SIE - An intelligent and automatic irrigation system to optimize water resources

Cristhian Gomez, Fabio Tovar, Miguel Ortiz, Oscar Martinez

Bogotá, Colombia

Abstract

The optimal use of water has been one of the main topics in most fields in the productive sector and especially in agriculture. We propose a solution aimed at mitigating the consequences of this problem, trying to optimize the water amount used in irrigation through predictions with Machine-Learning based in sensor readings and the use of an optimal irrigation system, such as drip irrigation.

This system needs real-time information and predictions, so that must be supported with a background network with a good reliability, low cost and low power consumption. These reasons led to selection of Xbee technologies together with MQTT-SN protocol as the best way to face this challenge

As result, we obtained a system that presents lower water consumption than the traditional systems or methods to irrigate, predicting the ideal moments to do this, based in the environmental conditions. Additionally, this system has low implementation cost ($335.25 \times n + 137.8$ dollars for a plot with n subplots of $269ft^2$) being affordable for most farmers.

Keywords: Irrigation system, Precision Agriculture, Sensor networks

1. Introduction

1.1. Current scenario

Water is the main element for life because it is essential for the development of most vital processes of living beings. However, in recent years, due to the lack of awareness in certain productive processes, water has begun to be scarce. According to the Food and Agriculture Organization of the United Nations (FAO), agriculture represents seventy percent of global water extraction [1], in addition, lack of water affects four out of ten people in the world as reported by the World Health Organization (WHO),showing the importance of addressing the problem of waste of water in agriculture in an effective way.

Although Colombia is one of the countries with more fresh water in the world, there are several regions where people still have difficulty finding it. The reasons are multiple. Figure 1 illustrates the problem that this country is facing.

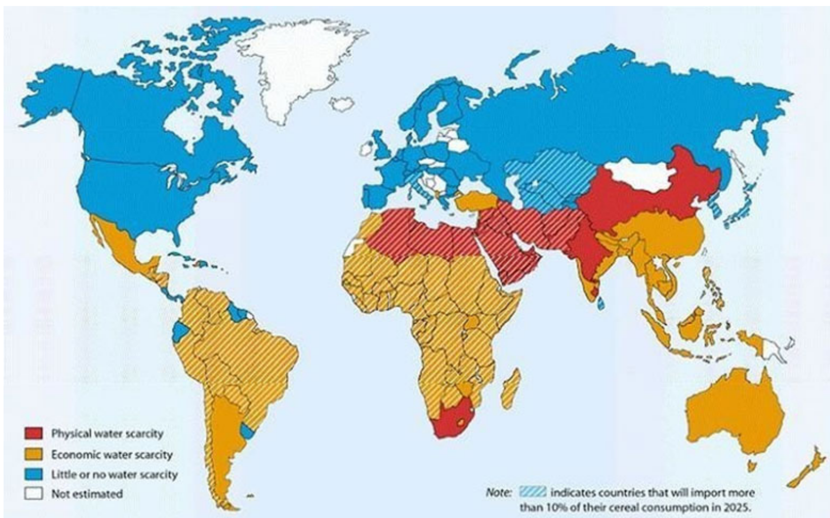


Figure 1: Water scarcity in the world [1]

1.2. Project motivation

Techniques used in agriculture and resource extraction are rudimentary in many places of the world - especially in countries that are not economically favored - with consequences such as environmental damage.

Countries like Colombia don't usually adjust their agricultural practices despite the fact that in agriculture the main resource used is water, implying that it is necessary to optimize the use of this resource to make agriculture sustainable in the long term. Therefore, the optimization of irrigation and water resources is proposed as a solution to the problem, through the design and implementation of intelligent and automatic irrigation systems.

1.3. Overview

In the present document, we are introducing SIE (name of the Chibcha goddess of water) as a proposed solution to the need of optimizing water usage in agriculture by implementing an intelligent and automatic irrigation system using IoT and machine learning.

In the first place, we deploy a network that communicates sensors of humidity, soil moisture, temperature, raindrop and water flow meters with a Raspberry Pi using *MQTT-SN (MQTT for Sensor Networks)* protocol and *Digimesh*, which is a mesh protocol developed by Digi for Xbee Modules.

We built a module for the Raspberry Pi, which is responsible for processing the sensors' data. This module receives the MQTT-SN packets and processes them to extract the data and feed a random forest model to produce predictions about the times that the system must irrigate based on the environmental conditions and the irrigation patterns dictated by the farmer or agronomist responsible for the crop.

The process described above will feed SIE irrigation system with the help of actuators as electrovalves to control the flow of water for the crop.

The data that we used were soil moisture, air temperature, humidity, and the timestamps at which it was irrigated. Additionally, we used the water temperature and the water flow to estimate water consumption.

We claim that this system can reduce the water use for crops in comparison with the classical ways to irrigate, the level of water savings can vary according to the crop environmental characteristics, type of crop, station, among others.

2. Analysis and Design

The objective of any irrigation system is to provide the necessary amount of water at the right time to each of the plants of a crop. To make a good selection of an irrigation system it is necessary to consider several factors, which include the topography and the characteristics of the land on which the crop is going to be established; the type of soil and its water-related characteristics (porosity, capacity of drainage, availability, quality and purity of the resource); and, naturally, the type of crop and the water requirements that are demanded in each stage of its development.

The first step in the design of the project was to establish a context in which the system was used and for practicality, we think of a crop with approximately 269 ft^2 as shown in Figure 2 to model the entire system.

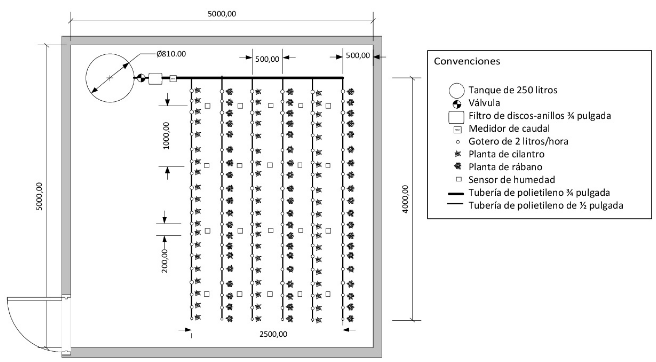


Figure 2: Standard crop with irrigation system

This arrangement of the irrigation elements allows us to have total control of the amount of water supplied to the crop and obtain accurate data about the environmental conditions.

Secondly, we selected an irrigation system. Among the irrigation systems, drip irrigation stands out as the most efficient in water management. This system offers a slow distribution of the resource and allows a better use of water by the plants, which translates into an increase in crop yield. It also represents a considerable reduction in the volume of water lost through runoff and percolation and can be adapted to any type of terrain, even those with a high slope. This system, in comparison with the others that are used in agriculture, requires a higher initial investment, however, given its efficiency (between 90 and 95 percent), it recovers quickly as long as the right conditions are maintained for its good operation. It also requires a high purity of the water with which the crop is irrigated due to the very small diameters of the exit holes of the drippers that can be plugged with tiny solid particles present in the water; that makes indispensable the use of a filter that traps those particles.

As a case point,in Niger, drip irrigation was compared with irrigation with watering cans and was shown the drip irrigation is more optimal and helps to achieve higher yields, higher returns to water and higher returns to labor. [2]

In addition, there are several advantages over other irrigation systems, such as helping farmers to conserve fertilizer since it can be applied through the drip irrigation system. [3]

Knowing this, the drip irrigation system was chosen with the intention of implementing machine learning in its design and operation, in such a way that it is possible to partially automate the agricultural production of diverse crops and reduce the time and labor demand for the farmers.

Regarding to communications, we implemented a low-cost wireless network, because sometimes it is hard to find land with access to electric power in many places in Colombia, so a wired system would imply a hard deployment and higher costs.

By this line, we explore some approaches of WPAN networks (IEEE 802.15.4 standard) as Zigbee and Digimesh, Bluetooth, and Bluetooth Low Energy (BLE). We found that BLE has the best performance in a cyclic sleep scenario, due its low energy consumption and the optimization of battery life, followed by Zigbee and ANT [4]. However, we chose Digimesh, a protocol based in Zigbee technology because it offers the possibility of having a mesh network, greater range of transmission, and the difference in consumption is not so considerable.

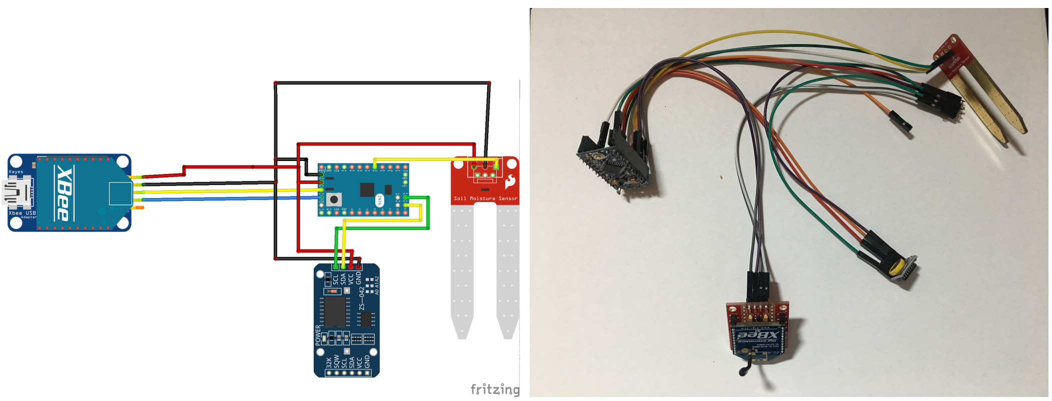
2.1. Internet of Things

In order to accomplish an effective communication between sensors, controllers and actuators, MQTT for sensor networks (MQTT-SN), Zigbee and Mosquitto were used in this project.

Digimesh is a standard for wireless technology that complies with IEEE 802.15.4 standard for communication in wireless personal area networks (PAN) and is based on Zigbee. Digimesh has the characteristic of being a low cost technology, with low power consumption, simple protocol and high density of nodes. The technology uses a single device type, instead of the multiple device types of Zigbee. All of these devices have the ability to connect to each other as long as they are in the supported range (30m indoor and up to 90m outdoor) [5].

To provide the Xbee modules with the ability of sending accumulated data in certain periods and make actuators perform a certain operation as we want, it is necessary to connect all the components with a micro-controller that can synchronize all these operations with low-cost and low energy consumption. Finally, we chose Arduino Pro Mini as micro-controller board and we developed the following components:

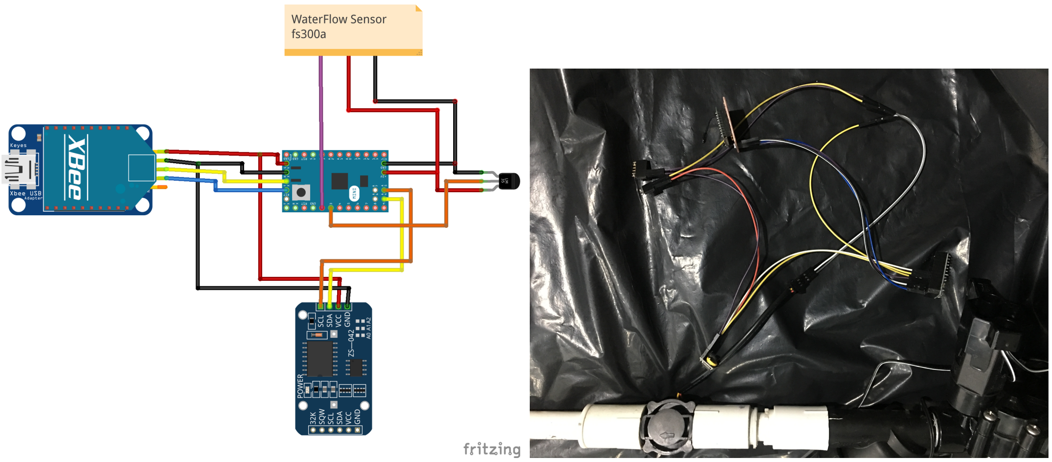
1. **Crop Controller.**
   Goal: It is in charge of measuring the soil moisture of the crops, label with a timestamp and send via MQTT-SN to the gateway.
   Elements: Xbee module, soil moisture sensor, real time clock



(a) Diagram (b) Picture

Figure 3: Crop controller overall view

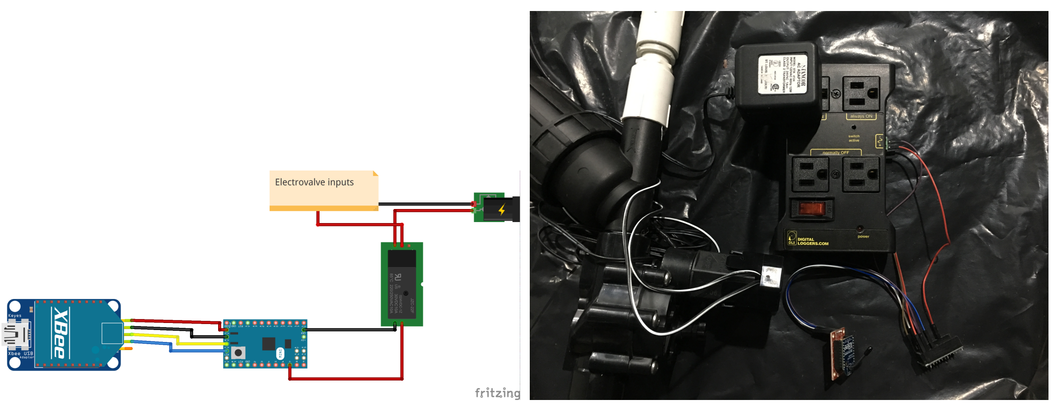
2. **Tank Controller.**
   Goal: It is in charge of measuring the water flow and temperature of the water supply, label with a timestamp and send via MQTT-SN to the gateway.
   Elements: Xbee module, water temperature sensor, water flow sensor, real time clock.



(a) Diagram (b) Picture

Figure 4: Tank controller overall view

3. **Flow Controller.**
   Goal: It is in charge of controlling the electrovalve to allow the passage of water when it is spliced. This component handles communication via MQTT-SN with a broker and a specialized client.
   Elements: Xbee module, electrovalve, relay.

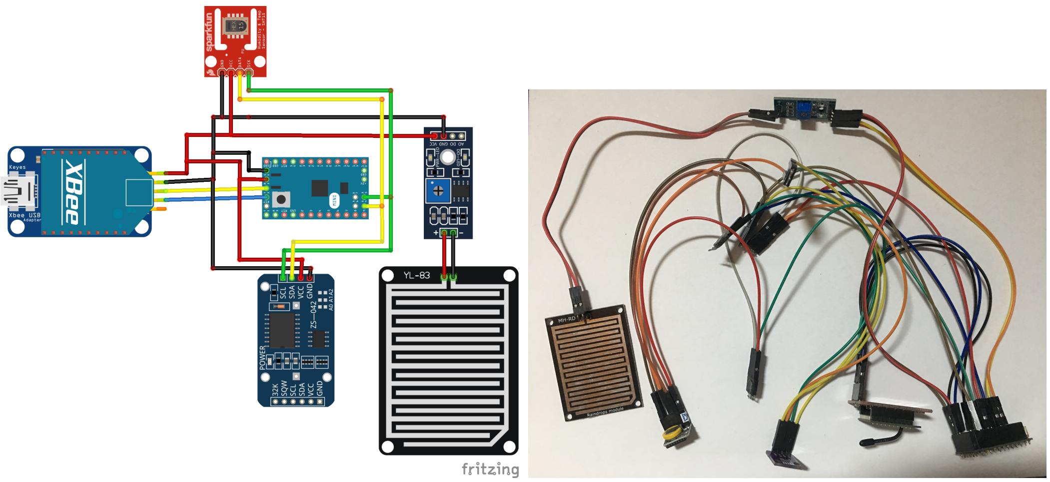


(a) Diagram (b) Picture

Figure 5: Tank controller overall view

4. **Environment Controller.** Goal: It is in charge of measuring all the important data of the environment, as the rain frequency and the temperature and humidity.

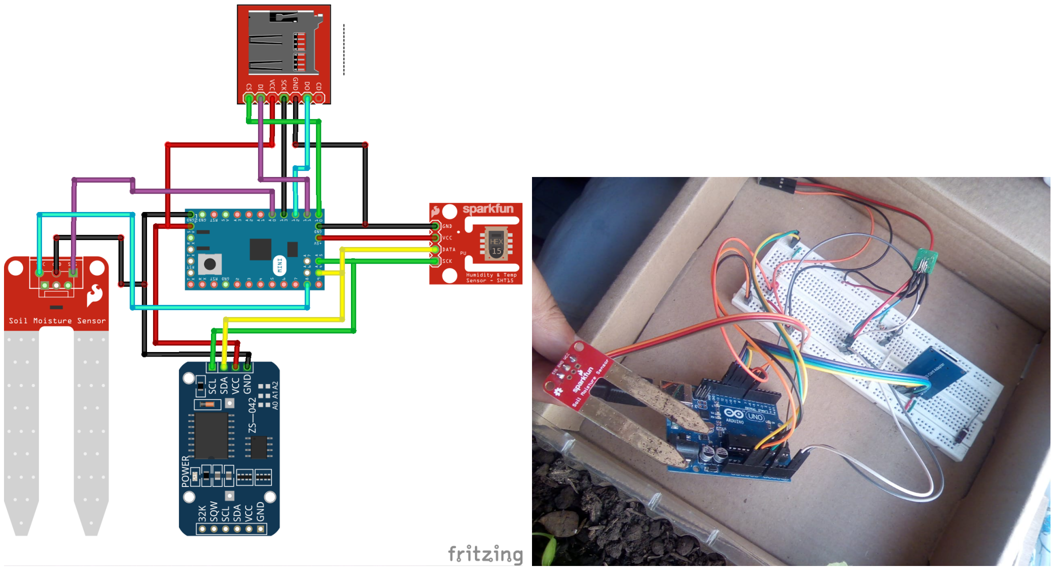
Elements: Xbee module, rain drop sensor, temperature and humidity sensor, real time clock.



(a) Diagram (b) Picture

Figure 6: Environment controller overall view

5. **Data-logger Controller.** Goal: It is in charge of measuring and saving data of the test scenarios to train the first models to predict the optimal time to perform the irrigation.
   Elements: Xbee module, humidity and temperature sensor, soil moisture sensor, real time clock, microSD module.



(a) Diagram (b) Picture

Figure 7: Data-logger controller overall view

Now, we talk about MQTT-SN, which is a publish/subscribe protocol for wireless sensor networks. This protocol is optimized for implementation on low cost, battery-operated devices with limited processing and storage resources. MQTT-SN has three types of components, MQTT-SN clients, MQTT-SN gateways, and MQTT-SN forwarders [6]. MQTT-SN clients use MQTT-SN gateways and MQTT-SN forwarders to connect to a message broker like Mosquitto over Internet. MQTT-SN makes use of UTF-8 strings called topics to filter messages for each connected client, in other words a MQTT-SN client can publish on a certain topic and only the clients subscribed to this topic will see the message .

Mosquitto provides a server implementation of the MQTT protocol, being as minimal and efficient as possible. We use Mosquitto as a message broker in the following way, each controller sends data via MQTT-SN over the Digimesh network. These messages are received by the Raspberry Pi through a Xbee module, which performs as MQTT-SN module, this publishes the data in the appropriate topic and store it in MongoDB.

2.2. Machine Learning

Given that the project is aimed at helping farmers save money in water expenses and reduce efforts, the motivation to use machine learning in this project was to develop an Expert System or Decision Support System in a short Time to Market to automate and help the irrigation decision process.

The objective of the system is to learn the tacit knowledge of the farmers. This means that the farmers use SIE to trigger irrigation and the model learns from this and the feedback given by the farmers based on the predictions.

Random Forests are used with good results in literature [7]. Decision Trees resemble an Expert System in how choices are made. In fact, some Expert Systems are built with manually constructed decision trees.

Throughout the project there have been advances in the machine learning models before and after the arrival of new data, though mostly at the arrival of such data as it allows the inclusion of more features and further development of the model and more importantly to make more tests.

2.2.1. Data Collection

Through two weeks, the data logger controller was set in a collection scenario (which will be described in section 3), gathering data in a local device. An agronomist checked daily the scenario and irrigated when it was necessary, taking note of the time in which irrigation was performed and measuring the volume of water used.

2.2.2. Characterization of the Data

The Data collected has a non-stationary distribution, this means as times passes the variance of the features changes and the model will lose precision and would need to be retrained or updated. This suggests the need for an online model or retraining every certain time. This is due to the varying biophysical conditions which are hard to predict. This also suggests that for a different setup in a different place or with different type of crops the model and data need to be different.

Table 1 shows the features that were fed into the model as of the first and current physical implementation of the project.

Date data is processed into categorical Features 'WeekDay', 'Hour' and 'Minute' this leads to a total of 7 features.

Feature	Description
Soil Moisture	Moisture is one of the main features; it helps us tell when irrigation is done
Humidity	Amount of water vapor in the air
Temperature	Temperature of the air
Date	The moment at which the other features are collected
TimeSinceLast	Total seconds since last irrigation. The model learns how often should it irrigate based on this feature and other Biophysical conditions
Irrigation	This is the target feature, it is marked when the irrigation button is pressed or when the system automates the call to the process

Table 1: Brief description of features

The irrigation label is not only applied to the exact moment of irrigation given by the farmer input but to a range of hours prior to irrigation. Before the counter since last irrigation is reset. This allows the model to learn about prudent irrigation intervals based on how much time has passed since last irrigation and the bio-physical conditions.

The Correlation Matrix (Figure 8) provides some intuition into our data set, the co-relation rate is represented by the colors in the left palette, where the top red values are a positive co-relation and the blue bottom values represent a negative co-relation.

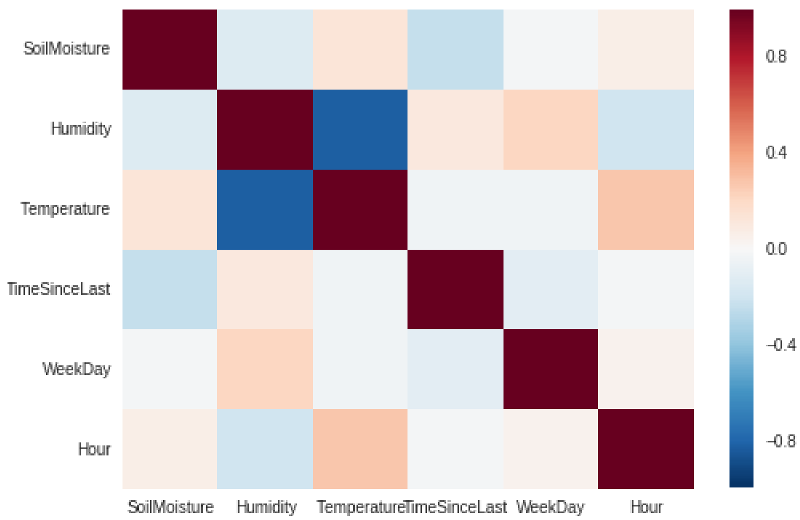


Figure 8: Correlation Matrix of the numerical Features

The pair plot of the data in Figure 13 shows distribution of every feature along with a scatter vs plot of every pair of features.

Before proceeding into the model, it is important to note that the data has a grave unbalance. Only 7% of the data has a positive label. This is handled by providing additional parameters to the model.

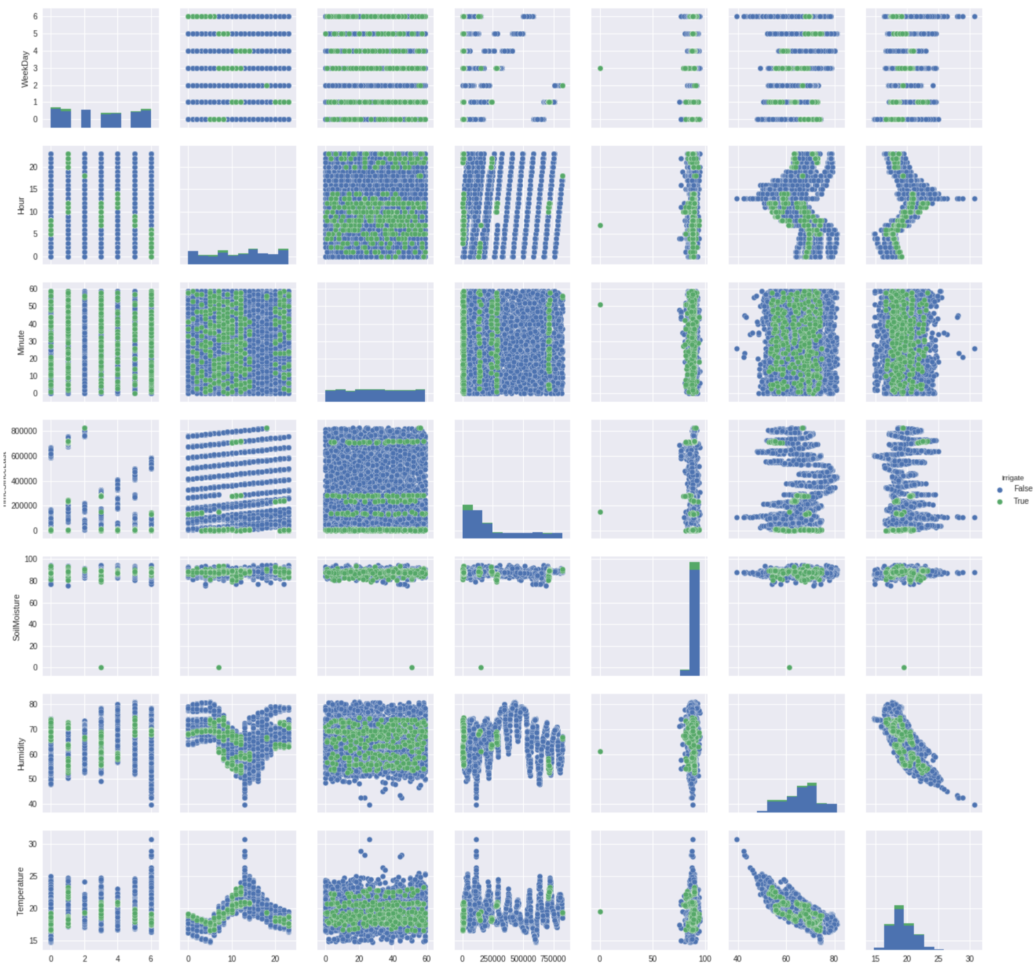


Figure 9: Pair plot of data distribution

2.2.3. Characterization of the Model

We want to know when it is the best time to irrigate, so that we must choose a classifier model that allows us make these predictions. As we say before, we used the Random Forest model, and specifically, the implementation of sci-kit learn, a popular Machine Learning library. Random Forest is an ensemble of decision trees, so that you could see it as a forest where every tree votes on the label of the predicted example and the majority wins. This reduces the variance of the predictions by as much as the number of trees used.

Entropy and Gini-impurity are the most popular criteria for it helps in the selection of attributes used at each node of the tree to split data, in building of the forest trees. The definition of both functions are in Eq. (1).

$$\begin{aligned} Gini(E) &= 1 - \sum_{j=1}^{c} p_j^2 \\ Entropy(E) &= - \sum_{j=1}^{c} p_j log p_j \end{aligned} \tag{1}$$

Selection of the criterion is a difficult task because there are no formal arguments to select one or the other, Laura Elena Raileanu and Kilian Stoffel say that the criteria disagree only in 2% at most of all cases, so that is not possible to decide which one performs better [8]. However, in computational resources, the Gini function should perform better than Entropy because is more difficult to calculate a log than a power.

So, we chose the Gini criterion for this project, the objective of the function is to minimize Gini impurity in the splits of the data for any of the features.

As the data is unbalanced, the model needs to be balanced to prevent the model to be biased toward the dominant label, this can be done in the scikit learn Random Forest implementation by using the "balanced" or "balanced-subsample" options for the class-weight parameter of the Random Forest.

2.2.4. Hyper-parameters tuning

Fitting the model to the data was an expensive task. We started with the use of Cross Validation Grid Search technique, which allows us test the model with all the hyper-parameters combination that we want, to find the best fit of the model according to the results of the cross validation set.

With this initial approach, we were able to determinate some constraints to data tuning. For example, for a depth greater or equals than six for each tree as you can see in Figure 10, overfitting occurs in the model due to the direct relation between the complexity of the ensamble and the depth of the trees.

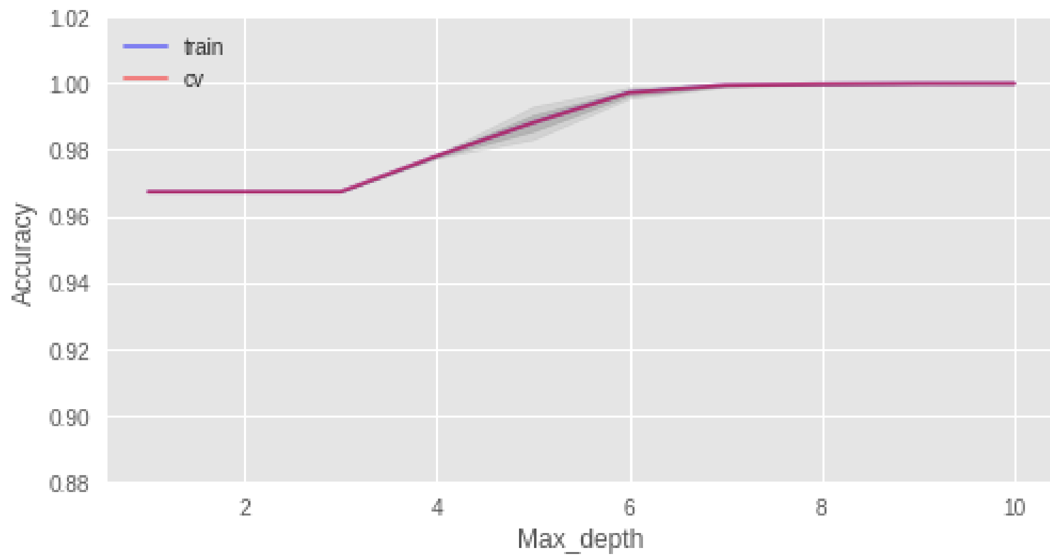


Figure 10: Relation between accuracy and depth of the trees

Finally, the hyper-parameters selected for the final training of the Random Forest Classifier were the following:

1. The number of the maximum trees in the forest was determined by the hyper-parameter **n-estimators**, which was set to **25 trees**. [9]
2. The maximum depth of each tree was determined by **max-depth** hyper-parameter, which was set to **4**. [9]
3. The minimum number of samples required to be at a leaf node **min-samples-leaf** was set to **7** [9]

2.2.5. Feedback Loop

As time passes and the model loses accuracy it needs to be retrained through a feedback loop, only when its predictions are labeled as wrong, similar to reinforcement learning.

If the model were to be retrained on its own predictions the feedback would be biased and thus the model would converge into over-fitting of what it has initially learned.

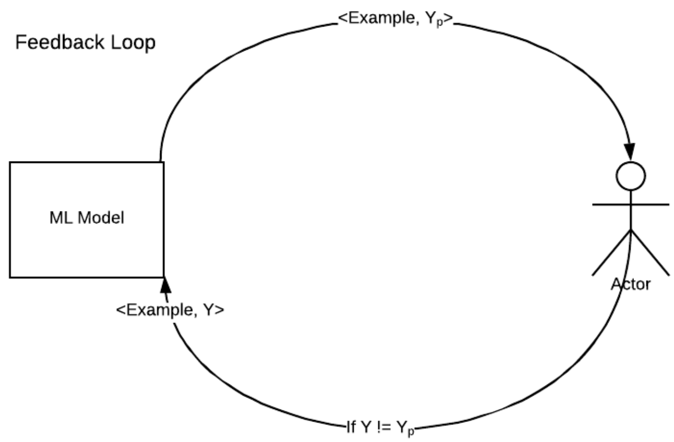


Figure 11: Feedback model

2.3. Integrated system

To integrate the previously designed systems to work together, the Raspberry Pi 3 was used as a low-cost interface between the environment data collected by the sensors and their storage in cloud and the Machine Learning model that guide SIE for an optimum use of water.

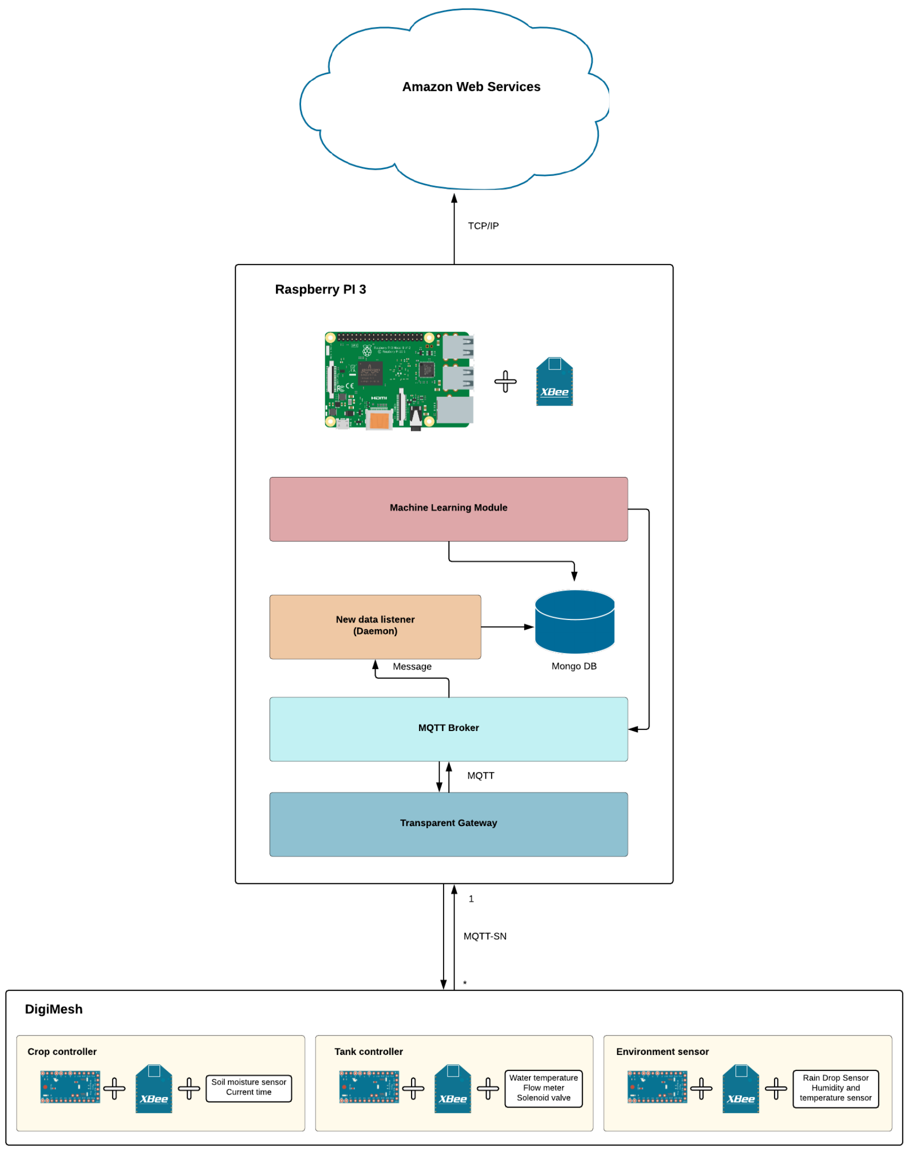


Figure 12: SIE General Architecture

We decided to expand the architecture with cloud services, since the Raspberry Pi 3 does not have large processing capacity or storage and the model evolution and the collected data would break these constraints. Therefore, we use Amazon Web Services to save the data only, and this interface provides us with the possibility of developing new services and features in cloud that we believe are necessary to improve the performance of the system as it evolves.

In the Digimesh design, we model different components to make the architecture easily scalable, allowing the farmer or agronomist to add new sensors without disconnecting or affecting any part of the system due that all of them are independent.

3. Testing

3.1. Collection scenario

The collection scenario is composed of two small scale crops with a Data-Logger controller that records all the changes in this environment. Measurements are taken every five minutes and they are saved in a microSD memory.

We use coriander and radish as the two test crops for data monitoring and analysis, as well as training the models and tuning the parameters.

3.2. Testing scenario

The tests were performed in an indoor setup, measuring 42x27 inches. We made two grooves, one for coriander and one for radish.

The first scenario is the classical scenario where the irrigation is done using watering cans. The second scenario is the test scenario to SIE irrigation system. In this scenario, there are the following deployment components: Crop Controller, Environment Controller and Tank controller. We installed a drip irrigation system that is controlled via MQTT-SN over Xbee with the Tank Controller.

4. Results

4.1. Machine Learning model

The Random Forest Classifier model behaved well on our data set. The metrics resulting after training are in Tables 2, 3, where it is shown that the decision to irrigate has a high precision but a relative low recall value, this means that the model predicts the times to irrigate with accuracy, but skips some of them.

Training	Precision	Recall	F1-score	Support
Do not irrigate	0.994	0.998	0.996	483192
Irrigate	0.974	0.915	0.944	36684
avg / total	0.992	0.992	0.992	519876

Table 2: Metrics results over the training set

Validation	precision	recall	f1-score	support
Do not irrigate	0.994	0.998	0.996	482998
Irrigate	0.973	0.917	0.944	36878
avg / total	0.992	0.992	0.992	519876

Table 3: Metrics results over the validation set

For testing we isolated 2 days of data when irrigation was done by the agronomist, data never seen before by the model in comparison to the validation and training sets, which shared the same origin but were randomly split with a validation size of 60%. Small training size is used due to the ease of over-fitting we didn't want our model to learn very specific rules, but more general rules. A bigger training size would result in getting 0 predictions in the test set.

4.1.1. Feature Importance

Feature importance is a way to get some intuition into the way the decision trees are using the features to make the splits in the data. and also lets us know if there is a label leak (for example when a feature is the only feature used by the model to tell which label to assign).

Ranking of Feature Importance

1. TimeSinceLast 0.307030
2. Temperature 0.215345
3. WeekDay 0.163684
4. Humidity 0.159223
5. Hour 0.126863
6. SoilMoisture 0.024723
7. Minute 0.003131

This ranking gives some insight into what our model is really learning. The four more important features suggest us that our model is learning to irrigate at moments where temperature is low, late at night. This corresponds to the knowledge proposed by the agronomist as he has suggested that irrigation at low air-temperature means that water will not evaporate from the soil so easily.

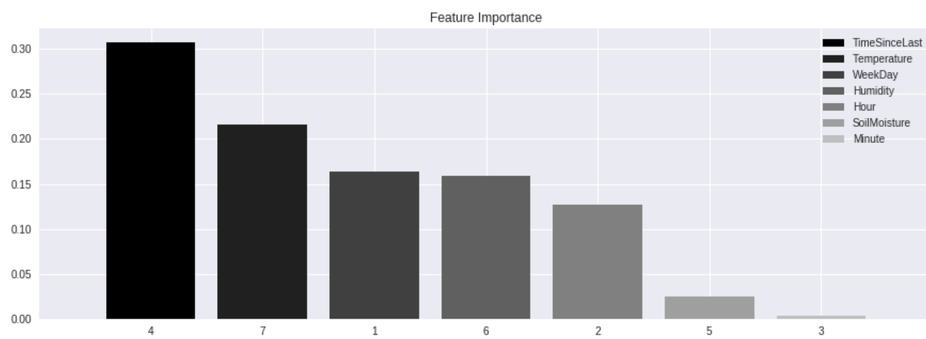


Figure 13: Feature Importance

4.2. SIE system

The SIE system was placed in operation in one of the two crops of the testing scenario. Each of these crops has two grooves of coriander and radish, that were planted five days before at the same time, and by this time, the crops were irrigated by an agronomist.

After placing the system in operation, the agronomist continued to irrigate the crops that do not have SIE installed. The Tank Controller, Environment Controller and Crops Controller were working and sending the data to the database in the Raspberry Pi.

The predictions related with the data input were stored in the Raspberry PI to show the behavior of the model in a real environment. In Figure 14 are the positive predictions obtained by hour, the sensors send data each five minutes approximately. These results show that it is necessary to make a second validation after predicting, to some isolated irrigation signals with lower frequency.

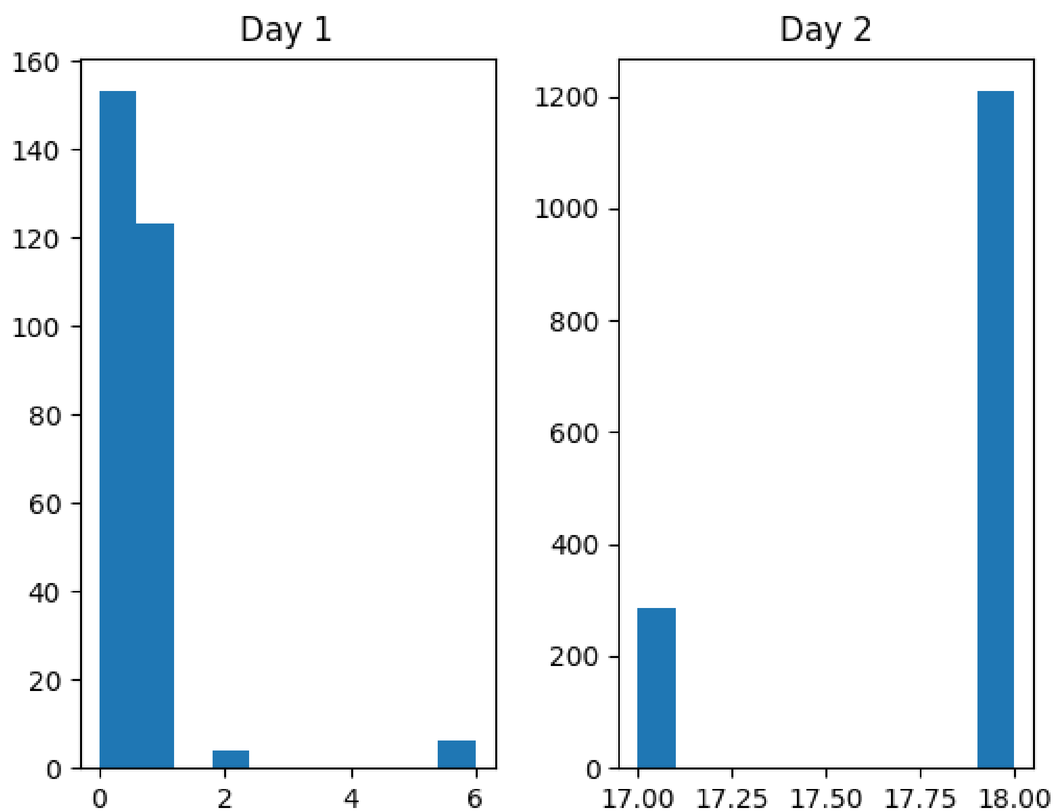


Figure 14: Times that the model predict irrigation by hour

On other side, the agronomist who irrigates the second crop, added 500 ml of water at 10:00 in day 1, and 400 ml of water at 18:55 in day 2. The Tank Controller registered 756,94445 ml of water consumption in both days, so that SIE saved 15,89% of the hydric resource. Taking in account that the system had a bad behavior in two moments in Day 1, the water saving would really be higher.

4.3. SIE costs

In Colombia few companies offer the service of automatic irrigation systems, but these have inflated prices, which are not accessible by farmers. As stated in section 2.1 SIE is composed of several controllers, the following list shows the retail acquisition cost of each component.

1. Crop Controller USD \$37.40
2. Tank Controller USD \$52.85
3. Environment Controller USD \$34.95
4. Flow Controller USD \$36.95

Each plot of $269ft^2$ is proposed to be divided in three subplots, in which two Crop Controllers and one Flow Controller are used, additionally for each tank we used one Tank Controller and for each system one Environmental Controller and a Raspberry Pi 3. Then if a plot has n subplots of $269ft^2$, $335.25 \times n + 137.8$ dollars are expected to be spent.

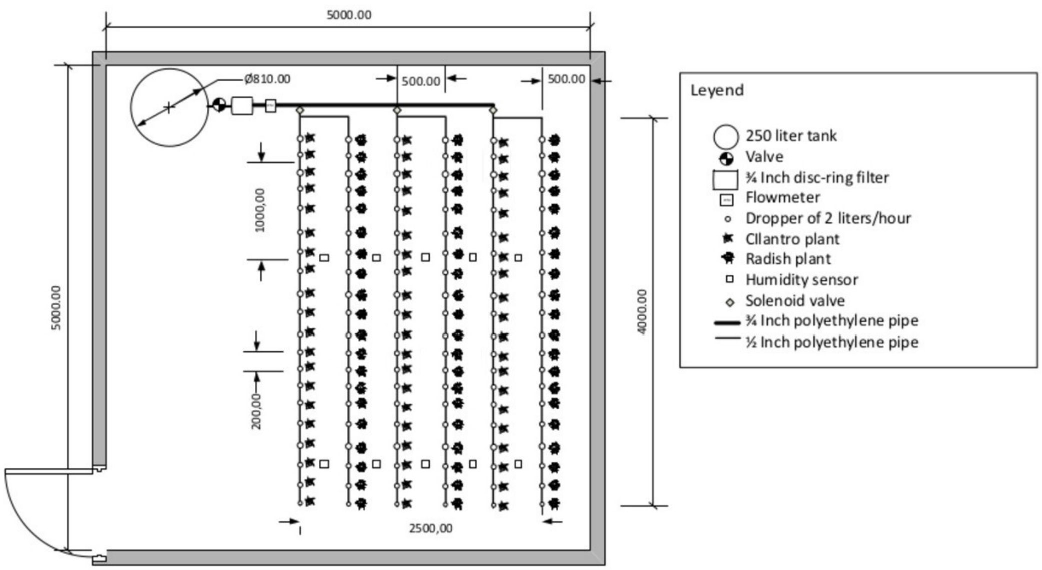


Figure 15: Proposed system arrangement in a base crop

5. Conclusion and Evaluation

SIE can be implemented with real crops, providing the functionality and results expected by the farmer, being a good and low cost alternative for irrigation automation and improvement of water use. Furthermore, MQTT-SN and Xbee technology are a good combination to communicate controllers in a low cost, efficient, fast and reliable way.

Regarding the machine learning model, a lot was learned but still there is much to work on. Random Forests and decision trees are good environmental models, but they are not the only ones used with good results [10]. For this system and data Random Forests gave a good out of the box result. We plan to try other models once more testing and refinement is done to the current model.

Generalization of the model to the test set needed low hyper-parameters e.g low number of estimators (5-50) and low depth of the decision trees (4-6). We think this may be related to the size of the dataset and the small feature set, with the inclusion of more data and more features along with more tests and simulations as described in the following section a more natural generalization would be achieved.

We think Precision and Recall metrics as is are not that good for the model at hand, since for labelling what we did was taking the moment at which irrigation was done and label all the examples within 5 hours prior to the irrigation moment and labelled them as an "Irrigate" moment. We still need to develop "interval-metrics" for the project. Still the model achieved good traditional metrics results on validation-data.

We didn't want our model to learn a precise and exact moment to irrigate but more prudent intervals to irrigate. For now the metric that we used was simulation. We took the model and saw the intervals at which the model predicted irrigation on the never-seen before test data. Only for smaller model (low-hyperparameters) the model predicted some moments to irrigate around 1-2000. For bigger models all test examples were ruled out, this suggest over-fitting for the bigger models.

A better metric and feedback is needed for true testing and simulation. What we mean by this is that we require good traditional metrics on training and validation data, but for test data we need continuous learning feedback, this means that the system will suggest moments to irrigate and the farmer or user will approve or disapprove of them.

If all suggested moments are good for the user then our model is good, if some are not then the model needs to be retrained without forgetting all it is has learned. This continuous learning idea is taken from [11] which main idea is the feedback loop, Gupta uses this idea to predict spam twits for enterprises, where that data also has a drifting, non-stationary distribution.

6. Further Work

1. Environmental data are also being collected every 10 minutes with requests to open weather API. Although it is not currently used in the model, we plan to use it in the future.
2. Online Random Forests seems to be a better option, but we could not find an open implementation, so its development and testing would require much more time. Online Random Forests add splits to their Decision Trees when new feedback arrives and tests their accuracy. when the score of a tree drops dramatically it is pruned from the forest and replaced with a newly trained tree.
3. Cloud services and mobile apps are a great alternative to complement SIE. We want to use a mobile application that connects to cloud services, where the training data of each crop is uploaded to make more robust our model and retrain each SIE system connected to internet. In addition this mobile app will show to the farmer important data about his crop, helping him to make decisions based on a numerical analysis.
4. A proposed solution for the varying biophysical conditions is incremental learning, since random forests are easily trained, a brute force search for the smallest hyper-parameters that satisfy a precision threshold for any given setup or crop would. This would provide models which satisfy the needs of the user with a not so complex model.
5. Another interesting option would be to explore CART(Classification and Regression Trees) where, for example labeling would be done in a scale from 0 to 1, and examples prior to irrigation would be labeled according to how close in time they are to the actual moment. Similar to a logistic Regression.

7. Bibliography

[1] FAO, The state of the worlds land and water resources for food and agriculture (SOLAW) Managing systems at risk, Food and Agriculture Organization of the United Nations, Rome and Earthscan, London, 2011.

[2] J. N. Lennart Woltering, Dov Pasternak, The african market garden: The development of a lowpressure drip irrigation system for smallholders in the sudano sahel (2011).

[3] S. C. B. Yuan Zhou, Knowledge Driven Development: Private Extension and Global Lessons, Academic Press, 2015.

[4] A. Dementyev, S. Hodges, S. Taylor, J. Smith, Power consumption analysis of bluetooth low energy, zigbee and ant sensor nodes in a cyclic sleep scenario, 2013 IEEE International Wireless Symposium, IWS 2013 (2013).

[5] D. I. Inc, XBee S2C DigiMesh 2.4 User Guide, 2018.

[6] H. L. T. Andy Stanford-Clark, MQTT For Sensor Networks (MQTT-SN). Protocol Specification, 2013.

[7] A. M. S. B. O. L. G. R. Anat Goldstein, L. Fink, Applying machine learning on sensor data for irrigation recommendations: revealing the agronomists tacit knowledge, Springer Science + Business Media New York 2017 (2017) 2–4.

[8] K. S. Laura Elena Raileanu, Theoretical comparison between the gini index and information gain criteria, Kluwer Academic Publishers (2004) 16.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[10] M. M. Sanyogita Andriyas, Recursive partitioning techniques for modeling irrigation behavior, Elsevier Environmental Modelling Software (2012) 1–4.

[11] A. Gupta, Building continuous learning systems, 2016. PyData DC 2016.