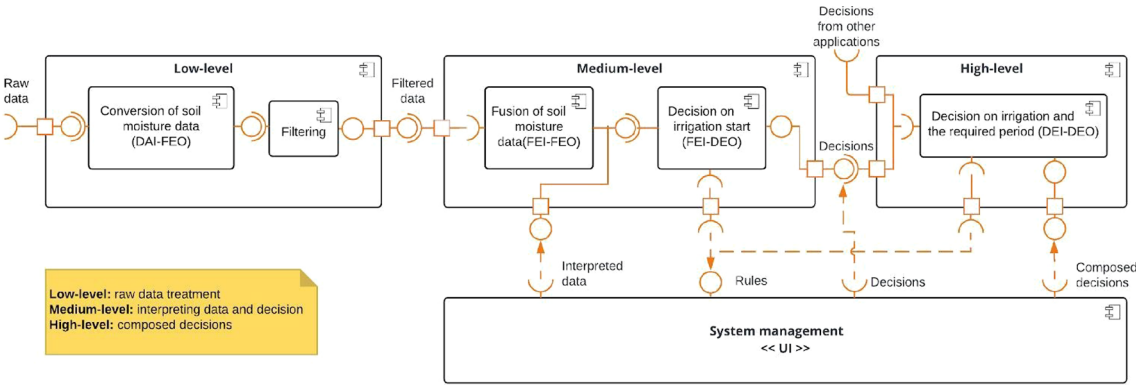

Fig. 4. Soil moisture application levels.

analyze.

$$soilMoisture = \frac{3.213 * \frac{rawData}{1000} + 4.093}{1 - 0.009733 * \frac{rawData}{1000} - 0.01205 * soilTemperature} \quad (1)$$

soilMoisture: equation result that represents soil moisture (kPa);
rawData: raw data provided by the soil water tension sensor (Ohm);
soilTemperature: soil temperature (°C).

5.1.2. Low-level: Filtering

Ensuring the generated data within standard operating parameters is crucial. A simple filtering process using the minimum and maximum values can avoid atypical data (outliers) to cause a wrong decision. Outliers are values that deviate from other readings in a sample, and the readings variability can be one of the causes (Torres et al., 2017). This filtering step can also identify problematic sensors that might require repairs or replacement.

5.1.3. Medium-level: FEI-FEO Fusion

Although it is possible to perform data fusion from various sensors to get more accurate data, a sensor can generate outliers within its standard operating parameters, caused either by hardware failure or communication issues, affecting the fused output. To solve such an issue, it requires to adopt outlier detection (e.g., extreme value analysis, statistical modeling, proximity-based models) and outlier treatment (e.g., trimming, imputation, discretization) methods. Finally, after dealing with the outliers, data can be fused using techniques such as a weighted average, or a more robust one, such as a Kalman filter.

This implementation of FEI-FEO fusion takes data from multiple moisture sensors, filters possible outliers, and combines them into more accurate data (more details in Section 7.2). This information is used by the decision step and by the management layer, where the user can monitor the crop and sensor status.

5.1.4. Medium-level: irrigation decision

Irrigation decision is based on rules defined by the user on the management level and, based on them, data about soil moisture and rain status are assessed to decide whether to irrigate the crop or not. Thus, we have a FEI-DEO fusion.

5.2. Evapotranspiration application

Evapotranspiration is the combination of two processes, water evaporation from the soil surface and transpiration from the crop (Allen et al., 2006), and it is a significant factor for agricultural water management (Anderson and French, 2019). The Reference Evapotranspiration (ETo) is an estimate calculated using weather data (formula described in Section 5.2.2), and it is used by this application to estimate the amount of water needed to irrigate a crop adequately. The application obtained the weather data from sensors installed at the farm and from a local weather station.

Fig. 5 shows the application levels using Hydra architecture. The low-level layer is responsible for data filtering, the medium-level is responsible for calculating evapotranspiration and decision-making the irrigation period, and the high-level is responsible for composed decisions. More details are given in the following subsections.

5.2.1. Low-level: filtering

The weather station receives raw data from many sensors and converts them into physical units (e.g., solar radiation balance into MJ m^{-2} d^{-1}). This data goes through the same basic filtering scheme as the soil moisture application before the data is relayed to the medium-level layer.

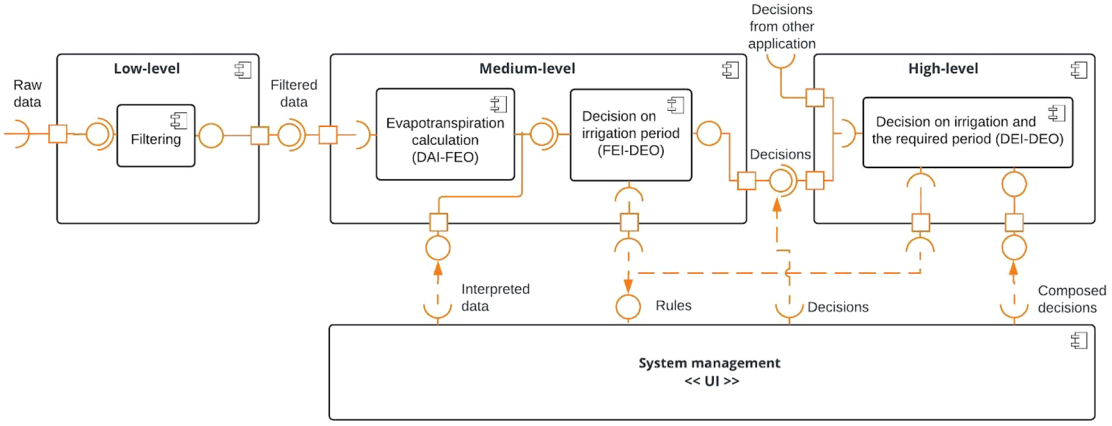

Fig. 5. Evapotranspiration application levels.

Table 2
Experiments goals.

Experiment	Objective	Metric
I	Assess the prototype sensor nodes	Power consumption
II	Assess the outlier identification algorithms and signal smoothing filters using data gathered by the medium-level layer	Original and filtered data, total outliers detected
III	Evaluate the data generated by the Evapotranspiration and soil moisture applications	Data correlation
IV	Evaluate new evapotranspiration model using gathered sensor data	Determination, mean square error, mean absolute error, prediction speed, training time, data correlation

5.2.2. Medium-level: evapotranspiration calculation

Currently, the most precise ETo calculation is based on the Penman-Monteith method (Allen et al., 2006), which uses many meteorological and crop variables (Eq. 2). However, these variables depend on having a weather station with many sensors, hence using them in all scenarios is not viable.

$$ETo PM = \frac{0,408 * \Delta * (Rn - G) + \frac{\gamma * 900 * U_2 * (e_s - e_a)}{T + 273}}{\Delta + \gamma * (1 + 0,34 * U_2)}, \quad (2)$$

ETo PM: reference evapotranspiration by the Penman-Monteith method;
Δ: slope of saturation vapor pressure curve (kPa $^{\circ}C^{-1}$);
Rn: net radiation (MJ m^{-2} d^{-1});
G: soil heat flux (MJ m^{-2} d^{-1});
γ: psychrometric constant (kPa °C);
U_2: wind speed at 2 m above the ground surface (m s^{-1});
es: saturation vapor pressure for a given time period (kPa);
ea: actual vapor pressure (kPa);
T: air temperature (°C).

This step performs a FEI-FEO fusion in which interpreted data from the weather station is fused to generate new data, ETo, which is used by the decision step.

5.2.3. Medium-level: irrigation period decision

Based on Allen et al. (2006), the decision about how much water should be used for irrigation is determined by the following factors:

- ETo (calculated by the application on the previous step);
- Kc (static coefficient, according to the crop kind);
- Irrigation system efficiency (static value provided by the user, in this case, an Embrapa researcher).

5.3. High-level decision

The high-level layer receives the decision taken by the medium-level of both applications and performs a new decision, according to the rules defined in the management level.

5.4. Management and user interface level

The management level is responsible for the user interface, in which medium and high-level rules, rule priority, and application execution time are defined.

6. Experimental design

This section details the experimental design, presenting the goals of the experiments, material (environment, hardware, communication, and software), execution methods, the database gathering process, and the parameters and algorithms used to perform both low-level and medium-level layers data filtering.

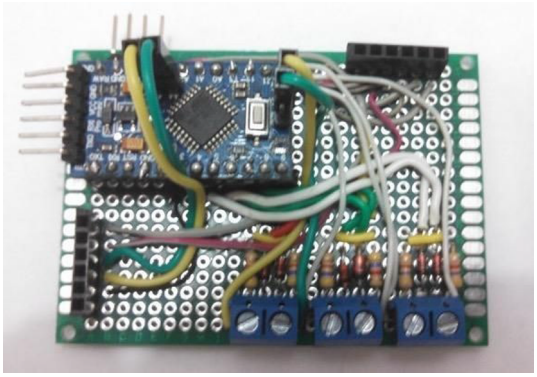

Fig. 6. Type A sensor node.

6.1. Experiments goals

We performed four experiments (Table 2). The first experiment aims to assess the energy consumption of our modified sensor nodes. The goal of the second experiment is to assess various outlier identification algorithms and signal smoothing filters (low-level). The goal of experiment three is to evaluate the data generated by the medium-level of the evapotranspiration and soil moisture applications. Moreover, experiment four evaluates the feasibility of creating a new evapotranspiration model using the gathered sensor data.

6.2. Hardware (Prototype)

The hardware prototypes developed were comprised of two types of sensor nodes, which are responsible for gathering data from sensors connected to them. Additionally, we designed a sink node responsible for receiving data from the sensor nodes and relaying them to the internet.

We designed the first kind of the sensor node (Type A - Fig. 6)) to assess the usage of the Irrometer Watermark 200SS[1] sensor (soil moisture sensor) with an Arduino board. The second one (Type B - Fig. 7) was developed to collect weather information (soil temperature and rain status), also using the Arduino board. As an energy source, AA 1.5v batteries powered both nodes. We optimized the Arduino board by removing the tension controller and LED status to achieve a higher battery lifespan (see Experiment I). The Arduino platform is a low-cost and easily attainable open-source hardware (Creative Commons Attribution Share-Alike License) that allows quick prototyping. We developed the sink node based on a Raspberry Pi Zero, and it was the only equipment connected directly to the power grid.

The complete hardware specifications are described as follows:

- Type A sensor node:
  - Arduino Pro Mini;
  - 3× Soil Moisture Sensor (Irrometer Watermark 200SS);

[1] Irrometer Watermark 200SS: http://www.irrometer.com/sensors.html#wm.

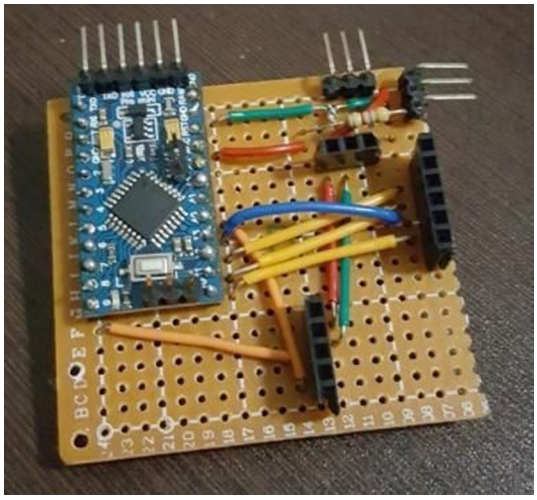

Fig. 7. Type B sensor node.

- 2.4GHz Radio (nRF24l01+ with an external antenna and power amplifier).
- Type B sensor node:
  - Arduino Pro Mini;
  - Wetness sensor;
  - Waterproof temperature sensor;
  - 2.4GHz Radio (nRF24l01+ with an external antenna and power amplifier).
- Sink node:
  - Raspberry Pi Zero;
  - 2.4GHz Radio (nRF24l01+ with an external antenna and power amplifier);
  - Wi-fi adapter.

6.3. Environment and deployment

We deployed our prototypes at an experimental farm maintained by Embrapa at Paraipaba (Ceará, Brazil). We monitored two different crops (precocious dwarf-cashew and coconut) by using the prototypes. The weather station used in the experiments is located at Itapipoca and maintained by the National Institute of Meteorology (INMET). We chose this station due to its proximity to the experimental farm used in the experiments (approximately 46 kilometers). We deployed four sensor nodes of type A on the precocious dwarf-cashew crop, installed according to Fig. 8. The sensor nodes were installed on an open field, 11–22 m away from the sink node, with natural obstacles between them (e.g., leaves, branches, tree trunks).

On the coconut crop, we deployed both type A (4 units) and type B (1 unit) sensor nodes, installed according to Fig. 9. Sensor nodes were also installed on an open field, 48 to 98 meters away from the sink node. The sink node was installed on a shed (support building) at the farm.

6.4. Software

We developed the sensor nodes firmware by using C++ and libraries developed for the Arduino platform. Besides, we developed the sink node software by using Python language, and that software is responsible for receiving, storing, and uploading sensor data to the Internet. We used the Thingspeak[2] platform for implementing the management level and the user interface. Thus, the sink node can connect to the Internet and publish the gathered data to Thingspeak platform using the MQTT protocol, allowing real-time data visualization.

[2] Thingspeak: http://www.thingspeak.com/.

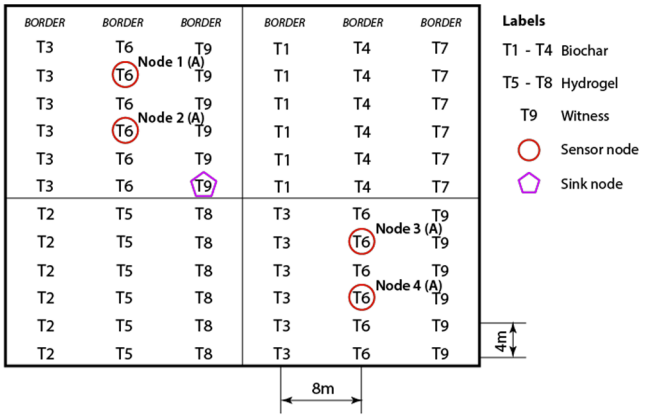

Fig. 8. Precocious dwarf-cashew deployment sketch.

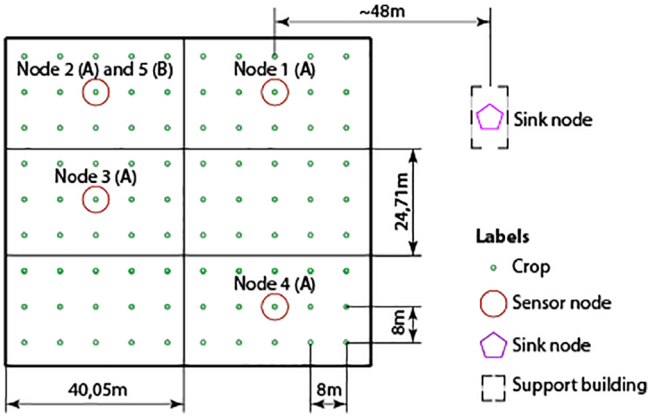

Fig. 9. Coconut crop deployment sketch.

6.5. Communication

Communication among nodes was performed using a non-IP network using 2.4 GHz radio modules (nRF24L01+) and the nRF24[3] and nRF24Network[4] libraries. Fig. 10 shows an example of a message structure transmitted (in this case, sent from a type B sensor node) in the network. After each transmission, the node sleeps for one hour to preserve power and, after waking up, it restarts the transmission cycle.

Moreover, the sink node communication with the Internet occurred by using the Wi-Fi protocol and the MQTT message protocol. We chose the MQTT protocol because it adopts the Publisher/Subscriber pattern, where data is centralized on a gateway called the broker, which manages the data and client requisitions. Thus, the sink node can send the data generated by the WSN (wireless sensor network) to a broker on the cloud, making the data accessible to anyone or any service on the Internet.

6.6. Data gathering

We performed two distinct data-gathering sessions: (i) on the precocious dwarf-cashew crop data collection occurred for seven days (from 31 January 2017 to 07 February 2017); and (ii) on the coconut crop data collection occurred for 35 days (from 25 April 2017 to 29 May 2017).

In the first data-gathering session, the sink node received hourly data from sensor nodes (Table 3 lists the collected attributes). Each node is connected to a moisture sensor, which requires soil water

[3] nRF24: https://github.com/nRF24/RF24.
[4] nRF24Network: https://github.com/nRF24/RF24Network.

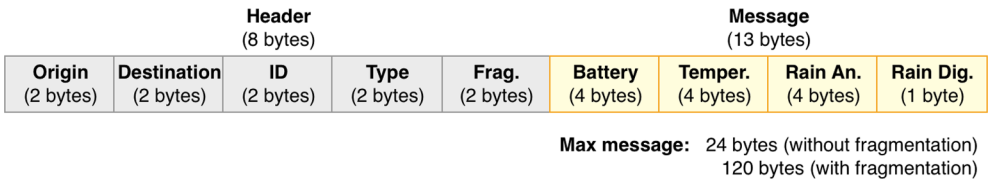

Fig. 10. Message structure example.

Table 3
Data gathering 01 – complete list of attributes.

Node	Attribute	Unit
Nodes 1 to 4 (type A)	Soil moisture at 15cm, 45cm and 75cm depth	Ohm

tension data from three depths (15 cm, 45 cm, and 75 cm) to monitor soil water absorption and determine the humidity of the soil. Raw data had some gap interval due to hardware and communication issues. After preprocessing the data, the number of tuples dropped from 168 (original) to 151 (current).

On the second data-gathering session, the sensor nodes collected data every 30 minutes, and the sink node published them to an online service (Thingspeak) hourly via Wi-Fi. Meteorological data were also collected hourly from the automatic weather station, located at Itapipoca, Ceará. Raw data have some gaps due to communication or hardware issues. We normalized the data set to 60 min intervals aiming to facilitate comparison and analysis, resulting in 667 tuples. After preprocessing, filtering, and outliers removal, the final dataset presents around 340 tuples. Table 4 lists the collected attributes.

7. Experiments and results analysis

This section provides details about the experimentation performed with Hydra.

7.1. Experiment I: Energy consumption

Before deployment, we performed a power consumption test using a USB voltage and current meter to assess the energy consumption of our modified Arduino nodes and modules. We compared the equipment in the following scenarios: idle without radio module, idle with radio module, and transmitting data. The reduction in energy consumption on idle was up to 99,67%, and during transmission, it was up to 32,69% (Table 5). Since the sensor nodes will stay in hibernation most of the time, such a reduction in energy consumption can increase their lifespan considerably.

Table 4
Data gathering 02 – complete list of attributes.

Node	Attribute	Unit
Nodes 1 to 4 (type A)	Soil moisture at 15cm, 45cm and 75cm depth	Ohm
Node 5 (type B)	Rain (wetness)	–
	Soil temperature at 45cm depth	Celsius
Weather Stations	Temperature – instant, max, and min	Celsius
	Relative air moisture – instant, max, and min	%
	Dew point-instant, max, and min	Celsius
	Atmospheric pressure – instant, max, and min	kPa
	Weather Station Wind speed	m/s
	Wind direction	Degree
	Wind gust	m/s
	Solar radiation	kJ/m^2
	Precipitation	mm

Table 5
Nodes power consumption.

Activity	Arduino Pro Mini	Arduino Pro Mini (modified)	Energy Consumption Reduction
Idle without radio	3,05 mA	> 0,01 mA	99,67%
Idle with radio	11,50 mA	2,20 mA	80,87%
Transmitting data	26,00 mA	17,50 mA	32,69%

7.2. Experiment II (Part I): Outlier detection and removal

Before sending the data to the medium-level layer, identifying and removing outliers that may affect data accuracy is necessary. Thus, we performed a filtering process at the Hydra Low-Level. Therefore, we determined some of the removal criteria (Table 6) based on each sensor's nominal working parameters and the domain expert experience.

Even after filtering data on the low-level, some outliers still can be within the sensor nominal working range, caused either by a malfunction or by signal noise during reading. The soil moisture application fuses data from multiple soil moisture sensors, making it possible to compare values among sensors using outlier-identifying methods. As mentioned in Section 3.2, we evaluated the following outlier-identifying methods: Chauvenet's Criterion (Taylor, 1997), Peirce's Criterion (Ross, 2003), Z-Score (NIST/SEMATECH, 2013), Modified Z-Score (NIST/SEMATECH, 2013), Adjusted Boxplot (Hubert et al., 2008) and Generalized Extreme Studentized Deviate (ESD) (NIST/SEMATECH, 2013).

We used the data from the first gathering session (on the precocious dwarf-cashew crop) to evaluate various outlier identification methods and signal filtering. Fig. 11 presents outliers detected by Peirce's Criterion and Modified Z-Score at 15cm depth. We do not present the result of the other methods since they did not detect outliers. Peirce's Criterion detected some peaks on node 2 but failed to detect higher values, most likely because node 3 stopped working, and it only had three values to compare. Modified Z-Score failed to detect the smaller initial peaks but managed to detect outliers after the death of node 3.

At 45cm depth, there are some definitive outliers on node 3 (Figs. 12 and 13). Due to the extreme values from node 3, the Y-axis in the figures is presented on a logarithmic scale to maintain legibility. Figs. 12 and 13 present the detected outliers, and again some methods failed to detect anything. For example, the Adjusted Boxplot method failed to detect some outliers (on January 31 and February 1), but in other periods, this method managed to detect node 4 outliers (February 5). The ESD method successfully detected the outliers on January 31 and February 1 and detected the same outliers as the Adjusted Boxplot. The Peirce and modified Z-score criteria were the most sensitive methods, detecting outliers in five of the seven monitored days, eventually removing valid data.

At 75cm depth, all the methods worked as expected and did not detect outliers since the gathered data did not show noticeable variation. Based on our results, we adopted the ESD method for being able to detect apparent

Table 6
Outliers removal criteria: hydra low-level layer.

Attribute	Criteria
All	Invalid values (e.g., −1)
Soil moisture	< 0 or > 200 kPa
Soil temperature	< 20 or > 32 °C

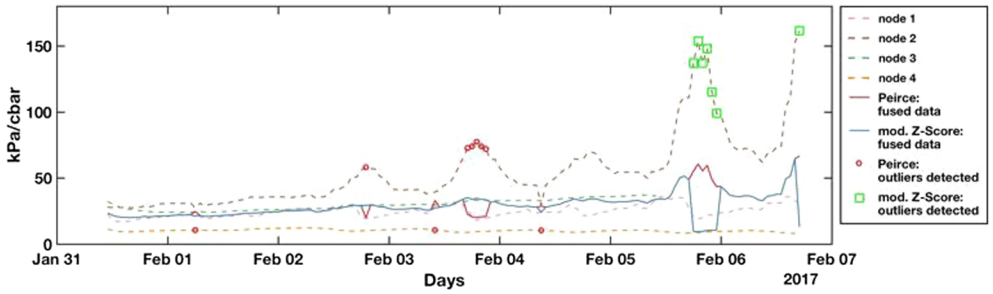

Fig. 11. Identified outliers and fused data (15 cm depth).

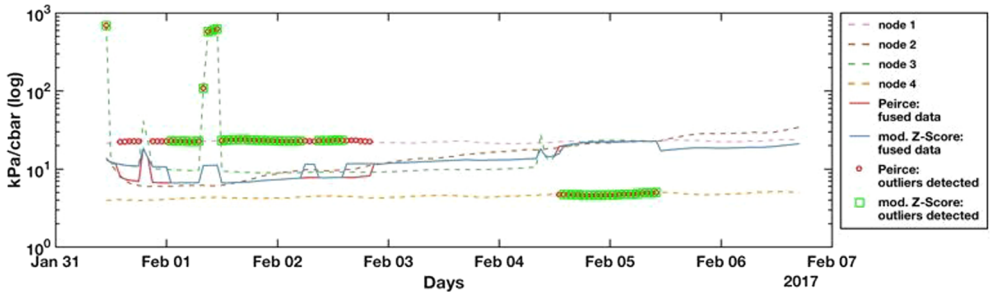

Fig. 12. Identified outliers and fused data - Mod. Z-Score and Peirce's Criterion (45 cm depth).

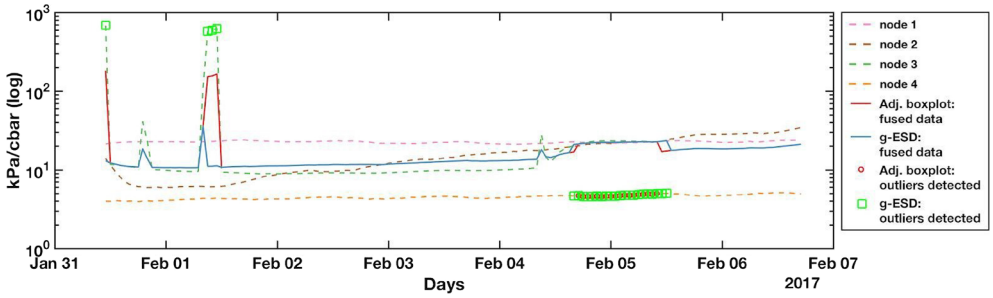

Fig. 13. Identified outliers and fused data - Adjusted Boxplot and ESD (45 cm depth).

Table 7
Outliers removal criteria: hydra low-level layer.

Method	15 cm	45 cm	75 cm	Comments
Adjusted Boxplot	0	18	0	Did not remove obvious outliers
Chauvenet's Criterion	0	0	0	Did not detect outliers
ESD	0	25	0	Chosen method
Peirce's Criterion	9	76	0	Too sensitive
Z-Score	0	0	0	Did not detect outliers
Modified Z-Score	7	63	0	Too sensitive

outliers and not being too sensitive and removing possible relevant data. Table 7 presents a summary of each method's performance.

7.3. Experiment II (Part II): Signal filtering

We also evaluated the filters used to smooth noise inherent in sensor readings. Thus, after using ESD to remove outliers, we filtered the data to remove inherent signal noise from the sensor signal reading and any residual outlier by using the following filters methods (previously explained on Section 3.3): Kalman filter, Weighted outlier-robust Kalman filter (WRKF), Savitzky-Golay, Robust locally weighted scatterplot smoothing (rloess and rlowess) and Scale-space. All of the tested filters

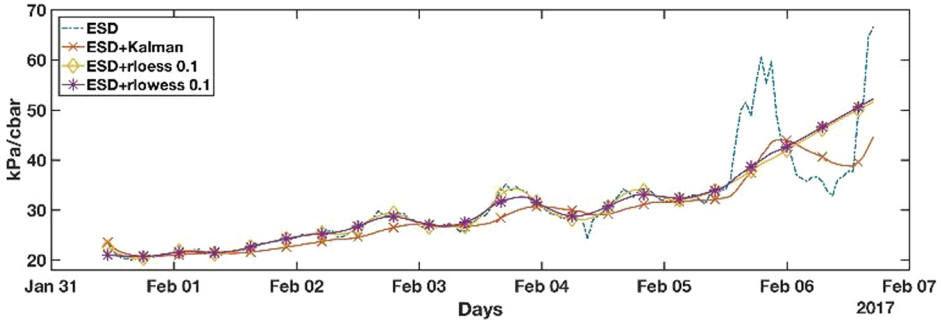

Fig. 14. Original and filtered data - Kalman, rloess, and rlowess (15 cm depth).

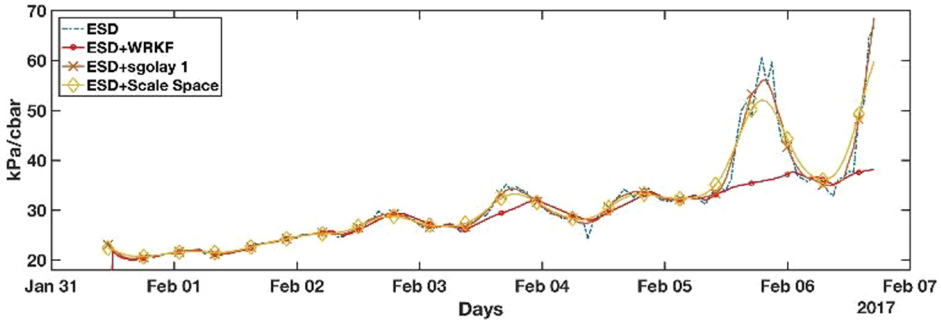

Fig. 15. Original and filtered data - WRKF, Savitzky-Golay, and Scale-space (15 cm depth).

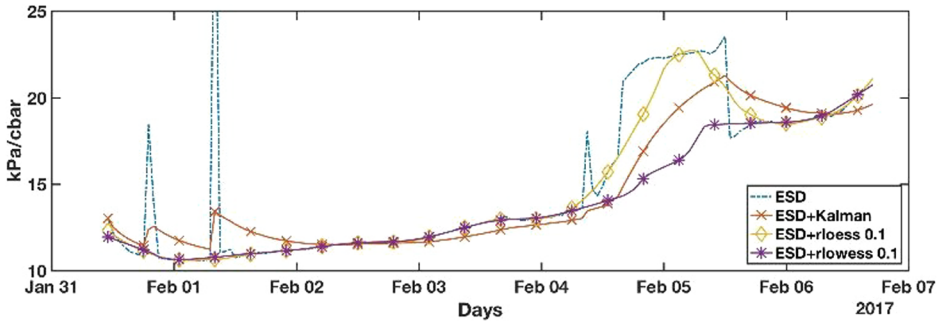

Fig. 16. Original and filtered data - Kalman, rloess, and rlowess (45 cm depth).

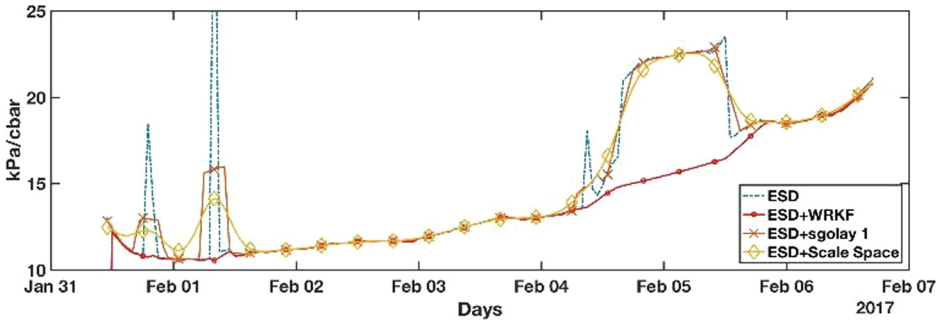

Fig. 17. Original and filtered data - WRKF, Savitzky-Golay, and Scale-space (45 cm depth).

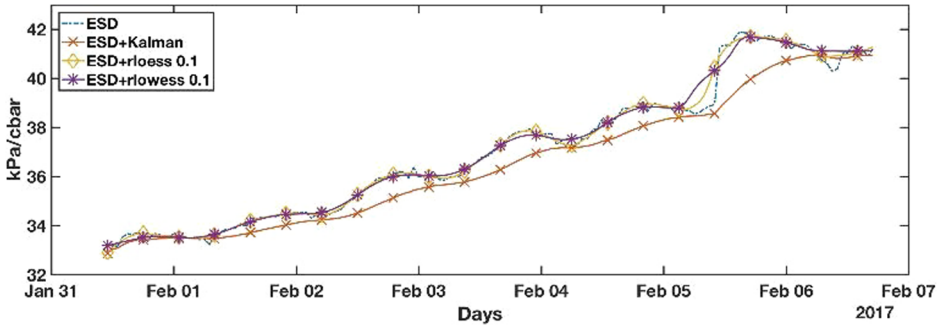

Fig. 18. Original and filtered data - Kalman, rloess, and rlowess (75 cm depth).

successfully smoothed data, but the results vary in regards to removing peaks in the data.

On data from 15cm depth sensors, rloess and rlowess smoothed too much of the data after February 5th (Fig. 14). The Kalman filter presented results closer to the source data, but the peak on February 5th also influenced it. Fig. 15 shows Savitzky-Golay and Scale-space methods smoothed the signal, but they failed to remove the February 6th peaks. Overall, the WRKF filter obtained the best result.

On data from 45cm depth sensors, although ESD managed to remove some apparent outliers, some anomalous data still lingered. Figs. 16 and 17 show how the peaks on January 31st and February 01st affected the Kalman, Savitzky-Golay, and Scale-space methods. Such peaks did not affect rloess and rlowess methods, but rlowess alone was resilient to the peaks on February 05th. Once again, the WRKF method obtained the best result.

Finally, on data from 75 cm depth sensors, almost all evaluated methods obtained similar results, with only a Kalman filter presenting a negative bias (Figs. 18 and 19). Therefore, based on our results, for this

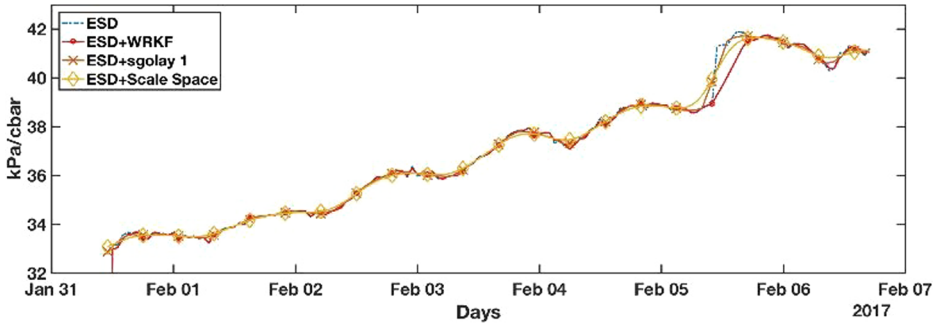

Fig. 19. Original and filtered data - WRKF, Savitzky-Golay, and Scale-space (75 cm depth).

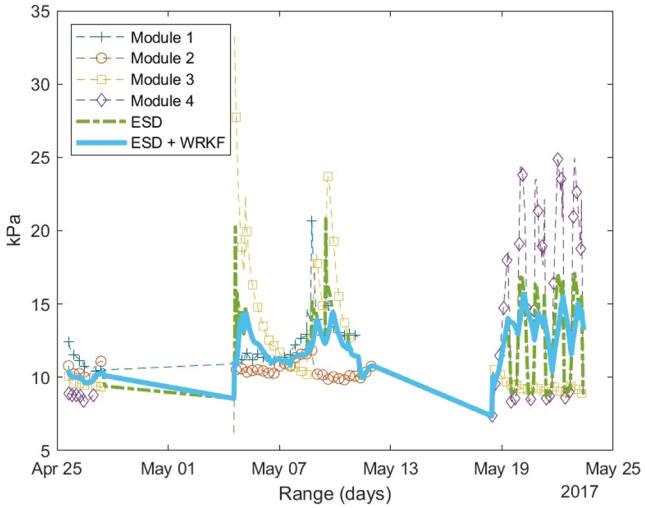

Fig. 20. Original and filtered data – ESD and ESD + WRKF (15 cm depth).

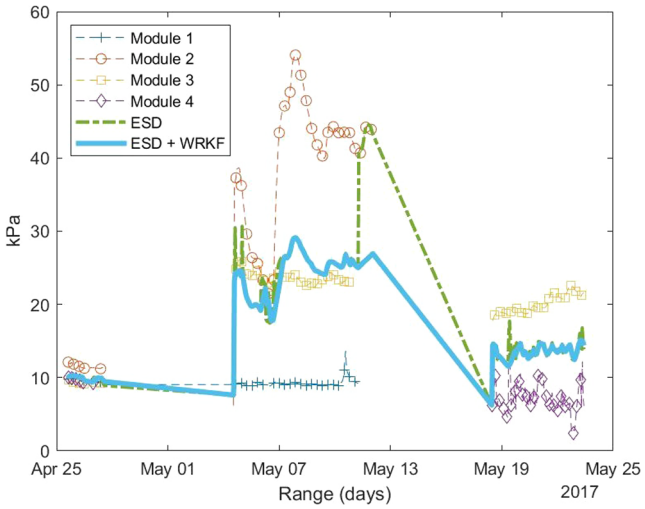

Fig. 22. Original and filtered data – ESD and ESD + WRKF (75 cm depth).

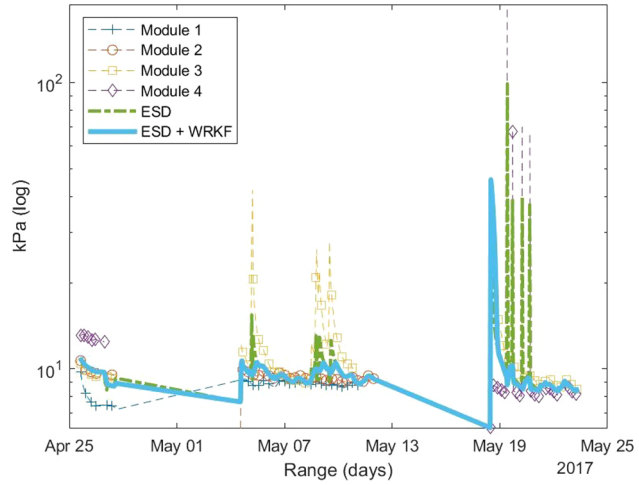

Fig. 21. Original and filtered data – ESD and ESD + WRKF (45 cm depth).

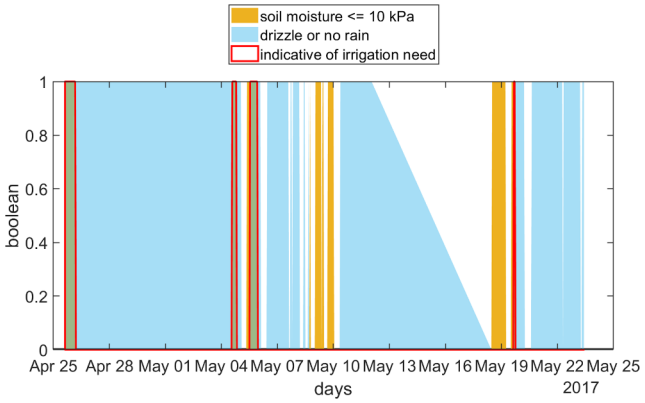

Fig. 23. Irrigation decision results.

context, we assessed that the WRKF filter is the most proper method since it achieved the greater resilience to the residual outliers and anomalous peaks of the data (possibly caused by faulty sensors).

Based on the data collected and on the experiment, we decided to adopt the ESD method in conjunction with the WRKF filter because they were able to detect trivial outliers and were not too sensitive as to remove relevant data. Both algorithms (ESD and WRKF) were also applied to the data from the second gathering session (coconut crop), more specifically in the medium-level fusion layer, to identify and remove outliers in the fusion of soil moisture sensors. Figs. 20–22 show the results for data from 15 cm, 45 cm, and 75 cm depth sensors, respectively. The ESD and WRKF filters were the most proper methods since it achieved greater resilience to the residual outliers and anomalous peaks of the data (possibly caused by faulty sensors).

7.4. Experiment III (Part I): Validation of the soil moisture application

This experiment focused on analyzing the two applications using the Hydra framework, based on data from the second data-gathering session, by using a data mining and machine learning software called Weka.[5]

The medium-level result of this application is a decision regarding

[5] Weka: https://www.cs.waikato.ac.nz/ml/weka/.

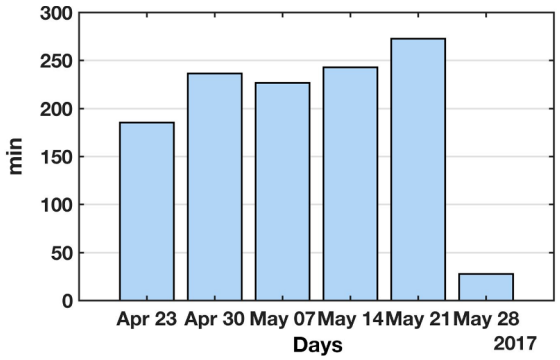

Fig. 24. Irrigation time decision results.

when the crop irrigation should start. This decision is based on rain information and soil moisture. The monitored period coincides with a period of constant rain, meaning that the soil moisture never got close to the reference irrigation level informed by the Embrapa specialist (60 kPa). However, to evaluate the decision making of the application, the soil moisture minimum level was set to 10 kPa, resulting in data presented in Fig. 23.

Our experiment highlighted the application correctly assessed the right moment to start irrigating the crop, only when the soil moisture data intersects with the light or no rain data, as explained in 5.1.4. That intersection is necessary because while the soil moisture sensor might indicate low moisture, it could be raining, and it takes some time for the water from permeating the soil. Therefore, deciding to irrigate based just on the soil moisture sensors is not ideal.

7.5. Experiment III (Part II): Validation of the evapotranspiration application

While the decision-making associated with the soil moisture application refers to the right moment to start irrigating the crop, the decision-making regarding the evapotranspiration application refers to the ideal irrigation duration for the crop. The ideal period is calculated by using data from sensors of the weather station to determine the ETo variable by the Penman-Monteith method.

By using the hourly ETo and crop parameters, we obtained the ideal irrigation time for the experimental coconut crop (Fig. 24). This decision is specific to this crop because it considers the irrigation efficiency (value provided by the Embrapa specialist) and a specific coefficient for this kind of crop.

7.6. Experiment III (Part III): Data validation

The purpose of this step is to validate the data generated by the sensor nodes and the developed applications (soil moisture and evapotranspiration) by checking their correlation to data from the weather station at Itapipoca (Ceará). For this purpose, we use a predictive method to generate the reference evapotranspiration (ETo) with all sensor data and weather station data and verify if its result is as reliable as the Penman-Monteith method. We performed 10-fold cross-validation with data from all sensor nodes and the weather station to define the training and test sets. We used the M5P classification algorithm, which combines decision trees with linear regression models, and it generated a decision tree with six rules, using solar radiation as the base parameter (Fig. 25). The model has a high correlation (0.9958) with the reference values generated by the Penman-Monteith model (Fig. 26), where soil temperature and soil moisture data were used in every rule, proving their relevance to ETo computation. We do not show all prediction models due to space constraints.

7.7. Experiment IV: New evapotranspiration model

Finally, Experiment IV aims to explore the possibility of using the collected data from the sensor nodes to improve the accuracy of an existing evapotranspiration model to be used instead of Penman-Monteith. We chose the Hargreaves-Samani model, considered adequate to the semiarid weather, and it requires only air temperature data but tends to overestimate ETo (da Silva et al., 2015). Due to the Hargreaves-Samani model not relying on as many sensors as Penman-Monteith (precipitation, wind speed, solar radiation, temperature, air humidity, air temperature, atmospheric pressure), our new model yields a reduction of implementation time and maintenance costs. Besides, our new model also yields an improvement in network resources consumption (processing, memory, communication, and energy) since few variables are monitored.

For this experiment, we analyzed the fused soil moisture data (obtained by the sensor nodes at Paraipaba) and air temperature data

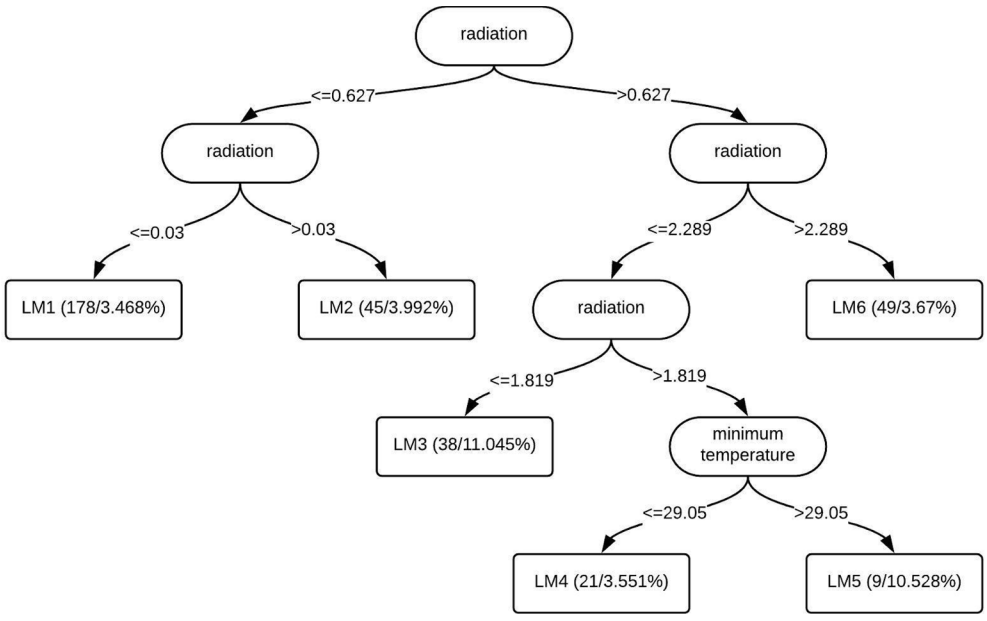

Fig. 25. Decision tree generated by the M5P algorithm.

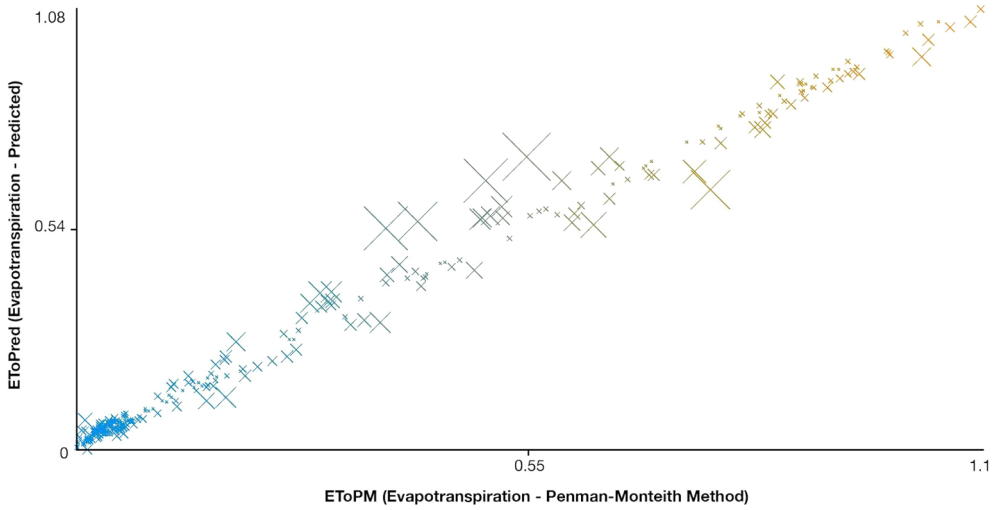

Fig. 26. Real ETo (x) and predicted ETo (y).

Table 8
Linear regression models results.

Method	RMSE	R2	MSE	MAE	Prediction speed (obs/s)	Training time (s)
Linear regression						
Linear Regression	1,12	0,48	1,25	0,9	~91	5,3134
With Interactions	6,58	−17,10	43,28	4,28	~130	0,92712
Robust	1,11	0,49	1,22	0,89	~260	0,93718
Stepwise	1,06	0,53	1,13	0,93	~210	18,881
Decision tree						
Simple	1,55	−0,00	2,39	1,14	~640	0,22344
Medium	1,55	−0,00	2,39	1,14	~840	0,21824
Complex	1,78	−0,32	3,15	1,42	~290	3,8375
SVM						
Linear	1,12	0,48	1,25	0,88	~420	2,6551
Quadratic	0,79	0,74	0,63	0,69	~590	0,26581
Cubic	0,85	0,70	0,72	0,72	~1000	0,17695
Fine Gaussian	1,50	0,07	2,24	1,04	~820	0,19472
Medium Gaussian	1,29	0,30	1,67	0,49	~1100	0,14973
Coarse Gaussian	1,37	0,21	1,88	0,93	~620	0,17864
Ensemble of trees						
Boosted Trees	1,57	−0,03	2,46	1,21	~260	1,5438
Bagged Trees	1,55	−0,00	2,39	1,13	~240	1,2767
Gaussian Process Regression (GPR)						
GPR Squared Exponential	1,33	0,26	1,76	1,00	~440	1,3399
GPR Matern 5/2	1,17	0,42	1,38	0,91	~720	0,29289
GPR Exponential	1,15	0,49	1,32	0,91	~800	0,28968
GPR Rational Quadratic	1,30	0,30	1,68	0,99	~860	0,2812

(obtained from the weather station at Itapipoca) by using Linear Regression. Table 8 lists the results of using different regression models. The best method was the Support Machine Vector (SVM) with a quadratic kernel, resulting in a model that obtained the best value of the root mean square error (RMSE) of 0.79 (the closer to zero the more accurate it is) and coefficient of determination (R2) of 0.74. The coefficient of determination (R2) is a statistical measure of how well the regression line approximates the real data points (Dantas Caminha et al., 2017), ranging from 0 to 1. An R2 of 1 indicates that the regression line perfectly fits the data.

Fig. 27 presents a comparison of ETo generated using Hargreaves-Samani, Penman-Monteith, and our proposed new model. Confirming the information provided by da Silva et al. (2015), Hargreaves-Samani overestimated ETo when compared to Penman-Monteith (correlation of 0.8169), and our new model achieved a more accurate result (correlation of 0.9299). Thus, we can provide a predictive model containing accuracy greater than Hargreaves-Samani. Based on this aspect, we can assume the fused data are valid and that it is feasible to create an evapotranspiration application with high accuracy even with the existence of an incomplete set of sensors in a weather station.

8. Conclusion

This work presented the Hydra framework, multilevel data fusion for the Internet of Things in Smart Agriculture. In partnership with Embrapa, we developed two applications based on Hydra architecture to monitor two experimental crops of coconut and precocious dwarf-cashew. In parallel with Hydra fusion architecture, we developed a low-cost IoT system using the MQTT protocol and a low-cost WSN infrastructure. The sensor nodes were also modified, and thus, we achieved a reduction in energy consumption.

Using Hydra fusion levels resulted in higher data accuracy, event detection, and automated decision-making. Moreover, we also use the gathered data to generate a new evapotranspiration model, which is accurate and requires few sensors when compared to the reference method. This aspect impacts in a reduction of resource consumption in the network such as processing, memory, communication, and energy since few variables need to be monitored. Besides, it allows a scenario without a weather station, with an incomplete database, or with a low number of sensors. Regarding the data, our proposal also improves data quality, improving the number of models, and the number of equations found since the data are correlated.

For the context of the smart agriculture domain using moisture soil sensors, we indicate that the best method to identify and remove outliers was a combination of the ESD method (Extreme Studentized Deviate) and WRKF filter (weighted outlier-robust Kalman filter). Moreover, the SVM (Support Machine Vector) quadratic machine-learning model generated an evapotranspiration model that resulted in values close to the evapotranspiration reference model (Penman-Monteith).

Future works will focus on improving the high-level decisions,

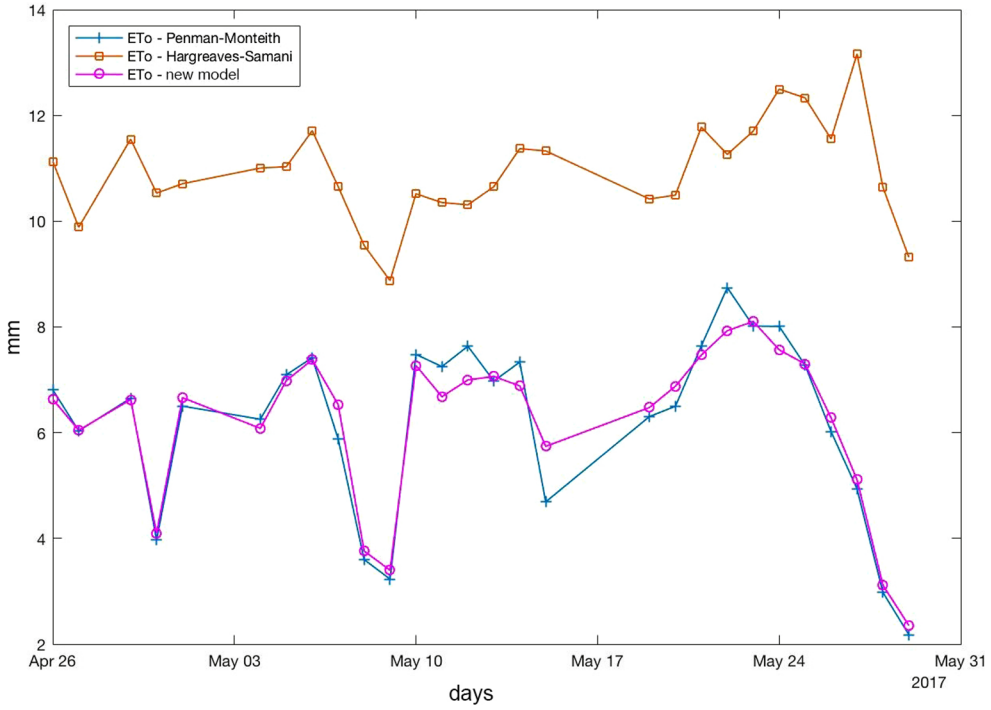

Fig. 27. ETo methods comparison.

composing of decisions, the priority of concurrent applications, and the priority of automatic actions. Furthermore, we focus also on gathering data for more extended periods to confirm the effectiveness of the new evapotranspiration model.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)—Finance Code 001. The authors also acknowledge the financial support of the CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico—Brasil, process #432820/2016-7) and São Paulo Research Foundation – FAPESP, through grant number 2015/24144-7.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.compag.2020.105309.

References

Adamchuk, V.I., Hummel, J.W., Morgan, M.T., Upadhyaya, S.K., 2004. On-the-go soil sensors for precision agriculture. Comp. Electron. Agricult. 44 (1), 71–91. https://doi.org/10.1016/j.compag.2004.03.002.. ISSN 0168–1699, http://www.sciencedirect.com/science/article/pii/S0168169904000444.

Allen, R.G., Pereira, L.S., Raes, D., Smith, M., 2006. Crop evapotranspiration - Guidelines for computing crop water requirements - FAO Irrigation and drainage paper 56, 56, FAO, Rome < http://www.kimberly.uidaho.edu/water/fao56/fao56.pdf http://www.fao.org/docrep/X0490E/x0490e00.htm > .

Anderson, R.G., French, A.N., 2019. Crop evapotranspiration. Agronomy 9 (10), 6–8. https://doi.org/10.3390/agronomy9100614. ISSN 20734395.

André, P.B., Andrade, A.T.C., Callegaro, R., Montez, C., Moraes, R., Pinto, A., 2017. An architecture for information fusion and for detection, identification and treatment of outliers in wireless sensor networks. In: Branco, K., Pinto, A., Pigatto, D. (Eds.), Communication in Critical Embedded Systems, vol. 702. Springer International Publishing, Cham, pp. 81–100, doi:https://doi.org/10.1007/978-3-319-61403-8_5, URL http://link.springer.com/10.1007/978-3-319-61403-8_5, ISBN 978-3-319-61402-1 978-3-319-61403-8.

Bish, S., Rohrer, M., Scheffel, P., Bennett, K., 2016. Multi-sensor fusion development. SPIE 9831, 98310T. https://doi.org/10.1117/12.2224138.. ISSN 1996756X, http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2224138.

Boström, H., Andler, S.F., Brohede, M., Johansson, R., Karlsson, A., Laere, J.V., Niklasson, L., Nilsson, M., Persson, A., Ziemke, T., 2007. On the definition of information fusion as a field of research. IKI Tech. Rep. (October), 1–8. http://www.isif.org/sites/isif.org/files/FULLTEXT01.pdf.

Cleveland, W.S., 1979. Robust locally weighted regression and smoothing scatterplots. J. Am. Statist. Assoc. 74 (368), 829–836. https://doi.org/10.1080/01621459.1979.10481038.. ISSN 0162-1459, 1537-274X, http://www.tandfonline.com/doi/abs/10.1080/01621459.1979.10481038.

Dantas Caminha, H., Coelho da Silva, T., Rego da Rocha, A., Vieira Lima, S.C.R., 2017. Estimating reference evapotranspiration using data mining prediction models and feature selection. In: Proceedings of the 19th International Conference on Enterprise Information Systems. SCITEPRESS - Science and Technology Publications, Porto, Portugal, pp. 272–279, doi:https://doi.org/10.5220/0006327202720279, ISBN 978-989-758-247-9 978-989-758-248-6 978-989-758-249-3 < http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006327202720279 > .

Dasarathy, B., 1997. Sensor fusion potential exploitation-innovative architectures and illustrative applications. Proc. IEEE 85 (1), 24–38. https://doi.org/10.1109/5.554206.. ISSN 00189219, http://ieeexplore.ieee.org/document/554206/.

da Silva, M.G., Oliveira, I.d.S., Carmo, F.F., Lêdo, E.R.F., da Silva Filho, J.A., 2015. Estimativa da Evapotranspiração de Referência pela Equação de Hargreaves-Samni no Estado do Ceará, Brasil. Brazil. J. Biosyst. Eng. 9 (2), 132–141. https://doi.org/10.18011/bioeng2015v9n2p132-141.

De Paola, A., Ferraro, P., Gaglio, S., Lo Re, G., Das, S., 2016. An adaptive bayesian system for context-aware data fusion in smart environments. IEEE Trans. Mob. Comput. 1. https://doi.org/10.1109/TMC.2016.2599158.. ISSN 1536-1233, http://ieeexplore.ieee.org/document/7539530/.

Hubert, M., Vandervieren, E., 2008. An adjusted boxplot for skewed distributions. Comput. Statist. Data Anal. 52 (12), 5186–5201. https://doi.org/10.1016/j.csda.2007.11.008.. ISSN 0167-9473, http://www.sciencedirect.com/science/article/pii/S0167947307004434.

Iglewicz, B., Hoaglin, D., 1993. How to Detect and Handle Outliers, ASQC basic references in quality control, ASQC Quality Press, ISBN 978-0-87389-247-6 https://books.google.ca/books?id=siInAQAAIAAJ.

Iyengar, S.S., Chakrabaraty, K., Qi, H., 2001. Introduction to special issue on distributed sensor networks for real-tie systems with adaptive configuration. J. Frank. Inst. 338, 651–653.

Luo, R., Yih, C.-C., Su, K.L., 2002. Multisensor fusion and integration: approaches, applications, and future research directions. IEEE Sens. J. 2 (2), 107–119. https://doi.org/10.1109/JSEN.2002.1000251. ISSN 1530–437X, 1558–1748, 2379–9153.

Martins, G., de Farias, C.M., Pirmez. L., 2018. Athena: a knowledge fusion algorithm for the internet of things. In: Proceedings of the 14th ACM International Symposium on

QoS and Security for Wireless and Mobile Networks, Q2SWinet'18. ACM, New York, NY, USA, pp. 92–99, doi:https://doi.org/10.1145/3267129.3267141 < http://doi.acm.org/10.1145/3267129.3267141 > , ISBN 978-1-4503-5963-4.

Nakamura, E.F., Loureiro, A.A.F., Frery, A.C., 2007. Information fusion for wireless sensor networks: methods, models, and classifications. ACM Comput. Surv. 39 (3). https://doi.org/10.1145/1267070.1267073.. 9–es, ISSN 03600300, http://portal.acm.org/citation.cfm?doid=1267070.1267073.

National Water Agency (ANA), Brazilian Water Resources Report - 2017, Full Report, Ministry of Environment, Brasilia, Brazil, 2018 < http://www.snirh.gov.br/portal/snirh/centrais-de-conteudos/conjuntura-dos-recursos-hidricos/conj2017_rel_ingles-1.pdf > .

NIST/SEMATECH, 2013. e-Handbook of Statistical Methods < https://www.itl.nist.gov/div898/handbook/ > .

Ross, S., 2003. Peirce's criterion for the elimination of suspect experimental data, J. Eng. Technol. 20.

Savitzky, A., Golay, M.J.E., 1964. Smoothing and differentiation of data by simplified least squares procedures. Anal. Chem. 36 (8), 1627–1639. https://doi.org/10.1021/ac60214a047.. ISSN 0003-2700, 1520-6882, https://pubs.acs.org/doi/abs/10.1021/ac60214a047.

Shock, C.C., Barnum, J.M., Seddigh, M., 1998. Calibration of watermark soil moisture sensors for irrigation management. In: Proceedings of the 1998 Annual Meeting of the Irrigation Association, pp. 139–146.

Taylor, J., 1997. Introduction To Error Analysis: The Study of Uncertainties in Physical Measurements, ASMSU/Spartans.4.Spartans Textbook, University Science Books, ISBN 978-0-935702-75-0 < https://books.google.ca/books?id=giFQcZub80oC > .

Ting, J.-A., Theodorou, E., Schaal, S., 2007. A Kalman filter for robust outlier detection. In: 2007 IEEE/RSJ international conference on intelligent robots and systems. IEEE, San Diego, CA, USA, pp. 1514–1519, doi:https://doi.org/10.1109/IROS.2007.4399158, http://ieeexplore.ieee.org/document/4399158/, ISBN 978-1-4244-0911-2 978-1-4244-0912-9.

Torres, A.B.B., Filho, J.A., da Rocha, A.R., Gondim, R.S., de Souza, J.N., 2017. Outlier detection methods and sensor data fusion for precision agriculture. In: IX Simpósio Brasileiro de Computação Ubíqua e Pervasiva - SBCUP.

Wang Jun, Gao Yuan, Ran Chenjian, Huo Yinlong, 2015. State estimation with two-level fusion structure. In: 2015 International Conference on Estimation, Detection and Information Fusion (ICEDIF), pp. 105–109, doi:https://doi.org/10.1109/ICEDIF.2015.7280171.

Wichit, N., 2014. Multisensor data fusion model for activity detection. In: 12th International Conference on ICT and Knowledge Engineering.

Witkin, A., 1984. Scale-space filtering: a new approach to multi-scale description. In: ICASSP '84. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 9. Institute of Electrical and Electronics Engineers, San Diego, CA, USA, pp. 150–153, doi:https://doi.org/10.1109/ICASSP.1984.1172729, http://ieeexplore.ieee.org/document/1172729/.