AN IOT AND MACHINE LEARNING APPROACH FOR SITE-SPECIFIC IRRIGATION IN RESIDENTIAL IRRIGATION SYSTEMS

A Thesis

by

AKSHAY RAMUHALLY SUDHEENDRA MURTHY

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Radu Stoleru
Committee Members,	I-Hong Hou
	Roozbeh Jafari
Head of Department,	Dilma Da Silva

August 2019

Major Subject: Computer Engineering

ABSTRACT

Irrigation schedules on traditional irrigation controllers tend to disperse too much water by design and cause runoff, which results in wastage of water and pollution of water sources. Previous attempts at tackling this problem used expensive sensors that aren't applicable to the residential landscape. In this thesis, we propose Weather-aware Runoff Prevention Irrigation Control (WaRPIC), a low-cost, practical solution that optimally applies water, while preventing runoff for each sprinkler zone. WaRPIC involves experiments conducted by homeowners on their landscape as part of a two-week data collection phase. The gathered data is used to build machine learning models that can accurately predict the Maximum Allowable Runtime (MAR) for each sprinkler zone given weather data obtained from a network of weather stations. We have also developed a low-cost module that can retrofit irrigation controllers in order to modify its irrigation schedule. We built a neural network-based model that predicts the MAR for any set of antecedent conditions, using data collected from a sprinkler zone. The model's prediction is compared with a state-of-the-art irrigation controller and the volume of water wasted by WaRPIC was only 2.6% of that of the state-of-the-art. We have deployed our modules at residences and estimate that the average homeowner can save 38,826 gallons of water over the course of May-Oct 2019, resulting in savings of $192.

DEDICATION

To my parents, who have been a constant source of support.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Radu Stoleru for his guidance and support throughout this process. I am particularly grateful for the assistance provided by Mr. Charles Swanson. The advice provided by Dr. Theodora Chaspari was immensely helpful. I would also like to thank the members of the Laboratory for Embedded & Networked Sensor Systems (LENSS) for their insights and suggestions. Special thanks to my family and friends, without whom none of this would be possible.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor Radu Stoleru and Professor Roozbeh Jafari of the Department of Computer Science and Engineering and Professor I-Hong Hou of the Department of Electrical and Computer Engineering.

The data analyzed for Chapter 4 was provided by Charles Swanson, from the Department of Biological and Agricultural Engineering. The cloud-based server-end software was developed in collaboration with Curtis Green, a student from the Department of Computer Science.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

This thesis was supported by a Water Seed Grant, courtesy of Texas A&M AgriLife Research.

NOMENCLATURE

MAR	Maximum Allowable Runtime
ET	Evapotranspiration
WMY	Water My Yard
PoP	Probability of Precipitation
LF	Light and Frequent Irrigation
ET_o	Potential Evapotranspiration
T_c	Turf Coefficient
A_f	Adjustment Factor
WR	Watering Requirement
PAW	Plant Available Water
SWHC	Soil Water Holding Capacity
MAD	Managed Allowable Depletion
IF	Irrigation Frequency
RT	Runtime recommendation
PR	Precipitation Rate
MQTT	Message Queue Telemetry Transport
HTTP	Hyper Text Transfer Protocol
AP	Access Point
DNS	Domain Name Service
FTP	File Transfer Protocol

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION AND LITERATURE REVIEW

1.1 Introduction

The average US household consumes over 300 gallons of water per day, with roughly 30% of that water being used outdoors [2]. Outdoor use of water occurs primarily in landscape management [3], where a major portion goes toward watering lawns. Lawns are estimated to cover an area of 128,000 km^2 in the United States [4]. In places like California, mandatory water restrictions were placed during the severe drought of 2016. Irrigation of lawns had to be curtailed as it represented a huge portion of the state's water usage [5]. With less than 1% of Earth's freshwater available for human use [6], and water demand increasing by the year due to a burgeoning population, it is imperative that we find ways to decrease water usage in all spheres of daily life.

On this note, the potential wastage of water occurring due to improper landscape irrigation practices is considered. Irrigation of turf has to follow a balanced approach. If the amount of water applied to turf is too less, turf starts to turn brown and dies. If turf is watered too much, runoff occurs. Runoff not only wastes water, but carries sediment, chemicals from fertilizers, and garbage [7] [8] [9] [10], causing *non-point source pollution* [11]. Homeowners tend to have irrigation schedules that stay the same over the course of the landscape irrigation season. This is inconsiderate of the potential for runoff from each sprinkler zone as the schedule assumes that factors such as soil depth, slope, etc. are uniform across the landscape. Another shortcoming of having fixed irrigation schedules is the possible wastage of water that could occur when turf-grass is irrigated even though the watering requirements have been met by rainfall caused by local rainfall. The watering requirements of turf-grass tend to fluctuate even during the landscape irrigation season [12]. Thus, having fixed irrigation schedules causes wastage of water and can possibly pollute water sources.

There have been attempts to conserve water used for landscape irrigation. Irrigation controllers that use moisture sensors were proposed in [13] [14]. Other strategies include using mechanical rain sensors, which are ineffective because of the fact that they are activated after rain has occurred, and do not have the preemptive ability to shut off the irrigation system before predicted rain. However, these solutions do not consider local weather conditions. Evapotranspiration-based (ET) controllers are available on the market, such as the ones by Rachio [15], Hunter [16], Rain Bird [17] that take into account weather conditions using a network of weather stations, but their prohibitively high cost makes them in-feasible for adoption by homeowners. Also, specifications for smart irrigation controllers [18] attempt to prevent runoff by defining a Maximum Allowable Runtime (MAR) for a sprinkler zone. However, they make simplifying assumptions about the nature of soil that renders such definitions ineffective. Irrigation water savings programs, such as the Water My Yard (WMY) [19] program from Texas A&M AgriLife Extension, that use weather data to determine the watering needs of turf. The program also notifies homeowners about when to turn off irrigation via text message/email. However, there is no way of ascertaining whether the homeowners followed through on the watering recommendations. Thus, previous efforts only solve a part of the problem, or are expensive solutions inapplicable to the residential landscape.

To address this issue, the Weather-aware Runoff Prevention Irrigation Control (WaRPIC) is proposed, which aims to maximize water savings in homeowners' irrigation schedules, taking into account local weather information and the potential for runoff on a sprinkler zone-by-zone basis. It is a low-cost, practical solution that is compatible with legacy irrigation controllers. It integrates with Water My Yard (WMY) for sprinkler runtime recommendations. It also prevents runoff by using machine learning-based models to predict the Maximum Allowable Runtime (MAR) for each sprinkler zone. To leverage the predictive ability of the machine learning models, the WMY module was constructed, which retrofits legacy irrigation controllers and adjusts the water dispersal by sprinklers dynamically depending on weather data. Thus, WaRPIC is an end-to-end low-cost solution that can easily integrate with existing irrigation infrastructure.

Data from a site on the campus grounds of Texas A&M University was collected, and a highly accurate neural network that prevents runoff by predicting the Maximum Allowable Runtime for any set of antecedent conditions was built. The model was found to have a very low error score

(Mean Squared Error) of 0.43. The predictions of the model was compared with a state-of-the-art smart irrigation controller and found that WaRPIC wasted only 2.6% as much water as that of the smart irrigation controller over the course of our evaluations.

WMY modules were deployed at 12 residences in the city of College Station. Even though the modules were used in the basic mode of operation, homeowners stand to save $192 dollars, and 38,826 gallons of water, on average, over the course of the current watering season. These savings, when compared to the low cost of manufacturing one WMY module ($18.81), represents a huge return on investment for the homeowners, and commendable water savings for water districts, public utilities and municipalities. The contributions in this thesis are as follows:

- Development of site-specific machine learning models that can accurately predict the maximum allowable run-time to prevent runoff for a given sprinkler zone, given weather data and a history of irrigation.
- Development of a low-cost actuator module that retrofits legacy irrigation controllers. An irrigation schedule has to be established on the irrigation controller.
- Automated control of legacy irrigation controllers based on the run-time recommendations generated by a network of weather stations, as well as site-specific data.

1.2 Literature Review

Much of the previous work in this field has focused on deploying wireless sensor networks, where each node has a soil moisture sensor attached to it, and relays sensor data back to a central controller. S. Fazackerley et al. [20] created an irrigation control system that deployed wireless sensors in a municipal green space. The sensors relayed data back to a central controller that had the ability to interface with a commercial irrigation controller. The controller sent the moisture data to a central web server, which presented the data in the form of a website. It also had the capability to by-pass the controller's irrigation schedule by shutting off irrigation. The system used a modified adaptive irrigation algorithm that used soil moisture sensors to ascertain the quality of watering.

The algorithm, however, did not take into consideration the slope of the field being watered, and the water drop rate is considered a constant. This doesn't account for local weather conditions such as solar radiation, wind speed, humidity.

The work done by N.Sales et al. [21] also involved creating a Wireless Sensor and Actuator Network, dubbed WSAN. The WSAN is a cloud-based solution that acts a framework that can be used to remotely control a irrigation system, based on information gathered from the sensors which are deployed as part of the WSAN. It also uses weather information from a 3rd party service provider. The key metric that is utilized from the weather data is Probability of Precipitation(PoP). The sensor information, along with PoP, are used as inputs to an irrigation algorithm that controls sprinkler runtimes. Even though this implementation considered local weather conditions, the efficiency of its irrigation algorithm largely depends on the accuracy of the moisture sensors, as well as the communication latencies of the wireless network infrastructure that is chosen. This solution also doesn't account for runoff occurring due to long irrigation runtimes.

D.Winkler et al. [22] came up with a solution that creates an optimal irrigation schedule based on a mathematical model generated by using a network of wireless sensor and actuator nodes. A partial differential equation(PDE)-based model was developed, and was optimized to help calculate accurate sprinkler run-times to that would consume the least amount of water. The work done on generating a site-specific mathematical model to help compute sprinkler run-times is commendable, but this work, like previous efforts, suffers from the inaccuracies that arise from trying to model the real-world through mathematical equations. The parameters in these equations are not uniform, and it is in-feasible to measure them at each site individually. Many assumptions were made to simplify the mathematical model so that it was solvable in a practical amount of time. The cost of each node in the wireless sensor network was also very high, making it in-feasible for adoption in an irrigation system with a larger number of sprinklers.

In the follow-up work to [22], D.Winkler et al. [23] chose a data-driven approach to precision irrigation. It used the same hardware setup as the previous work, but the PDE-model was eschewed in favor of an adaptive approach that involved models trained from sensor data. This enabled the system, PICS, to "learn and adapt" to the soil. Long-term and short-term models were developed to describe the movement of water through soil. We found some issues with the moisture profiles presented in [23]. The decay of Volumetric Water Content was shown to be much quicker than in real-world scenarios. Any model derived from such data is bound to irrigate lightly and frequently(LF irrigation) and this has been found to be inefficient method of irrigation. There is also the very high per-node cost of the hardware. The hardware setup requires each sprinkler to be fitted with an wireless sensor-actuator mote. So the actual cost of the system increases linearly with the number of sprinklers in the landscape.

2. BACKGROUND AND MOTIVATION

A typical lawn watering system consists of an irrigation controller, a network of underground pipes connecting the water supply to the sprinkler valves, and sprinkler solenoids. The irrigation controller governs water flow to sprinklers via a solenoid that controls water pressure at each sprinkler head. The irrigation controller can be scheduled to water the lawn automatically by setting an irrigation schedule. The irrigation controller works as a timer that uses the irrigation schedule to decide which sprinkler solenoid to activate and deactivate at what time. Irrigation controllers are runoff-agnostic because the homeowner is tasked with setting the runtimes for each sprinkler zone. But, the watering requirements for turf, as well as soil's potential for runoff as a consequence, change dynamically with the season. The irrigation schedules are inflexible in the sense that homeowners need to manually disable/enable them to control irrigation.

There have been solutions in research, as well on the market that promise smarter irrigation of turf. Evapotranspiration-based controllers, which integrate with large networks of weather stations have been on the market for sometime now. They are pricey solutions, however, and do not provide site-specific irrigation programs. Also, they cannot be retrofitted into existing irrigation controllers. The homeowner is tasked with the installation of a brand-new irrigation controller.

Runoff is typically caused by the application of water at a rate faster than unsaturated soil can absorb it. It is important to note that runoff doesn't occur instantaneously. Rather, there is a time interval of water application called Maximum Allowable Runtime (MAR) that determines how long we can irrigate the soil without runoff.

2.1 Irrigation water management

Water My Yard (WMY) [19] is a program of Texas A&M AgriLife Extension Service that aims to improve residential water savings by partnering with municipalities and public utilities. It deploys a network of weather stations that compute Potential Evapotranspiration (PET) on a daily basis. This helps in the calculation of sprinkler runtime recommendations. The recommendations

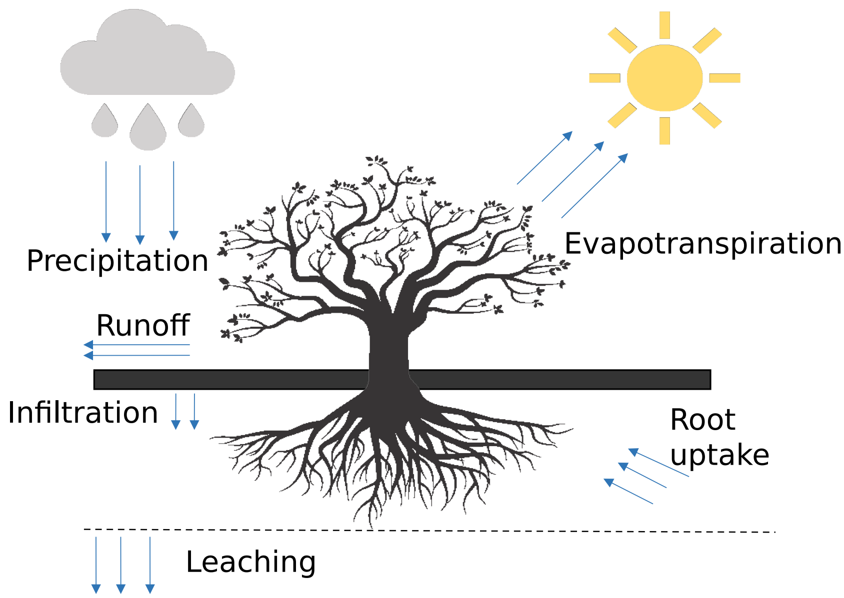


Figure 2.1: Interaction of atmospheric water with the earth's surface. Water My Yard computes the amount of water needed for plants to grow by considering processes such as precipitation and evapotranspiration.

are sent to homeowners on a weekly basis via email or text message. The methodology used to compute weekly sprinkler runtime recommendations is called the *water balance* method [24]. Figure 2.1 shows the interaction of various processes involved when atmospheric water interacts with the earth's surface. Evapotranspiration is one these processes.

2.1.1 Evapotranspiration

Water that is used by plants (or evapotranspirators, as they are called) is measured by a quantity called evapotranspiration. Potential Evapotranspiration (PET or ET_o) is a measure of potential water use by a cool season grass growing 4 inches tall under well-watered conditions. This is a reference measure that is used to estimate the water requirements of other plants (T_c). PET also depends on climate, surface conditions, free water, etc. An adjustment factor (A_f) is also applied to the water requirement to account for "allowable stress". Allowable stress reflects the degree of acceptable turf quality with reduced water supply. Thus, the watering requirement (WR)(in.) for a sprinkler zone, for a time period of 1 week, is defined by,

$$WR = ET_o \times T_c \times A_f \qquad \text{[in.]} \quad (2.1)$$

2.1.2 Effect of rainfall

Rainfall is a major factor that affects the watering requirement of turf-grass. It can be considered as an application of water, akin to a sprinkler running for a set amount of time. The watering requirement is adjusted to account for rainfall. The rainfall over the past week in a particular geographical area is considered as the measure of water added by rainfall (R). However, total rainfall is weighted by a factor called effective rainfall coefficient, giving us the effective rainfall (R_f). This is taken into consideration as a portion of rainwater becomes runoff. So, the watering requirement is now,

$$WR = ET_o \times T_c \times A_f - R_f \qquad \text{[in.]} \quad (2.2)$$

2.1.3 Plant available water

The ability of plants to absorb water from the soil is dictated by the depth of the root zone. The deeper the root zone, the greater the amount of water available for the plant to absorb. A shallow root zone can cause problems in the dry season, as water is lost quicker from the the top layers of soil, and plants need to be irrigated more frequently. The amount of water that can be held by the soil is defined as Soil Water Holding Capacity (SWHC). SWHC affects the root zone depth, and consequently sprinkler runtimes. SWHC is mainly influenced by soil type. It is found that fine-textured soils, such as clays, have high SWHC values, while course-textured soils, like sands, have lower SWHC. Once water has been applied to soil, various natural processes affect the quantity that stays in the root zone where it can be absorbed by plants. A factor called Managed Allowable Depletion (MAD) is also applied to account for the amount of water that can be absorbed from the soil before the plant shows stress. Thus, Plant Available Water (PAW) is given by,

$$PAW = SWHC \times D \times MAD \qquad \text{[in.]} \quad (2.3)$$

2.1.4 Deciding irrigation runtimes and frequency

The irrigation frequency (IF) need to be determined taking into account the watering requirement as well as the utility of the applied water. This is determined by the following calculation,

$$IF = \frac{WR}{PAW} \quad (2.4)$$

Finally, the water to be applied needs to be interpreted in terms of sprinkler runtime. The metric that needs to be determined is Precipitation Rate (PR) of the sprinklers. This can be obtained from manufacturer readings, as well as other means such as the catch-can or meter-reading methods. The weekly sprinkler runtime recommendation (RT) is then calculated using,

$$RT = \frac{WR \times 60}{IF \times PR} \qquad \text{[min.]} \quad (2.5)$$

2.2 Maximum Allowable Runtime (MAR)

We learned in the previous section about the considerations that are taken into account when computing the total sprinkler runtime on a weekly basis. The challenge, however, is that of ensuring irrigation takes place without causing runoff. There has been plenty of work on studying soil behavior in response to the application of water [1] [25], and in determining the amount of water that can be applied without causing runoff [22], but these models cannot be applied to residential landscape because of the inherent variation of factors such as soil depth, slope, exposure to sunlight, etc. Many of the solutions for site-specific irrigation derive complex models to explain moisture movement through soil. However, it is computationally in-feasible and unreliable to devise a mathematical model to represent moisture movement through the landscape in a residential setting. Also, it is expensive to deploy and maintain sensors throughout the landscape that continuously monitor the state of the soil at all times. We will now try to understand the process of runoff and the factors that cause it.

The conditions that cause this phenomenon fall under two categories. First, if water is applied at a rate greater than the soil can absorb the applied water, runoff occurs. This is called infiltration excess overland flow [26]. Second, if the entire soil profile has reached saturation, applied water will runoff over the soil surface. This is known as saturation excess overland flow [27].

A property of soil that is critical to the process of runoff is Infiltration capacity. Infiltration capacity (f) was defined in [1] as the maximum rate at which water can enter soil. Infiltration capacity of soil under constant application of water varies with time as shown below:

$$f = f_c + (f_o - f_c) \times \epsilon^{-k \cdot t} \qquad \text{[in./hr.]} \quad (2.6)$$

Where f_c is the final infiltration capacity, f_o is the initial infiltration capacity and k is the exponential decay constant.

The above equation implies that water applied at a constant rate to soil may not be absorbed completely over a given time interval. The excess water is lost as *surface runoff.* The Time to Ponding (T_p) can be derived from the above equation, for a given precipitation rate (ρ), as:

$$T_p = \frac{1}{k} \times \ln \frac{f_o - f_c}{\rho - f_c} \qquad \text{[min.]} \quad (2.7)$$

In Figure 2.2, we've depicted the variation in infiltration capacity of soil due to the steady application of water at the rate of 0.5 in/hr. The initial infiltration capacity starts at 1.5 in/hr and decays to a saturation value of 0.2 in/hr, with value of the decay constant equal to 0.35. The T_p is also shown in the figure. The cumulative infiltration, as depicted in the figure, is the amount of water that enters the soil at $t = T_p$. Precipitation applied after this point in time will start to accumulate on the soil surface and start to runoff.

However, this relation does not account for the recovery of infiltration capacity during dry periods. This was addressed in [28] where the Horton equation was adapted to complex storms. This was made possible by the incorporation of the concept of soil storage and drainage, introduced

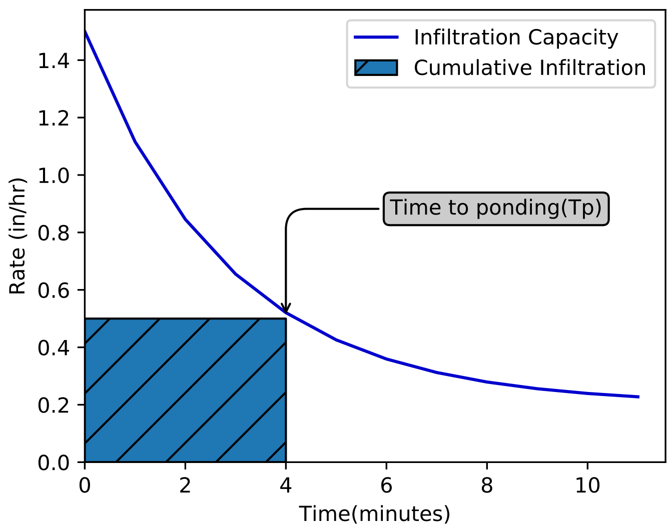


Figure 2.2: Decay of infiltration capacity with time according to Horton's equation [1]. At $t = Tp$, applied water starts to accumulate on the surface of the soil and begin to runoff

in [29]. The equation below shows the drainage rate (d) of water to lower layers,

$$d = f_c \times (1 - \epsilon^{-k \cdot t}) \qquad \text{[in./hr.]} \qquad (2.8)$$

The concepts and relationships detailed above help form the basic idea behind the *cycle-and-soak* method of irrigation [30]. In the cycle-and-soak method, the total sprinkler runtime is broken up into multiple cycles (time allowed for water to disperse) with periods of no irrigation (soak/absorb) so that applied water can be absorbed completely, and for the infiltration capacity of the soil to recover. This was also explored in [18], where an upper bound for sprinkler runtime called *Maximum Allowable Runtime* (MAR) was proposed. This equation also takes into account *Allowable Surface Accumulation* (ASA) and makes a simplifying assumption that the Infiltration Capacity (IR) is a constant value.

$$MAR = \frac{60 \times ASA}{PR - IR} \qquad \text{[min.]} \qquad (2.9)$$

The above equation could serve as a benchmark for deciding the cycle time in a landscape irrigation setting. However, the assumption that the initial infiltration capacity (f_o) stays constant is a problematic one. There is a large body of research which has observed that infiltration capacity varies on a spatiotemporal basis, affecting runoff rates. In [31], it was shown that seasonal changes in infiltration capacity caused by changes in the soil influences the rate and process of erosion on hill-slopes. The results from [32] and [33] also showed that spatially variable and temporally dynamic soil properties affect the erosional response of soil on hill-slopes and in Mediterranean badlands. Different soils have different responses to the application of water, according to [34], and this affects the potential of the applied water to be lost as runoff. Even factors such as plant species diversity and time of day affect the infiltration capacity, as shown by [35] and [36].

A naive approach of water application that might prevent runoff would be to schedule numerous irrigation cycles where the quantity of applied water is not enough to cause runoff. The problem with this approach, however, is that it affects the depth of plant roots. Plant roots tend to grow in relation to a soil moisture gradient. This property is known as hydrotropism [37] and this enables plants to grow deeper into the soil. If a higher quantity of water is applied to soil in each irrigation cycle, the applied water seeps deeper into the soil. This is essential during periods of high evaporation such as in the summer months. This led to better turf quality and in other indicators of plant health such as root thickness, mat depth, etc.

2.2.1 Predictive modeling of MAR

We have established that infiltration capacity is a dynamic quantity that varies on a spatiotemporal basis. We found that the primary influencing factor that affects the MAR is antecedent soil moisture. Soil moisture content is highly correlated with the flow of water into soil. Thus, infiltration capacity is also affected by the soil moisture content [38]. The antecedent soil moisture is influenced by past weather conditions (in the local geographic area) as well as previous applications of water. We consider a time frame of 1 week for this purpose. The following factors influence antecedent soil moisture, and in turn infiltration capacity at the beginning of an irrigation cycle:

- Evapotranspiration(Loss of water from the soil) (ETO).
- Antecedent Water Application (AWA) (Sum of water applied via rainfall and previous irrigation cycles).
- Last Water Application (LWA) Time since previous irrigation.

We believe that conducting a set of experiments on-site will help us gather the data needed to solve a supervised learning problem. Thus, we converted the complex problem of estimating the MAR for a given site, the parameters for which change on a spatiotemporal basis, into a predictive modeling problem. We need to find a function $\boldsymbol{g}$ that estimates the MAR.

$$MAR = \boldsymbol{g}(ETO, AWA, LWA) \tag{2.10}$$

3. SYSTEM OVERVIEW AND PROTOTYPE DEVELOPMENT

We propose a methodology that aims to prevent runoff due to irrigation of the landscape by implementing machine learning models trained on site-specific data, eliminating the need for unreliable models and expensive sensors and equipment. We delegate the responsibility of mapping antecedent conditions and the resultant MAR to machine learning models such as Artificial Neural Networks(ANN) that excel at learning non-linear relationships between parameters in a given dataset. The models will be used in the predictive modeling of MAR on a zone-by-zone basis, as well as in the creation of irrigation program on the legacy controller that is customized to the characteristics of each sprinkler zone present in the landscape.

To help leverage the predictions of the machine learning models, we built the WMY module. It is a wireless actuator node, built to be low-cost, and have a small footprint. It can retrofit legacy irrigation controllers via the rain sensor port or across the common wire that is connected to all the sprinkler valves. This enables the module to shut off irrigation, akin to a rain sensor. However, unlike traditional rain sensors, the module's actuations can be controlled based on accurate runtime recommendations generated by the network of stations that are part of the WMY program.

To complement the capabilities of the WMY actuator node and the predictive ability of the neural network, we implemented the Runoff-aware Decision Engine (RaDE), which is a cloud-based control system that controls irrigation on a zone-by-zone basis. An irrigation program is created on the irrigation controller that maximizes the ability of RaDE to enable and disable irrigation during runtime of the sprinklers. The irrigation program also ensures that enough time is given for the application of water to soak into soil before applying water again. Control is then handed over to RaDE, which ensures irrigation occurs in timely manner and with an aim to prevent runoff from irrigation.

3.1 Weather-aware Runoff Prevention Irrigation Control (WaRPIC)

We illustrated in the previous section, the predictive modeling problem for MAR that takes into account past weather conditions and applications of water. In this section, we present the Weather-aware Sprinkler Control (WaRPIC), a low-cost solution for landscape irrigation where control of legacy irrigation controllers is driven by machine learning models built on site-specific data. The models are trained on data gathered from experiments conducted on-site. Figure 3.1 shows an overview of WaRPIC. WaRPIC does not involve a deployment of sensors in the ground as we found that accurate moisture sensors tend to be very expensive, and hard to maintain. It involves two phases:

- **Zone-wise data collection and training**: In order to construct machine learning models, WaRPIC requires data for each sprinkler zone. During this phase, the homeowner will observe the application of water to soil in each zone, and report the MAR at the end of each water application. At the end of this stage, we develop MAR-predict, a set of machine learning models that predict MAR on a zone-by-zone basis. Then, an irrigation schedule for the landscape is created that maximizes the capabilities of WaRPIC.

- **Deployment**: The Runoff-aware Decision Engine (RaDE) handles the actuation of sprinklers during the irrigation schedule. The decisions of RaDE are influenced by past weather data, as well as data about previous irrigation cycles. We also aim to keep the homeowner in the loop by having them report back to us about the performance of RaDE to further fine-tune its performance.

Most commercial irrigation controllers available on the market today are runoff-agnostic, in the sense that the run-times chosen for sprinklers in each zone is either arbitrarily chosen by the homeowner or is based on a generic recommendation that isn't zone-specific or considerate of past weather conditions. The solutions presented by Rachio [16] and Hunter [15] do have the feature of shutting off watering when rainfall is predicted in the future, but they are expensive solutions, and do not provide the degree of customization needed for site-specific irrigation.

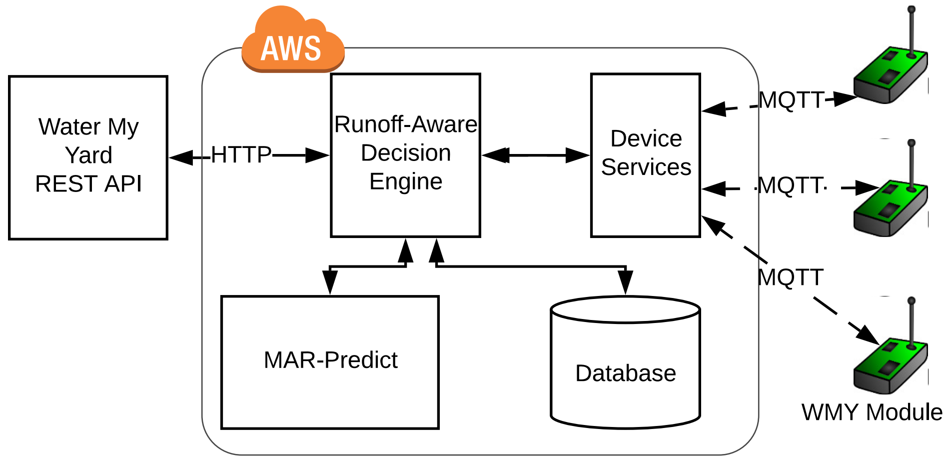


Figure 3.1: Overview of WaRPIC

Table 3.1: Summarized features of state-of-the-art vs. WaRPIC

Feature	Hunter	Rachio	WaRPIC
Low-cost	-	-	+
Runoff-aware schedule	-	-	+
Weather forecasts	+	+	-

A summary of features of the Rachio Generation 2 and Hunter HC-6 and in comparison with WaRPIC is shown in Table 3.1.

3.1.1 MAR-Predict

We propose MAR-predict, which uses the predictive ability of machine learning models to influence irrigation scheduling decisions. The key advantage of MAR-predict is that it is trained on site-specific data, lending a level of personalization to the irrigation schedule that cannot be provided by a generic model. We present a high-level overview of the procedure by which MAR-predict is generated for a typical residence:

- Conduct experiments on-site and site survey to collect data.
- Train site-specific machine learning models that can accurately predict MAR.

3.1.2 Site survey and zone-wise data collection

The occurrence of runoff is dependent on certain factors, which are intrinsic to the soil and don't change with time, such as soil texture, composition, slope, etc. It is also dependent on the amount of water that has been applied to the soil previously. This is known as antecedent soil moisture. Authors of previous works [23] [22] that address the problem of runoff advocate the use of moisture sensors that will be deployed in each sprinkler zone, and base scheduling decisions on sensor inputs. However, this approach is very expensive in terms of equipment and time required to set up. We propose an approach that involves conducting experiments on-site, on a zone-by-zone basis, and understand soil behavior to the extent that we can accurately predict the MAR for a given set of antecedent conditions.

The experiments involve the measurement of MAR at various stages of topsoil saturation, thus giving us a complete picture in terms of the potential of soil for runoff at different moisture levels. The data gathered from these experiments is used to build site-specific machine learning-based models that can accurately predict the MAR for any set of antecedent conditions. It is advisable to conduct these experiments during the summer/dry weather conditions as rain tends to drive the soil to saturation at a faster rate (a volumetric water content where soil cannot absorb any more water). Each experiment has two phases which are as follows.

Determination of MAR: The sprinklers in each zone are allowed to run until an appreciable quantity of ponding is visible on the soil surface. The sprinklers are allowed to run up to the point where adding any more water to the soil will result in runoff. This means that the infiltration capacity of the soil is lesser than precipitation rate, and the water on the soil's surface has reached ASA. This is the MAR for the given set of conditions.

Infiltration capacity probing: The soil's infiltration capacity is allowed to recover. To determine the minimum amount of time to wait before starting another irrigation cycle, we probe the infiltration capacity of the soil. This is done by allowing the sprinklers to run for a very short time period (10 seconds) and having the homeowner observe the state of the applied water. If the applied water remains on the surface of the soil, then infiltration capacity hasn't recovered to the

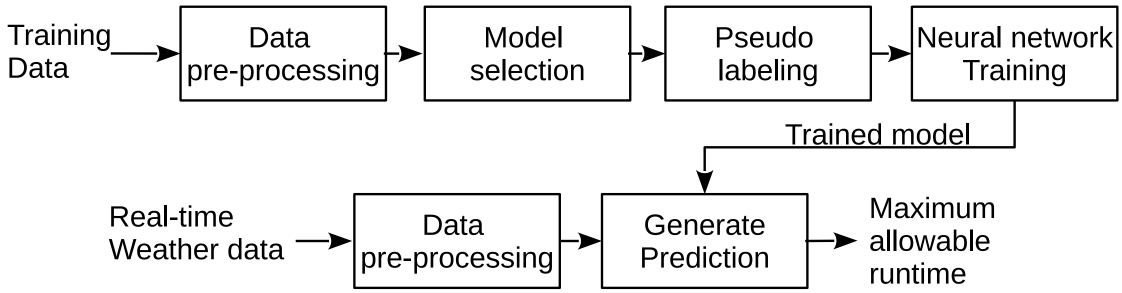


Figure 3.2: The machine learning pipeline that is used in our methodology.

point where the soil can absorb more water and we have to wait for more time. If the water has been absorbed, then this means we have waited long enough for the soil to recover. We conduct this "infiltration capacity probe" periodically (5 minutes), and thus determine the absorption time for an irrigation cycle.

An evaluation was conducted at one of the grounds at Texas A&M University's campus, using the methodology described above. We found the best approach to be to conduct experiments on a daily basis for a week. Each day, the experiment is to run as many cycles as possible. Once the soil is saturated, we allow the soil to drain away the water for a few days. Next week, we conducted the same experiments, but the time difference between experiments is increased. This is continued until 15-20 experiments are conducted.

The approach also demands that the homeowner conduct a site survey to ensure the sprinklers on-site are working and are dispersing water evenly. A catch-can or flow-meter test should be conducted to measure the precipitation rate of the sprinklers as this is important for server-side computations.

3.1.3 Machine learning pipeline

Our goal of eschewing the traditional parameter-heavy approach to MAR prediction and modelling the problem through a predictive modeling approach allows us to leverage the predictive ability of machine learning algorithms to perform highly accurate predictions of MAR. Now we describe the WaRPIC machine learning pipeline. An illustration of the methodology followed is shown in Figure 3.2.

3.1.3.1 Data pre-processing

Firstly, the dataset is standardized by converting it to a distribution with zero mean and unit variance. Then, it is divided into training and testing sets, using a 70/30 split. This is a re-sampling procedure that can help evaluate the final model that we will use. Since this is a predictive modeling problem, the best model/algorithm is the one that performs best on unseen data.

3.1.3.2 Model selection

WaRPIC uses 5-fold cross validation on the training data to compare the performance of various models on the training data, with mean squared error as the scoring metric. Cross validation is also a re-sampling procedure that subjects the model to different splits of the training data. It provides a population of performance measures, thus giving us an idea about the average predictive ability of the prediction algorithm for the given data distribution. *Ridge Regression* had the best score in cross validation for the experimental data collected from the site at Texas A&M University's campus. We used Scikit-learn's machine learning library [39] for the data pre-processing, cross validation and standard machine learning models.

3.1.3.3 Pseudo-labeling

Even though Ridge Regression had the best performance in cross validation, it is hard to justify its predictive ability as the number of samples is very low. *To overcome this shortage of data we use a Semi-Supervised Learning (SSL) technique called Pseudo-labeling [40].* Pseudo Labeling is a method used to increase the number of training data samples available to a learner. The steps involved in the pseudo-labeling procedure are as follows:

1. Train a model using the labeled (smaller) dataset.
2. Use the trained model to make predictions on the unlabeled data
3. Retrain another model on the augmented dataset in order to get better predictions.

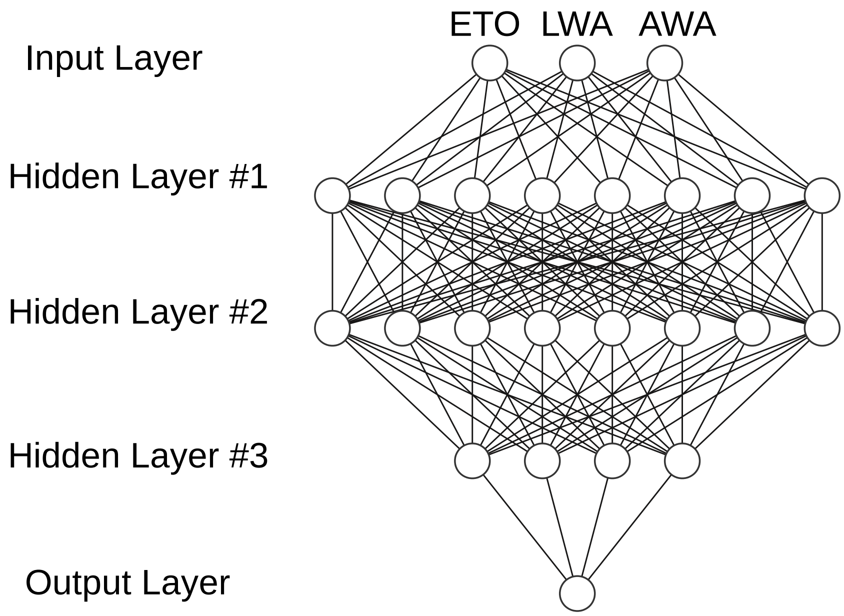


Figure 3.3: The neural network architecture that performs MAR prediction

WaRPIC also uses unlabeled data to perform pseudo-labeling. For this, weekly evapotranspiration and rainfall data from the nearest weather station to the homeowner's residence is used. This is historical data containing evapotranspiration and rainfall volumes. Pseudo-labeling augments the number of data samples using the best performing model from cross validation. To evaluate the performance of the regressor trained on the combined labeled and unlabeled dataset, a hold-out/test set is used.

3.1.3.4 Neural network training

The size of the augmented dataset makes it a suitable application of artificial neural networks (ANN). ANNs have proven to be excellent at mapping non-linear relationships between input features over the past few years, in diverse fields such as computer vision, banking and retail, medicine, etc. ANNs perform very well at the task of recognizing patterns in data thanks to hidden layers of computational units called 'neurons' whose behavior is inspired from their biological counterparts. When combined with the ability of optimization algorithms such as gradient descent [41] and the Adam optimizer [42] to quickly converge to a solution, means that neural net-

Algorithm 1: Algorithm for irrigation schedule creation

Input: Z, A, H, N, $S_{initial}$

Output: {S}, {W}

for $z \in [1, 2, \ldots, Z]$ **do**

  $W_z \leftarrow$ `windowOp(`M_z, H`)`

$S_0 \leftarrow S_{initial}$

for $i \in [1, 2, \ldots, N-1]$ **do**

  $S_i \leftarrow S_{i-1} + \sum_{z \in Z} W_z + A$

return S,W

Function `windowOp(`*model, history*`)`**:**

  $m \leftarrow \emptyset$

  for $sample \in history$ **do**

    $m \leftarrow m \cup model(sample)$

  return max(m)

works can be trained quickly and achieve high accuracy compared to traditional machine learning algorithms. The shortage of data points preventeds us from using ANNs for this application, but the pseudo-labeling technique solves this problem by increasing the size of the dataset. WaRPIC uses a neural network architecture on the augmented dataset that outperformed all other machine learning models such as Support Vector Regression, Ridge Regression, etc. The results are discussed in greater detail in Chapter 4. The neural network architecture is shown in Figure 3.3. We used the Python deep learning library Keras [43] to construct the neural network and train it.

3.1.4 Irrigation schedule creation

Once WaRPIC creates accurate machine learning models for each sprinkler zone, the homeowner needs to assist in leveraging the predictions of the models to affect landscape irrigation and prevent runoff. As we will explain in Section 3.2, the relay on the WMY module allows us to control the activation of sprinkler valves. However, the WMY module cannot, by itself, trigger the sprinkler solenoids on its own. The irrigation controller must be in the midst of an irrigation schedule. This serves as a **window-of-opportunity** during which the WMY module can enable/disable irrigation by controlling the state of the relay. Thus, there is a need for an irrigation schedule that maximizes the ability of the WMY module to control irrigation, while also allowing for sufficient absorption of the water applied to each zone. The key considerations for the irrigation schedule creation are presented below:

As we established in 2.2, the *cycle-and-soak* methodology is the preferred irrigation program for the prevention of runoff. It ensures that each application of water to the soil doesn't cause runoff and allows for the recovery of infiltration capacity between cycles. The time allowed for the applied water to soak into the soil and reach the root zone is known as *absorption time*. If water is applied too soon, we run the risk of adding water to saturated topsoil and cause runoff. Most modern irrigation controllers enable cycle-and-soak by adding the feature of multiple start times to the irrigation schedule. This means that at each start time, the valves selected for the particular schedule will run in sequential order. While designing a cycle-and-soak irrigation schedule, the start times must be spaced out enough that each zone is soaked well enough before water is applied to it.

We present an algorithm, shown in Algorithm 1 on the preceding page, for the creation of an irrigation program that uses the results from the simulation of machine learning models on historical data as well as experimental data gathered on-site. The inputs to the algorithms are a set of zone-specific machine learning models (Z), the absorption time between irrigation cycles (A), historical data from the nearest weather station (H), number of start times that can be set on the irrigation controller (N) and an appropriately chosen initial start time ($S_{initial}$). The initial start time is obtained from public utility recommendations (typically early morning before sunrise). The algorithm creates the irrigation program by constructing a set of start times (S) and a set of windows-of-opportunity (W). The *window-of-opportunity* is a specific window during which sprinkler valves can be controlled by the WMY module. The window must be chosen in such a way that irrigation can be enabled for the entire duration of MAR, but also not too long that, in case of a failure on the part of the WMY module, irrigation does not continue to the point where it affects the plants' health. To achieve this, WaRPIC uses historical weather data from WMY for the nearest weather station to the homeowner's residence. The machine learning model is developed through

Algorithm 2: Real-time control of WMY modules (advanced mode)

Function `sprinklerControl(`$RR, Z, S, R, cycle$`)`**:**
    $ETO, RFA, RFT \leftarrow$ `queryWMY()`
    $ST \leftarrow S_{cycle}$
    for $z \in [1, 2, \ldots, Z]$ **do**
        $SWA, SWT \leftarrow$ `queryLocal(`z`)`
        $LWA \leftarrow \max(RFT, SWT)$
        $AWA \leftarrow SWA + RFA$
        $\delta \leftarrow \min(RR_z,$ `getMAR(`M_z, ETO, LWA, AWA`)` $)$
        `EnableIrrigation(`$z, ST, ST + \delta$`)`
        $RR_z \leftarrow RR_z - \delta$
        $ST \leftarrow ST + R_z$
    `sprinklerControl(`$RR, Z, S, R, cycle + 1$`)`

the machine learning pipeline, which tunes its parameters so that it has the least generalization error. WaRPIC performs a simulation by using the machine learning models to predict the MAR for each instance of the historical data. The results of the simulation helps choose the window of opportunity. This is shown by the function windowOp() in Algorithm 1 on page 21. The maximum value in the set of MARs predicted by the model is chosen as the runtime for the sprinkler zone in the irrigation schedule created by the homeowner.

3.1.5 Runoff-aware Decision Engine (RaDE)

The Runoff-Aware Decision Engine (RaDE) is a sub-module of WaRPIC that coordinates the actuations of sprinklers installed in irrigation systems retrofitted with WMY modules. It manages sprinkler actuations to provide water to the turf in accordance with local weather conditions while ensuring that runoff doesn't occur. It communicates with WaterMyYard's REST API that provides accurate weather information as weekly sprinkler runtime recommendations. WaRPIC stores site-specific machine learning-based models that estimate the MAR on a zone-by-zone basis. It also communicates with the deployed WMY modules via the MQTT protocol. Once the irrigation program has been created on the homeowner's irrigation controller, the module can then work with the watering recommendations of WMY on a week-by-week basis to optimally apply water with

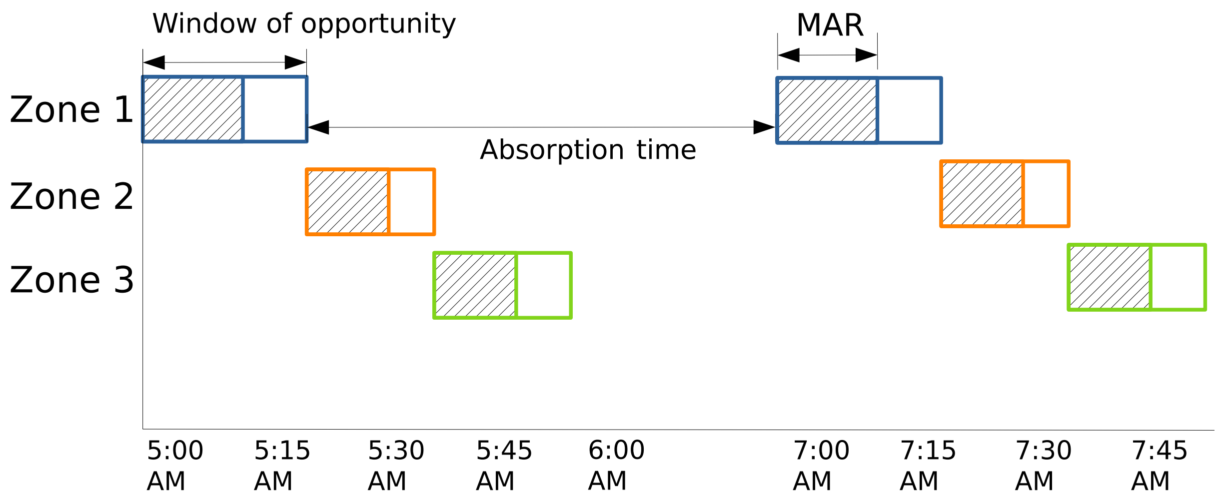


Figure 3.4: A typical irrigation schedule set on a irrigation controller by a homeowner. Each zone has a start time and window-of-opportunity determined by the irrigation schedule. WaRPIC adjusts the water dispersion by enabling and disabling the valve during the window-of-opportunity

the goal of preventing runoff. RaDE has two modes of operation which give it varying degrees of control over the application of water to turf:

3.1.5.1 Basic mode

This mode allows a greater degree of freedom to the homeowner in terms of choosing the irrigation schedule. The WMY module serves in a manner similar to that of a rain sensor whose actuations are controlled by RaDE. In this mode, the WMY module simply shuts off irrigation when the rainfall over the past 7 days has exceeded the watering requirements of turf. The signal to enable/disable irrigation are sent to the module on a daily basis by RaDE, upon checking the watering recommendations from WMY for the nearest weather station to the location of the homeowner's residence.

3.1.5.2 Advanced mode

In this mode, we see the effect of the predictive ability of the site-specific prediction models that have been trained on on-site data. The database on the cloud-based server stores information regarding the start times as well as the window of opportunity for each zone during which the

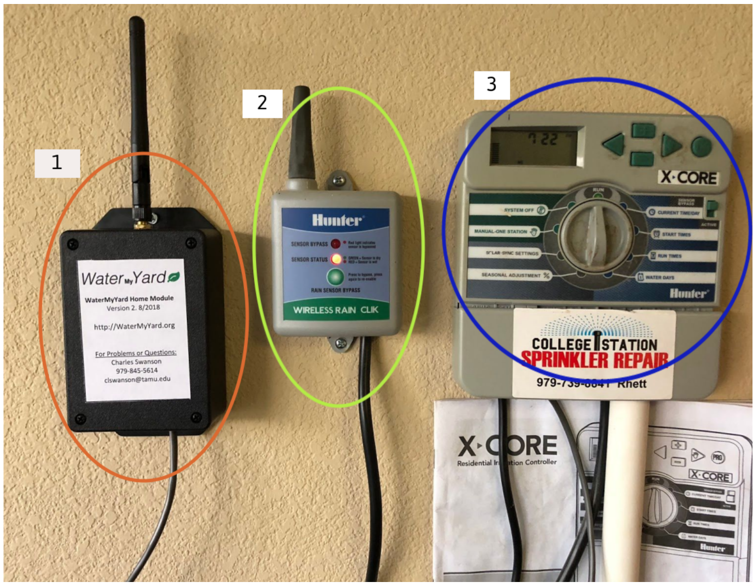


Figure 3.5: The WMY module post-installation (1) at homeowner's residence. The module can function as a solitary add-on to the irrigation controller (2) or in tandem with a rain sensor (3)

module can enable/disable irrigation. Using this information, as well as the highly accurate predictions of zone-specific models that take into account antecedent conditions, RaDE can perform precision control of irrigation at the homeowner's residence. The real-time control of sprinklers is shown in Algorithm 2. RaDE obtains the recommended weekly runtime recommendation (RR) at beginning of the week. RaDE stores the set of machine learning models (Z), start times (S), and set of windows-of-opportunity (W). The response from API endpoint also contains a 7-day weather summary. This helps us derive the evapotranspiration (ETO), rainfall (RFA) and time since rainfall (RFT). Data about past irrigation(Sprinkler Water Application (SWA), time since sprinkler watering (SWT)) is stored in the local database. The function then proceeds to obtain the MAR using the queried data. Finally, irrigation is enabled for the time period of MAR. The function then proceeds in a recursive manner to control the next cycle and so on.

3.2 Prototype Development

The WMY module, depicted in Figure 3.5, is a low-cost and practical solution that can automate functioning of legacy irrigation controllers. It has the following features:

- It can retrofit legacy irrigation controllers via the rain-sensor port or through the common wire connecting that acts as the ground for all the sprinkler valves.
- It can be setup by the homeowner through a captive portal and upon entering WiFi credentials, the module connects to the home's WiFi access point.
- It communicates with the cloud-based central server via the MQTT protocol, which enables functionality such as log-file uploads, firmware updates, and remote actuation of GPIO pins.

3.2.1 Hardware design

3.2.1.1 Power

The requirement of the WMY module to retrofit into existing sprinkler controllers requires it to use the same power source as the irrigation controller, which is 24-28V AC. A rectifier and a buck converter is used to step down the high AC voltage to a safe DC voltage for the voltage-sensitive micro-controller and sensing modules. The buck converter regulates the DC voltage as well, as 5V DC is the supply voltage for the rest of circuit.

3.2.1.2 Computation

The computation unit consists of an ESP8266 micro-controller which is responsible for coordinating communication, sensing and actuation, as well as the initial setup of the system at the homeowner's residence. The ESP8266 is a low-cost, low-power microchip that has full WiFi capabilities with a full TCP/IP stack [44]. It interfaces with the relay to change its state when sent a command to do so. The micro-controller also keeps a log of all the commands that were sent to it, along with information about outages in communication, like when it loses a connection with the WaRPIC server. It measures the signal strength of the wireless signal of the Access Point(AP) during an outage, which helps in pinpointing the source of the outage.

3.2.1.3 Actuation

The actuation component brings about control of the irrigation controller's schedule. Most irrigation controllers have a rain sensor port, which is meant to disable the controller when the rain

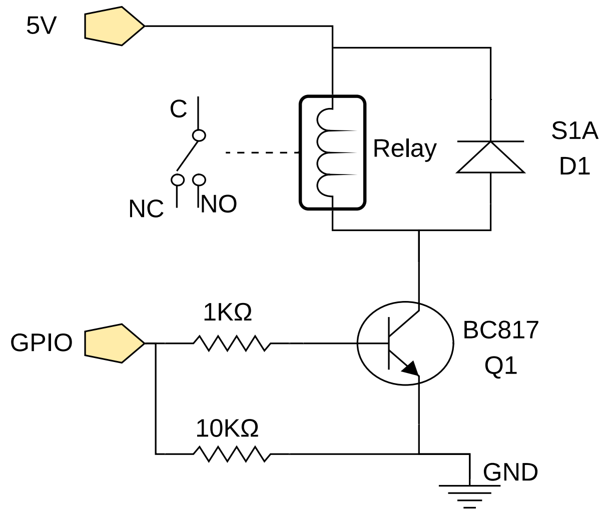


Figure 3.6: The actuator circuit that retrofits the irrigation controller.

sensor has been activated by rain. We delved into the methodology with which these rain sensors are activated and realized that they a rain sensor is a mechanically activated switch. The switch is activated by the force of water weighing down upon a contact. This activation causes the rain sensor circuit to become an open circuit. The irrigation controller detects the open circuit, and when it does, halts irrigation. Any irrigation schedules that are supposed to run don't proceed unless the rain sensor circuit becomes a closed circuit again. We realized that this behavior can be mimicked by a electro-mechanical relay. An electro-mechanical relay is nothing but a electrically activated switch, whose activation is caused by the application of a voltage across an electro-magnetic coil. A relay, when activated, creates a closed circuit with null resistance. When deactivated, the circuit is opened, causing infinite resistance. Upon detection by the irrigation controller, the irrigation schedule is immediately halted. The WMY module uses this principle to take control of the irrigation schedule. The circuit design for the actuator is shown in Figure 3.6.

3.2.2 Software design

3.2.2.1 Firmware

The software running on the device is responsible for coordinating communication with the central server, as well as performing other activities such as actuation, timekeeping, logging and updating firmware.

Config mode: The WMY module needs a working Internet connection to be able to communicate with the central server. So, upon booting up for the first time, the module creates an Wireless Access Point (AP) so that the homeowner can connect to it. The AP has a captive portal which functions as a Doman Name Service (DNS) and Hyper Text Transfer Protocol (HTTP) server. The user is directed to enter their WiFi credentials on a form in the portal's web server. The module then proceeds to shut down the AP, and goes to STA mode. If the module cannot connect to the home's WiFi due to mis-typed credentials, it will go back to AP mode, and the homeowner will have to enter the credentials again.

STA mode: Once the module connects to the home's WiFi, it attempts to connect with the central server. We choose the Message Queuing Telemetry Transport (MQTT) protocol as the basis of communication between the central server and WMY modules. MQTT is a messaging protocol well-suited for low-bandwidth, resource-constrained devices. It registers itself with the message broker, and the broker assigns a client ID to it. The broker uses this client ID to communicate with the device. If the connection is lost, the device attempts to reconnect to the broker, and is assigned a new client ID. MQTT is based on the publish-subscribe paradigm, where the broker (WaRPIC server) and the clients (WMY modules) communicate via communication channels called 'topics'. Each WMY module has two topics that it is associated with. The module is assigned a serial number while flashing its firmware, and this serial number is stored in Read-only Memory (ROM). This is done to ensure that the module's serial number stays the same across multiple reboots. The serial number helps the module identify the topics it has to subscribe and publish to. The commands sent by the broker to the module are executed as asynchronous event callbacks. These include control

Figure 3.7: The progressions of the WMY module through versions 1, 2 and 3. Version 1 was built using off-the-shelf components. We designed a PCB for Version 2. The design was further refined for Version 3 of the WMY module

of General Purpose Input/Output (GPIO) pins, firmware updates, file handling/uploads,etc. The firmware updates are via HTTP, and the log files are uploaded to a remote server using File Transfer Protocol (FTP). The PubSubClient library for the ESP8266 was used to implement the MQTT protocol on the device-end.

3.2.3 Cloud-based central server software design

The server-end software is responsible for sending and receiving messages from the modules deployed in homes across a large geographical region. It also needs a web-based portal that can be used to send commands to the modules manually. IoT services such as AWS IoT [45] are mostly aimed at more powerful IoT devices which run some version of embedded Linux/FreeRTOS [46]. Hence, we decided to implement the MQTT broker and web portal using Node.js and host on an AWS EC2 instance.

The MQTT broker is hosted using a library called Mosca that acts as a wrapper around the MQTT.js library. Mosca can use databases to log messages, which helps it scale easily. We used

MongoDB for storing MQTT packets,session information,etc. The broker maintains connections between the devices and itself, and allows them to publish, and subscribe to designated topics for communication. Another key aspect of the server-side software was a dashboard-like web portal that helps administrators manually communicate with the deployed WMY modules on their respective channels and monitor their behavior. The web portal connects to the MQTT broker and the message database. The web server was setup using Express, the standard web application framework for Node.js.

The web server also hosts the binary images of the WMY firmware so that the WMY modules can update their firmware Over-The-Air (OTA) via HTTP. The automated control of the modules on a daily basis is brought about by a cron job that runs on the EC2 instance which uses the MQTT broker to send messages to the modules based on the nearest WaterMyYard weather station. When each module comes online for the first time, it is associated with a weather station. This association is used by the script to ping the WaterMyYard API, and gather the latest watering recommendation. It makes a decision based on the runtime recommendation to turn on/off the irrigation controller, and communicates the same to the module.

4. PERFORMANCE EVALUATION

4.1 Site description

The site where we conducted experiments and gathered data to perform site-specific irrigation was situated on the grounds of Texas A&M University. The site contained a row of spray sprinklers which could be operated from an irrigation control station. There is appreciable coverage of grass over the site's area. The variety of turf was determined to be *Bermudagrass*. Upon examination of the site, we determined that only a small area of the land that could be irrigated by the sprinkler was useful for experiments, as many of the sprinkler heads were broken. We performed area measurements, and catch can tests to measure precipitation rate of the sprinklers. The terrain was mostly uniform and had a very slight slope associated with it. The site is exposed to a lot of sunlight owing to its distance from buildings, trees, etc.

4.2 Machine learning pipeline

4.2.1 Model selection through cross validation

As explained in 3.1.3.2, model selection is a comparison procedure performed through cross-validation that evaluates the performance of different machine learning algorithms on the training data. The results of 5-fold cross validation of different models are presented in Figure 4.1. The scoring metric chosen for model selection was Mean Squared Error(MSE). The model that performed best in the model selection procedure was Ridge Regression with a mean squared error of 1.47 and a standard deviation of 1.04.

Table 4.1: Comparing different machine learning models using K-fold cross validation

Model	Mean Squared Error
Ridge Regression	1.47
Bayesian Ridge Regression	1.66
ElasticNet Regression	1.70
KNearestNeighbors Regression	2.07
Linear Regression	1.92

Table 4.2: Comparison of various machine learning models on pseudo-labled data in terms of mean-square-error (MSE), mean-absolute-error (MAE), coefficient of determination (R2) score, and explained-variance-score (EVS).

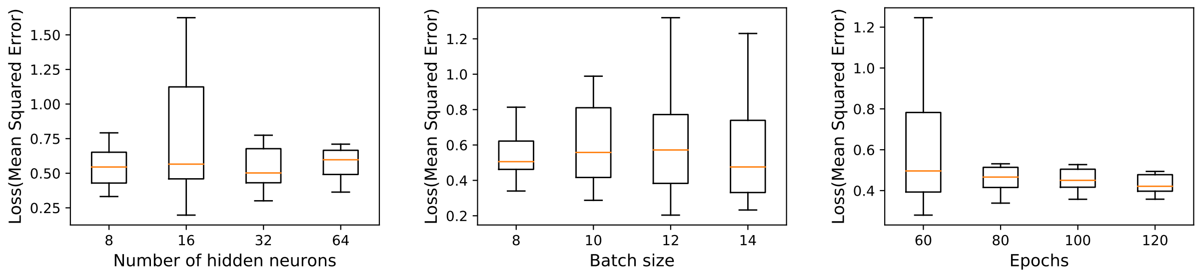
Model	MSE	MAE	R2	EVS
Neural Network	0.43	0.50	0.63	0.75
Ridge Regression	0.63	0.64	0.45	0.70
Support Vector Regression	0.66	0.65	0.42	0.70
Linear Regression	0.63	0.64	0.45	0.71
Lasso Regression	0.64	0.64	0.44	0.71

4.2.2 Semi-supervised learning

We needed unlabeled data to perform pseudo-labeling. For this, we used weekly evapotranspiration and rainfall data from the WMY weather station for College Station. The data contained evapotranspiration and rainfall amounts for the years 2015-2018. The number of unlabeled points numbered 202. We used the above methodology to augment the number of data samples using the best performing model from cross validation. Thus, the size of the training set increased from 15 to 220. To evaluate the performance of the regressor trained on the combined labeled and unlabeled dataset, we had separated a hold-out/test set before. We saw a boost in the performance of Ridge Regression trained on the combined dataset versus the smaller dataset as expected. The mean squared error (MSE)on the test set improved from 0.64 to 0.63. We then trained other models on the augmented dataset. The results of the same are presented in Table 4.2.

4.2.3 Neural network training

The biggest advantage of applying semi-supervised learning was the augmentation of the training data. This enabled us to train an Artificial Neural Network(ANN), and leverage its ability to map non-linear relationships between features. There are many factors that affect the predictive ability of a neural network and they need to be tuned to maximize predictive ability. These factors are called hyper-parameters and the process of finding optimal hyper-parameters is called hyper-parameter tuning. The hyper-parameters we chose to optimize were number of input neurons, batch size and the number of training epochs. Other parameters were kept constant, such



(a) Changing the number of neurons over 60 training epochs, and a batch size of 8

(b) Changing the batch size for 8 hidden layer neurons over 60 training epochs

(c) Changing the number of training epochs for 8 hidden layer neurons and a batch size of 8

Figure 4.1: Effect of different parameters of ANN on the actual performance

as the type of optimizer and the learning rate. We used the Adam optimizer and a learning rate of 0.001. The activation function used over the entire network was ReLU. All evaluations of the neural network were validated over 20 iterations.

Number of hidden neurons: We varied the number of neurons in the first hidden layer of the neural network and the results of the tuning are shown in Figure 4.1a. Hidden layers are the most important part of a neural network. They perform the function of distilling patterns from data that cannot be performed by off-the-shelf machine learning algorithms. The number of hidden neurons plays an active role in the determining the predictive ability of the neural network. We varied the number of neurons from 8 to 64. Higher number of neurons lead to a higher average Mean Squared Error(MSE). As seen from the figure, we found that having 8 neurons in the first hidden layer gave us the best scores on the test set.

Batch size: Since the size of the dataset is large, we divide the dataset into batches that are passed through the network. Finding the right batch size ensures we have a good representation of the dataset and prevents over-fitting. We varied the batch size from 8 to 14 and report the resultant average error rate in Figure 4.1b. We observe that a batch size of 8 is optimal with an average error rate of 0.55. Larger batch sizes leads to the presence of outliers.

Epochs: As discussed previously, the dataset is divided into batches that are passed through the ANN. A neural network is said to be trained through one epoch when the entire dataset, divided into batches, has been passed through the ANN as part of forward propagation, and the network's weights have been adjusted by the optimization algorithm. Training the network on very few epochs leads to under-fitting, and training it on too many epochs leads to over-fitting. We experimented with the total number of epochs used for training and the results are shown in Figure 4.1c.

4.3 Comparison with state-of-the-art irrigation controller

4.3.1 Configuring Rachio

The irrigation controller we chose for evaluation was the Rachio Generation 2. Rachio has a feature where information about a sprinkler zone can be customized. This helps Rachio personalize the runtime for each sprinkler zone. Using data gathered from our site survey , we ensured that the parameters in Rachio's 'Advanced Zone Configuration' was as personalized as possible. It must be noted that the Rachio has types of irrigation schedules, with varying degrees of control given to Rachio's weather-aware decision-making system called 'Weather Intelligence™' over the irrigation schedule. The 'manual' mode is one that is most similar to that of legacy irrigation controllers, where a fixed schedule runs irrespective of weather conditions. The 'Flex monthly' schedule accommodates for changes in watering frequency and duration on a monthly basis. The 'Flex daily' schedule is considered to be the most optimized of the three schedule types. However, it dynamically changes the watering frequency on a daily basis and holds off watering if rainfall is predicted in the near future. We found that this interfered with our ability to conduct evaluation at the site thanks to inclement weather during the time-frame of evaluation. So we used the 'Flex Monthly' schedule, which only skipped irrigation if rain had occurred previously and the soil was deemed to be saturated, as the basis for evaluation against WaRPIC.

4.3.2 Comparing MAR and water usage

We conducted four evaluations where the MAR as predicted by WaRPIC and Rachio were compared along with estimations of water usage. The ground truth measurement of the same was used as a yardstick to determine the efficiency of the water application. The results are presented in

Figure 4.2. It is clear from the figure that the predictions of WaRPIC come very close to the ground truth whereas those of Rachio, despite the advanced sprinkler zone configuration, are much higher. In some cases, it must be noted, that the ground truth value was higher than WaRPIC's prediction. This could mean the application of water was less than efficient. However, since the magnitude of error was very less, we feel that the effect of the application of water wouldn't be too different from that of the ground truth. We also estimated the potential water usage of applying water according to both strategies. Upon consultation with an irrigation specialist, we were informed that the valves used on the site where we conducted experiments were 5/8" valves, and used 40 IPS PVC pipes. We used the table for calculating flow (F) given friction losses for PVC pipes as given in [47]. Thus, we obtain the volume of water wasted as runoff, known as Runoff Volume (RV). The value of F used in our calculations was 12 gallons/minute. The equation used: $RV = F \times (PR - GT)$

Where PR is the predicted runtime, while GT is the ground truth. We computed the RV for WaRPIC and Rachio and found that, over the course of our trials, WaRPIC would've lost 4.08 gallons of water, while Rachio would've wasted 156.72 gallons. It is clear from the results of our evaluation that the water wastage on a cycle-by-cycle basis is much lower for WaRPIC when compared to Rachio (2.6%).

4.4 Household deployments

We deployed WMY modules in the residences of 12 homeowners based in College Station, Texas. An irrigation specialist from Texas A&M AgriLife Extension installed the WMY modules at the residences. The modules were set up in the basic mode of operation. A site survey was conducted of each home's landscape. Data was collected about each sprinkler zone such as the plant type, sprinkler head type, sprinkler runtime, watering days ,etc. This helped us estimate the water savings resulting from the installation of the WMY module at the homeowner's residence. As described previously in 3.1.5.1, in the basic mode of operation, the module shuts off irrigation before runtime when Water My Yard predicts that watering is not required. This serves as a guarantee of water saving when compared to the previous method of sending text messages/emails with no feedback on whether residents followed through on the watering recommendations.

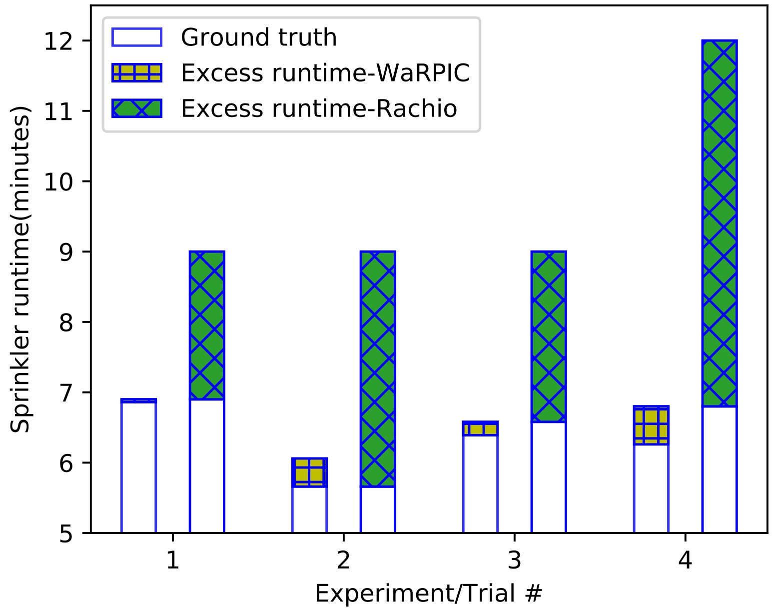


Figure 4.2: Comparison of runtimes predicted by WaRPIC and Rachio vs. ground truth

To estimate potential water savings, we used last year's watering recommendations from Water My Yard for the watering season (May-Oct). Last year, for 15 out of 24 weeks of the watering season, Water My Yard did not recommend watering (NW). Using data from the site surveys, we estimated the Total Weekly Sprinkler Runtime (TWR) for each resident. Making the assumption that all the homes that are a part of our deployment using 5/8" valves and 40 IPS PVC pipes, we estimate flow (F) to be 12 gallons/minute. We can then estimate potential water savings (PWS) for each home using the following equation: $PWS = F \times NW \times TWR$

After obtaining the potential water savings, we use the water rates for residential customers shown in the utilities website for College Station [48] to compute the potential money saved for each customer. The results of our evaluation are presented in Table 4.3.

4.5 Manufacturing cost

WMY has been pitched as a low-cost, practical solution that retrofits legacy irrigation controllers. We present the potential savings that can be brought about by the installation of WMY at

Table 4.3: Projected water and cost savings of household customers for watering season (May-Oct 2019)

Average water savings	38,826 gallons
Average money savings	$ 192.52692
Water Savings	8,640 - 111,780 gallons
Money Savings	$ 20.74 - 587.96

Table 4.4: Manufacturing Cost

Component	Cost
Antenna	$ 3.99
Enclosure	$ 3.72
Micro-controller	$ 2.36
PCB	$ 2.58
Misc. electronic components	$ 6.16
Total	$ 18.81

a residence. The return on investment, the amount of money saved versus the cost of installation of the module, is analyzed here. Table 4.4 shows the manufacturing cost for one WMY module, with the biggest contributors listed. It must be noted that since a small number of WMY modules were manufactured, the costs are much higher than on per-module basis compared to large-scale manufacturing. We believe that economies of scale can drastically reduce costs even further. It can be stated that the WMY module has a huge return on investment thanks to the high volume of water saved each watering season.

5. CONCLUSIONS AND FUTURE WORK

We proposed an Internet-of-Things-based, low-cost solution which control sprinklers to optimally disperse water depending on the sprinkler zone,soil type, weather conditions, etc. We developed modules that can be retrofitted to legacy irrigation controllers. We developed machine learning models trained on data gained from human feedback on runoff. After an initial trial of two weeks, the system learns sufficiently to cope with any weather condition. The system creates an optimal site-specific schedule that considers soil type, slope, soil depth, etc. We trained a machine learning model on data gathered from a site at Texas A&M University's campus. The model is highly accurate and saves more water than a state-of-the-art irrigation controller. We also deployed the modules at residences in College Station, Texas. The homeowners are projected to save around 38,826 gallons of water, worth $192, on average over the course of the watering season (May-Oct 2019).

Some directions for future work include: (1) Adding the ability of triggering solenoids to the WMY module. (2) Integrating weather predictions to the scheduling by RaDE, making it on-par with state-of-the-art in terms of features offered. (3) Adding more security to the WMY module, such as encrypting WiFi password and securing communication via MQTT. (4) Integration with voice-activated services such as Amazon Alexa and smart home platforms.

REFERENCES

[1] R. E. Horton, "The role of infiltration in the hydrologic cycle," *Transactions, American Geophysical Union*, vol. 14, no. 1, p. 446, 1933.

[2] W. DeOreo, P. Mayer, B. Dziegielewski, J. Kiefer, and W. R. Foundation, *Residential End Uses of Water, Version 2*. Water Research Foundation, 2016.

[3] C. Dieter, M. Maupin, R. Caldwell, M. Harris, T. Ivahnenko, J. Lovelace, N. Barber, and K. Linsey, "Estimated use of water in the United States in 2015," *U.S. Geological Survey*, vol. 1441, pp. 1–6, 2010.

[4] C. Milesi, S. W. Running, C. D. Elvidge, J. B. Dietz, B. T. Tuttle, and R. R. Nemani, "Mapping and Modeling the Biogeochemical Cycling of Turf Grasses in the United States," *Environmental Management*, vol. 36, pp. 426–438, July 2005.

[5] A. Nagourney, "California imposes first mandatory water restrictions to deal with drought," 2015. Available at *https://nytimes.com/2015/04/02/us/california-imposes-first-ever-water-restrictions-to-deal-with-drought.html*, last accessed on 05/15/19.

[6] National Geographic, "Clean water crisis facts and information." Available at *https://nationalgeographic.com/environment/freshwater/freshwater-crisis*, last accessed on 05/27/19.

[7] L. M. Shuman, "Phosphorus and nitrate nitrogen in runoff following fertilizer application to turfgrass," *Journal of Environment Quality*, vol. 31, no. 5, p. 1710, 2002.

[8] Z. M. Easton and A. M. Petrovic, "Fertilizer source effect on ground and surface water quality in drainage from turfgrass," *Journal of Environment Quality*, vol. 33, no. 2, p. 645, 2004.

[9] A. Taebi and R. L. Droste, "Pollution loads in urban runoff and sanitary wastewater," *Science of The Total Environment*, vol. 327, pp. 175–184, July 2004.

[10] M. C. Gromaire-Mertz, S. Garnaud, A. Gonzalez, and G. Chebbo, "Characterisation of urban runoff pollution in paris," *Water Science and Technology*, vol. 39, pp. 1–8, Jan. 1999.

[11] S. J. M. Mackenzie L. Davis, *Principles of Environmental Engineering and Science*. McGraw-Hill, 2003.

[12] "Quantifying water savings of homeowner irrigation programs," in *2015 ASABE / IA Irrigation Symposium: Emerging Technologies for Sustainable Irrigation - A Tribute to the Career of Terry Howell, Sr. Conference Proceedings*, American Society of Agricultural and Biological Engineers, Nov. 2015.

[13] B. Cardenas-Lailhacar, M. D. Dukes, and G. L. Miller, "Sensor-based automation of irrigation on bermudagrass during dry weather conditions," *Journal of Irrigation and Drainage Engineering*, vol. 136, pp. 184–193, Mar. 2010.

[14] J. Blonquist, S. Jones, and D. Robinson, "Precise irrigation scheduling for turfgrass using a subsurface electromagnetic soil moisture sensor," *Agricultural Water Management*, vol. 84, pp. 153–165, July 2006.

[15] Hunter Industries Inc, "Hydrawise HC-6." Available at *https://hunterindustries.com/irrigation-product/controllers/hc*, last accessed on 05/15/19.

[16] Rachio Industries Inc., "Rachio Generation 2." Available at *https://rachio.com/gen-2/*, last accessed on 05/15/19.

[17] Rain Bird Industries Inc., "Rain bird esp4me3." Available at *https://store.rainbird.com/esp4me3-indoor-outdoor-120v-irrigation-controller-lnk-wifi-compatible.html*, last accessed on 05/15/19.

[18] Environmental Protection Agency(EPA), "Watersense specification for weather-based irrigation controllers." Available at *https://epa.gov/sites/production/files/2017-01/documents/ws-products-spec-irrigation-controllers.pdf*, last accessed on 05/15/19.

[19] Texas A&M AgirLife Extension, "Water My Yard^{TM}." Available at *https://watermyyard.org*, last accessed on 05/15/19.

[20] S. Fazackerley and R. Lawrence, “Reducing turfgrass water consumption using sensor nodes and an adaptive irrigation controller,” in *2010 IEEE Sensors Applications Symposium (SAS)*, IEEE, Feb. 2010.

[21] N. Sales, O. Remedios, and A. Arsenio, “Wireless sensor and actuator system for smart irrigation on the cloud,” in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, IEEE, Dec. 2015.

[22] D. A. Winkler, R. Wang, F. Blanchette, M. Carreira-Perpinan, and A. E. Cerpa, “MAGIC: Model-Based Actuation for Ground Irrigation Control,” in *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 1–12, April 2016.

[23] D. A. Winkler, M. A. Carreira-Perpinan, and A. E. Cerpa, “Plug-and-play irrigation control at scale,” in *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, IEEE, Apr. 2018.

[24] L. Zhang, W. R. Dawes, and G. R. Walker, “Response of mean annual evapotranspiration to vegetation changes at catchment scale,” *Water Resources Research*, vol. 37, pp. 701–708, Mar. 2001.

[25] W. H. Green and G. A. Ampt, “Studies on soil phyics.,” *The Journal of Agricultural Science*, vol. 4, p. 1, May 1911.

[26] Soil and Water Lab, Cornell University, “Infiltration excess overland flow.” Available at *https://soilandwater.bee.cornell.edu/research/VSA/processes/processesinfil.html*, last accessed 05/15/19.

[27] Soil and Water Lab, Cornell University, “Saturation excess overland flow.” Available at *https://soilandwater.bee.cornell.edu/research/VSA/processes/processessat.html*, last accessed 05/15/19.

[28] G. Aron, “Adaptation of horton and SCS infiltration equations to complex storms,” *Journal of Irrigation and Drainage Engineering*, vol. 118, pp. 275–284, Mar. 1992.

[29] S. W. Bauer, "A modified Horton equation for infiltration during intermittent rainfall," *Hydrological Sciences Bulletin*, vol. 19, pp. 219–225, June 1974.

[30] Texas A&M AgirLife Extension, "Cycle and Soak." Available at *https://agrilifeextension.tamu.edu/blog/2015/08/06/cycle-soak-method-of-lawn-irrigation/*, last accessed on 05/15/19.

[31] S. A. Schumm and G. C. Lusby, "Seasonal variation of infiltration capacity and runoff on hillslopes in western colorado," *Journal of Geophysical Research*, vol. 68, pp. 3655–3666, June 1963.

[32] R. B. Bryan, "Soil erodibility and processes of water erosion on hillslope," *Geomorphology*, vol. 32, pp. 385–415, Mar. 2000.

[33] A. Cerdà, "Seasonal and spatial variations in infiltration rates in badland surfaces under mediterranean climatic conditions," *Water Resources Research*, vol. 35, pp. 319–328, Jan. 1999.

[34] A. Cerdà, "The effect of season and parent material on water erosion on highly eroded soils in eastern spain," *Journal of Arid Environments*, vol. 52, pp. 319–337, Nov. 2002.

[35] C. Fischer, J. Tischer, C. Roscher, N. Eisenhauer, J. Ravenek, G. Gleixner, S. Attinger, B. Jensen, H. de Kroon, L. Mommer, S. Scheu, and A. Hildebrandt, "Plant species diversity affects infiltration capacity in an experimental grassland through changes in soil properties," *Plant and Soil*, vol. 397, pp. 1–16, Dec. 2014.

[36] K. Clancy and V. M. Alba, "Temperature and time of day influence on double-ring infiltrometer steady-state infiltration rates," *Soil Science Society of America Journal*, vol. 75, no. 1, p. 241, 2011.

[37] D. Eapen, M. L. Barroso, G. Ponce, M. E. Campos, and G. I. Cassab, "Hydrotropism: root growth responses to water," *Trends in Plant Science*, vol. 10, pp. 44–50, Jan. 2005.

[38] R. E. Green, *Infiltration of water into soils as influenced by antecedent moisture*. PhD thesis, Iowa State University, 1962.

[39] 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, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[40] D. hyun Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," *Workshop on Challenges in Representation Learning, ICML 2013*, 2013.

[41] L. Bottou, "Stochastic gradient learning in neural networks," in *Proceedings of Neuro-Nîmes 91*, (Nimes, France), EC2, 1991.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[43] F. Chollet *et al.*, "Keras." *https://keras.io*, 2015.

[44] Espressif Systems, "Esp8266-EX." Available at *https://www.espressif.com/en/products/hardware/esp8266ex/overview*, last accessed on 05/15/19.

[45] Amazon Web Services, "AWS-IoT." Available at *https://aws.amazon.com/iot*, last accessed on 05/15/19.

[46] FreeRTOS, "FreeRTOS." Available at *https://freertos.org*, last accessed on 05/15/19.

[47] Hunter Industries Inc., "Friction loss tables." Available at *https://hunterindustries.com/sites/default/files/tech_friction_loss_charts.pdf*, last accessed on 05/15/19.

[48] College Station Utilities, "Water and sewer rates." Available at *http://cstx.gov/index.aspx?page=4005*, last accessed on 05/15/19.