Machine Learning-based Irrigation Control Optimization

Akshay Murthy, Curtis Green, Radu Stoleru, Suman Bhunia, Charles Swanson[†], Theodora Chaspari
Computer Science and Engineering, Texas A&M University
[†]Biological and Agricultural Engineering, Texas A&M University
[akshay.ramuhally,greencurr000,stoleru,sbhunia,clswanson,chaspari]@tamu.edu

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 either used expensive sensors or ignored site-specific factors. In this paper, 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 homeowner-assisted data collection on the landscape. The gathered data is used to build site-specific machine learning models that can accurately predict the Maximum Allowable Runtime (MAR) for each sprinkler zone given weather data obtained from the nearest weather station. 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. The model's prediction is compared with a state-of-the-art irrigation controller and the volume of water wasted by WaRPIC is 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.

CCS CONCEPTS

• **Computer systems organization** → **Sensors and actuators**; *Availability*; • **Applied computing** → *Environmental sciences.*

KEYWORDS

turf-grass, smart irrigation, Internet-of-Things, control, machine learning

ACM Reference Format:
Akshay Murthy, Curtis Green, Radu Stoleru, Suman Bhunia, Charles Swanson[†], Theodora Chaspari. 2019. Machine Learning-based Irrigation Control Optimization. In *The 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '19), November 13–14, 2019, New York, NY, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3360322.3360854

BuildSys '19, November 13–14, 2019, New York, NY, USA

ACM ISBN 978-1-4503-7005-9/19/11…$15.00
https://doi.org/10.1145/3360322.3360854

1 INTRODUCTION

The average US household consumes over 300 gallons of water per day, with roughly 30 percent of that water being used outdoors [12]. Outdoor use of water occurs primarily in landscape management [13], where a major portion goes toward watering lawns. Lawns are estimated to cover an area of 128,000 km^2 in the United States [27]. 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 [28]. With less than 1% of Earth's freshwater available for human use [2], 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, we consider the potential wastage of water occurring due to improper landscape irrigation practices. 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 [31] [15] [32], causing *non-point source pollution* [26]. 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 [1]. 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 [7] [4]. They used soil moisture inputs from the sensors to schedule irrigation of turf-grass. These are not feasibly applicable to the residential landscape as reliable moisture sensors are expensive. Smart irrigation controllers are available on the market, such as the ones by Rachio [24], Hunter [22], Rain Bird [23] that take into account weather conditions using a network of weather stations. According to specifications for smart irrigation controllers [3], they 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. Thus, previous efforts only solve a part of the problem or are expensive solutions inapplicable to the residential landscape.

Landscape irrigation must be performed in an optimal manner that ensures turf is healthy, without causing runoff. Water My Yard [17] provides irrigation recommendations based on local weather data. However, applying these generic recommendations without consideration to site-specific, spatially varying factors such as soil type, soil depth, slope, etc. increases the risk of runoff. These factors vary in each residence on a sprinkler zone-by-zone basis. To prevent runoff, the cycle-and-soak [16] method of irrigation, where irrigation is broken up into multiple cycles, with periods of no irrigation in between cycle to allow the applied water to be absorbed by soil, is recommended. 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 [14] and this enables plant roots to grow deeper in 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 leads to better turf quality in terms of indicators such as root thickness, mat depth, etc. Thus, the landscape must be watered in a manner that applies as much water as possible during each irrigation and prevents runoff caused by site-specific factors.

To address this issue, we propose the Weather-aware Runoff Prevention Irrigation Control (WaRPIC) that aims to optimally irrigate the landscape, taking into account local weather information and the potential for runoff on a sprinkler zone-by-zone basis. It involves data collected from the irrigation landscape, with assistance from the homeowner. The homeowner detects when runoff occurs during irrigation, over the period of 2-3 weeks. The data samples are then used in solving a supervised learning problem that aims to predict the Maximum Allowable Runtime (MAR), or the amount of time irrigation can occur without causing runoff. The predictive ability of the machine learning models is boosted by a semi-supervised learning technique called pseudo-labeling [20]. We also designed WaRPIC modules that retrofit existing irrigation controllers, in order to manipulate the irrigation performed by the controller. This means that there is no need to replace the existing irrigation infrastructure. The installed WaRPIC modules are controlled by the WaRPIC server, which coordinates the irrigation in an optimal manner, taking into account current weather data and the site-specific predictions from machine learning models for current conditions. Thus, WaRPIC can optimally irrigate the landscape in an automated, site-specific manner. The contributions of this paper are as follows:

- The 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.
- The development of a low-cost actuator module that retrofits legacy irrigation controllers. An irrigation schedule has to be established on the irrigation controller.
- An 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.

In Section 2 of this paper, we will discuss in further detail irrigation practices, runoff and the state of the art. Section 3 presents an overview of our system. Section 4 explains the design and implementation of the WaRPIC module and server-end software. We present the results of our evaluation in Section 5. Discussion is presented in Section 6. Finally, Section 7 provides insight into possible future work.

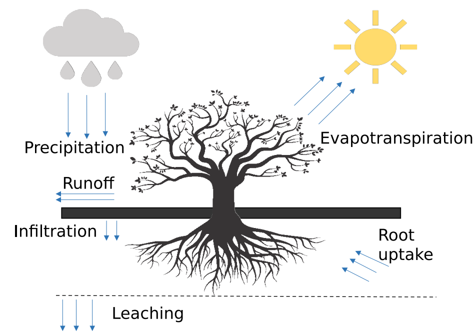


Figure 1: Interaction of atmospheric water with the earth's surface. Water My Yard computes the amount of water needed for plants to grow by considering aspects of the hydrologic cycle

2 BACKGROUND & STATE OF THE ART

In this section, we will introduce the methodology used to estimate the watering requirements of turf-grass taking into account the various process of the hydrologic cycle. We will then provide an introduction to the cycle and soak method of irrigation, the preferred method for runoff prevention. We will also discuss previous efforts that aimed to optimally irrigate the landscape.

2.1 Irrigation Water Management

Water My Yard is a program 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 methodology used to compute weekly sprinkler runtime recommendations is called the *water balance* method [36].

2.1.1 Computing irrigation runtimes. The water-balance method computes the amount of water that has to be applied to turf, called *Watering Requirement* (WR). The WR is computed taking into account various aspects of the interaction of atmospheric water with the earth's surface, as shown in Figure 1. The water to be applied needs to be interpreted in terms of sprinkler runtime. For this,the Precipitation Rate (PR) (the rate at which sprinklers disperse water) of the sprinklers is taken into account. The sprinkler runtime recommendation (RT) is then calculated using: $RT = \frac{WR \times 60}{PR} \quad (minutes)$

2.2 Preventing Runoff

We learned in the previous subsection 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. The irrigation method used to prevent runoff is the *cycle-and-soak* method of irrigation [16]. 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 [3], 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: $MAR = \frac{60 \times ASA}{PR - IR}$

The equation could serve as a benchmark for deciding the cycle time in a landscape irrigation setting. However, the assumption that the infiltration capacity (*IR*) 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 [30], 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 [6] and [8] 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 [9], 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 [18] and [11].

2.3 Previous Works

D. Winkler et al. [35] proposed a solution that creates an optimal irrigation schedule based on a mathematical model generated by using a network of wireless sensor and actuator nodes. They developed a partial differential equation(PDE) based model to model the soil's interaction with water. They formulated a Linear Program (LP) that was solved in order calculate sprinkler run-times 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 [35], D. Winkler et al. [34] 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 [34]. 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.

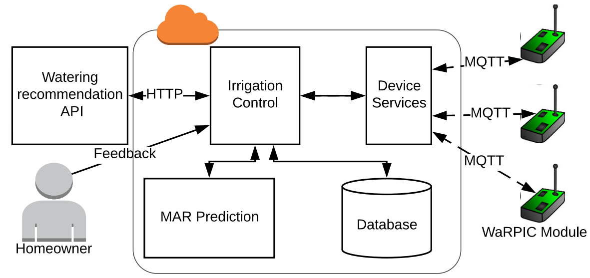


Figure 2: Overview of WaRPIC

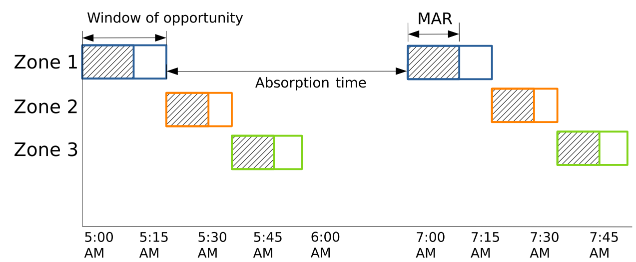


Figure 3: A typical irrigation schedule set on a irrigation controller by a homeowner. Each zone has a start time and window-of-opportunity determined by the irrigation schedule. WaRPIC adjusts the water dispersion by enabling and disabling the valve during the window-of-opportunity

3 WEATHER-AWARE RUNOFF PREVENTION IRRIGATION CONTROL (WARPIC)

In this section, we present the Weather-aware Runoff Prevention Irrigation 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 2 shows an overview of WaRPIC.WaRPIC's installation 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 a set of machine learning models that can 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 legacy irrigation controller is retrofitted with the WaRPIC module. The WaRPIC module handles the actuation of sprinklers during the irrigation schedule. The irrigation schedule serves as a **window-of-opportunity** when the WaRPIC module can impact irrigation. WaRPIC controls irrigation in accordance with an optimal schedule computed from past weather data, as well as data about previous irrigation cycles. The manipulation of the irrigation schedule is depicted in Figure 3.

Table 1: Summarized features of state-of-the-art vs. WaRPIC

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

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 [24] and Hunter [22] 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. A summary of features of the Rachio Generation 2 and Hunter HC-6 and in comparison with WaRPIC is shown in Table 1.

3.1 Prediction 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 [19]. 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 7 days 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.

To further illustrate the variation of MAR with antecedent soil moisture, we present some of the data samples we've collected from a site at the university campus in Table 2. In the first sample, we measured MAR after a period of moderate rainfall (AWA) in the past 7 days. The temporal difference (LWA) since the last application of water was 4 days. The volume of water lost to evapotranspiration (ETO) was also low (0.69 inches). This meant a MAR of 8.58 minutes i.e the sprinklers can be allowed to run for 8.5 minutes without causing runoff. In the second data sample, we observed a MAR of 6.25 minutes. When this sample was collected, 1.06 inches of water had been added to the soil. Also, this sample was collected after the first sample. It is important to note that the MAR reduced considerably from the first sample to the second sample. Furthermore, in the third data sample, the soil was close to saturation in terms of AWA (1.54 inches) and the LWA was also low (1 day). This meant that the MAR was only 4.53 minutes. So, MAR varied dynamically over the course of our data collection.

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.

$$MAR = \boldsymbol{g}(ETO, AWA, LWA) \tag{1}$$

The key advantage here is that $\boldsymbol{g}$ is derived from 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 prediction is implemented 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.

3.2 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 [34], [35] 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

Table 2: Data samples collected from conducting experiments at the site on the university campus

ETO (inches)	AWA (inches)	LWA (days)	MAR (minutes)
0.91	0.97	4	8.58
0.91	1.06	0	6.25
1.07	1.54	1	4.53

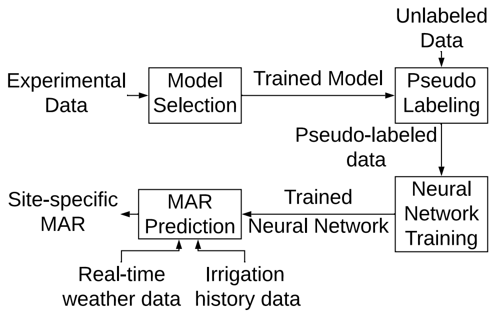


Figure 4: The machine learning pipeline that is used in our methodology.

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 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.

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 applied water for a few days. Next week, we conduct 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.

3.3 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 4.

3.3.1 Data pre-processing. Firstly, the dataset is normalized to follow 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 an unseen test set.

3.3.2 Model selection. WaRPIC uses 5-fold cross validation on the training data to compare the performance of various models, such as Linear Regression, K-Nearest Neighbors Regression, Ridge Regresssion, etc. 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 on our university campus. We used Scikit-learn's machine learning library [29] for the data pre-processing, cross validation and standard machine learning models.

3.3.3 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 [20].* 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 unlabeled data
(3) Retrain another model on the augmented dataset in order to get better predictions.

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.

3.3.4 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 stochastic gradient descent [5] and the Adam optimizer [25] to quickly converge to a solution, means that neural networks 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 Section 5. We used the Python deep learning library Keras [10] to construct and train the neural network .

3.4 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 4, the relay on the WaRPIC module allows us to control the activation of sprinkler valves. However, the WaRPIC 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 WaRPIC module can enable/disable irrigation by

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`$(M, \mathit{history})$:
  $m \leftarrow \varnothing$
  for $\mathit{sample} \in \mathit{history}$ **do**
    $m \leftarrow m \cup M(\mathit{sample})$
  return max(m)

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

Function `sprinklerControl`$(RR, Z, S, R, cycle)$:
  $ETO, RFA, RFT \leftarrow$ `queryWR()`
  $ST \leftarrow S_{cycle}$
  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`$(RR, Z, S, R, cycle + 1)$

controlling the state of the relay. Thus, there is a need for an irrigation schedule that maximizes the ability of the WaRPIC 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, 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 WaRPIC 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 WaRPIC module, irrigation does not continue to the point where it affects the plants' health. To achieve this, WaRPIC uses historical weather data from WaRPIC for the nearest weather station to the homeowner's residence. The machine learning model is developed through 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. 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.

3.5 Irrigation Control

WaRPIC coordinates the actuations of sprinklers installed in irrigation systems retrofitted with WaRPIC modules. It manages sprinkler actuations to provide water to the turf in accordance with local weather conditions while ensuring that runoff doesn't occur. Once the irrigation program has been created on the homeowner's irrigation controller, the module can then work with the watering recommendations by the Watering Recommendation API (based on Water My Yard) on a week-by-week basis to optimally apply water with the goal of preventing runoff.

3.5.1 Irrigation Control Algorithm. : 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 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, WaRPIC can perform precision control of irrigation at the homeowner's residence. The real-time control of sprinklers is shown in Algorithm 2. WaRPIC obtains the recommended weekly runtime recommendation (RR) at beginning of the week. WaRPIC 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. Then, irrigation is enabled for the time period of MAR for each cycle.

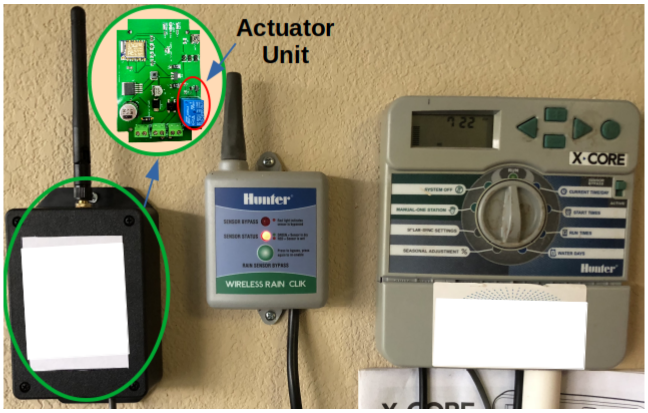


Figure 5: The WaRPIC module post-installation (circled) at homeowner's residence. It can control irrigation, thanks to the actuator unit (circled in red, inset).

4 PROTOTYPE DEVELOPMENT

The WaRPIC module, depicted in Figure 5, is a low-cost module that can automate functioning of legacy irrigation controllers. It has the following features: (1) It can retrofit legacy irrigation controllers via the rain-sensor port. (2) 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. (3) It communicates with the cloud-based central server via the MQTT protocol, which enables functionality such as log-file uploads, firmware updates, and remote control of I/O pins.

4.1 Sprinkler Control Mechanism

The actuation unit (shown in Figure 5) 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 sensor has been activated by rain. We delved into the methodology with which these rain sensors are activated and realized that 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 WaRPIC module uses this principle to take control of the irrigation schedule.

4.2 WaRPIC server 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. We decided to implement the server and web portal using Node.js and host it on an AWS EC2 instance. We decided to use MQTT for communication as it is a lightweight protocol suited for communication with small embedded devices. The central MQTT broker is hosted using a library called Mosca that acts as a wrapper around the MQTT.js library. 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 WaRPIC 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.

Table 3: 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
K-NearestNeighbors Regression	2.07
Linear Regression	1.92

5 EVALUATION AND RESULTS

5.1 Site description

The site where we conducted experiments and gathered data to perform site-specific irrigation was situated on our university campus grounds. The site contained a row of spray sprinklers which could be operated from an irrigation control station. 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.

5.2 Machine learning pipeline

5.2.1 Model selection through cross validation. The results of 5-fold cross validation of different models are presented in Table 3. 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.

5.2.2 Semi-supervised learning. We needed unlabeled data to perform pseudo-labeling. For this, we used weekly evapotranspiration and rainfall data from Water My Yard's weather station for the local area. 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

Table 4: 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

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.

5.2.3 Neural network training. 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 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 6a. 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. A possible reason for this is due to the small number of samples in the dataset. Larger number of neurons results in a larger number of parameters that need to be learned.

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 6b. 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 6c. Training the network on very few epochs (e.g., 60) leads to under-fitting, while increasing the number of epochs between 80 and 120 seems to benefit the performance on the test data.

5.3 Comparison with state-of-the-art

We also performed an evaluation of WaRPIC against the state-of-the-art in WiFi-enabled smart irrigation controllers.

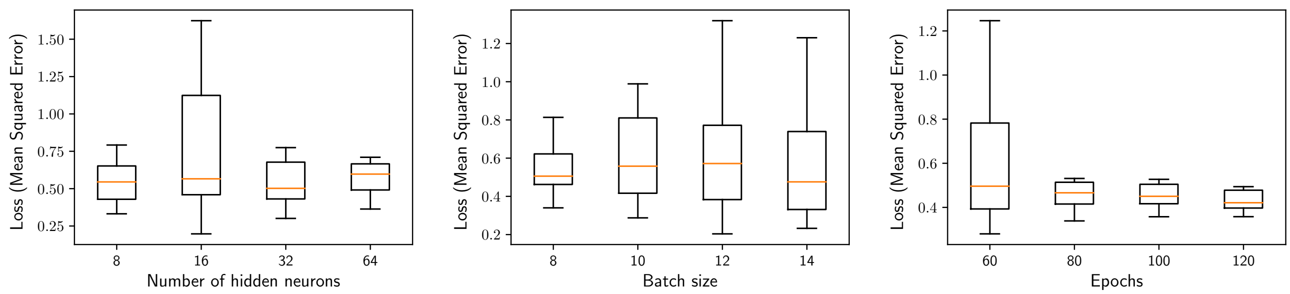
5.3.1 Configuring smart controller. We chose a smart irrigation controller for evaluation [24]. It has a feature where information about a sprinkler zone can be customized. This helps the controller personalize the runtime for each sprinkler zone. Using data gathered from our site survey , we ensured that the parameters were as personalized as possible. It must be noted that the controller had different types of irrigation schedules, with varying degrees of control. 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.

5.3.2 Comparing predicted MAR and water usage. We conducted four evaluations where the MAR as predicted by WaRPIC and the smart controller were compared along with estimations of water usage. The ground truth measurement of the MAR was used to determine the efficiency of the water application. The results are presented in Figure 7. It is clear from the figure that the predictions of WaRPIC come very close to the ground truth whereas those of the smart controller, despite the advanced sprinkler zone configuration, are much higher. 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 [21]. Thus, we obtain the volume of water wasted as runoff, known as Runoff Volume (RV), taking into account the predicted runtime (PR) and the ground truth (GT). The equation is given by: $RV = F \times (PR - GT)$

The value of F used in our calculations was 12 gallons/minute. We computed the RV for WaRPIC and the smart controller using the equation, and found that, over the course of our trials, WaRPIC would've lost 4.08 gallons of water, while the smart controller would've wasted 156.72 gallons. It is clear from that water wastage on a cycle-by-cycle basis is much lower for WaRPIC when compared to the smart controller (2.6%).

5.4 Household deployments

We deployed WaRPIC modules in the residences of 12 homeowners in a city in Texas. An irrigation specialist installed the WaRPIC 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 WaRPIC module at the homeowner's residence. The module shuts off irrigation before runtime when WaRPIC predicts



(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 and a batch size of 8

Figure 6: Effect of different parameters of ANN on the actual performance

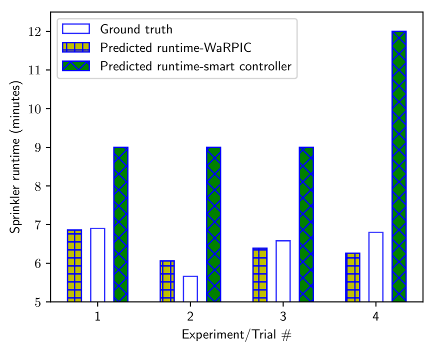


Figure 7: Comparison of runtimes predicted by WaRPIC and the smart controller vs. ground truth

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.

5.4.1 Water Savings. 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 by: $PWS = F \times NW \times TWR$

After obtaining the potential water savings, we use the water rates for residential customers shown in the city's utilities website [33] to compute the potential money saved for each customer. The results of our evaluation are presented in Table 5.

5.4.2 Service Availability. The deployed WaRPIC modules communicate using the MQTT protocol with the WaRPIC server. Over the course of the deployment, the modules would lose connection and attempt to reconnect. The disconnects were primarily due to network outages. The WaRPIC module logs the details of these

Table 5: Projected water and cost savings of household customers for watering season (May-Oct 2019)

Average water savings	38,826 gallons
Average money savings	$ 192.53
Water Savings	8,640 - 111,780 gallons
Money Savings	$ 20.74 - 587.96

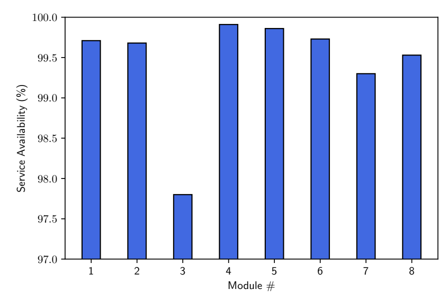


Figure 8: Service Availability of household deployment

disconnections such as the time of disconnection and time at which the module could reconnect to the server. Analyzing these device logs allowed us to calculate the duration of the disconnections. We can then compute the Service Availability (SA) of the internet connection of each deployment. The SA is calculated as a percentage of time the module can communicate with the sever versus the total duration of the deployment. Figure 8 shows the service availability of 8 of the modules deployed in households. We observe that although a majority of the modules have a high service availability, that of module #3 is low (98%).

6 DISCUSSION

We came across some limitations and issues during implementation of WaRPIC and household deployment. Firstly, we tried to use inexpensive soil moisture sensors to measure antecedent soil moisture. These sensors were highly inaccurate, and couldn't function reliably in an outdoor environment. Accurate soil moisture sensors are very expensive ($500+ for one sensor). Secondly, we overestimated the signal strength of the home's wireless AP in the area of the home where the irrigation controller was installed (typically the garage). This caused issues in homes where the router was installed far away from the garage. The WaRPIC module could not be installed in such houses. Thirdly, we couldn't retrieve the logs from all the deployed modules. Two modules had very weak connections to their wireless APs. This meant that the modules frequently disconnected and reconnected to the WaRPIC server. Consequently, the logs stored in the MCU's flash memory would fill up very quickly. Two deployed modules couldn't upload their logs. We believe that the outgoing port used for FTP is blocked in the wireless AP's firewall. Finally, while testing with different brands of irrigation controllers, we found that the rain sensor activation mechanism isn't the same in every irrigation controller. Some manufacturers do not use the method of detecting resistance across the terminals of the rain sensor port to halt irrigation. The WaRPIC module couldn't be used with such irrigation controllers as retrofitting them with the module would cause irrigation to be disabled permanently.

7 CONCLUSION 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 derived machine learning models using data collected by human feedback on runoff. After an initial trial of two weeks, the system learns sufficiently well 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 on our university 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 an urban area. 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 WaRPIC 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 WaRPIC 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] 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* (Nov. 2015), American Society of Agricultural and Biological Engineers.

[2] Competing for clean water has led to a crisis. www.nationalgeographic.com/environment/freshwater/freshwater-crisis, Jan 2017.

[3] Agency(EPA), E. P. Watersense specification for weather-based irrigation controllers. https://www.epa.gov/sites/production/files/2017-01/documents/ws-products-spec-irrigation-controllers.pdf, 2011.

[4] Blonquist, J., Jones, S., and Robinson, D. Precise irrigation scheduling for turfgrass using a subsurface electromagnetic soil moisture sensor. *Agricultural Water Management 84*, 1-2 (July 2006), 153–165.

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

[6] Bryan, R. B. Soil erodibility and processes of water erosion on hillslope. *Geomorphology 32*, 3-4 (Mar. 2000), 385–415.

[7] Cardenas-Lailhacar, B., Dukes, M. D., and Miller, G. L. Sensor-based automation of irrigation on bermudagrass during dry weather conditions. *Journal of Irrigation and Drainage Engineering 136*, 3 (Mar. 2010), 184–193.

[8] Cerdà, A. Seasonal and spatial variations in infiltration rates in badland surfaces under mediterranean climatic conditions. *Water Resources Research 35*, 1 (Jan. 1999), 319–328.

[9] Cerdà, A. The effect of season and parent material on water erosion on highly eroded soils in eastern spain. *Journal of Arid Environments 52*, 3 (Nov. 2002), 319–337.

[10] Chollet, F., et al. Keras. https://keras.io, 2015.

[11] Clancy, K., and Alba, V. M. Temperature and time of day influence on double-ring infiltrometer steady-state infiltration rates. *Soil Science Society of America Journal 75*, 1 (2011), 241.

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

[13] Dieter, C., Maupin, M., Caldwell, R., Harris, M., Ivahnenko, T., Lovelace, J., Barber, N., and Linsey, K. Estimated use of water in the united states in 2015. *U.S. Geological Survey 1441* (2010), 1–6.

[14] Eapen, D., Barroso, M. L., Ponce, G., Campos, M. E., and Cassab, G. I. Hydrotropism: root growth responses to water. *Trends in Plant Science 10*, 1 (Jan. 2005), 44–50.

[15] Easton, Z. M., and Petrovic, A. M. Fertilizer source effect on ground and surface water quality in drainage from turfgrass. *Journal of Environment Quality 33*, 2 (2004), 645.

[16] Extension, T. A. A. Cycle and soak. agrilifeextension.tamu.edu/blog/2015/08/06/cycle-soak-method-of-lawn-irrigation/.

[17] Extension, T. A. A. Water my yard. watermyyard.org.

[18] Fischer, C., Tischer, J., Roscher, C., Eisenhauer, N., Ravenek, J., Gleixner, G., Attinger, S., Jensen, B., de Kroon, H., Mommer, L., Scheu, S., and Hildebrandt, A. Plant species diversity affects infiltration capacity in an experimental grassland through changes in soil properties. *Plant and Soil 397*, 1-2 (Dec. 2014), 1–16.

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

[20] hyun Lee, D. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, 2013.

[21] Inc., H. I. Friction loss tables. hunterindustries.com/sites/default/files/tech_friction_loss_charts.pdf, 2012.

[22] Inc., H. I. Hydrawise hc-6. hunterindustries.com/irrigation-product/controllers/hc, 2016.

[23] Inc., R. B. I. Rain bird esp4me3. store.rainbird.com/esp4me3-indoor-outdoor-120v-irrigation-controller-lnk-wifi-compatible.html, 2018.

[24] Inc., R. I. Rachio generation 3. rachio.com/rachio-3/, 2016.

[25] Kingma, D. P., and Ba, J. 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).

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

[27] Milesi, C., Running, S. W., Elvidge, C. D., Dietz, J. B., Tuttle, B. T., and Nemani, R. R. Mapping and modeling the biogeochemical cycling of turf grasses in the united states. *Environmental Management 36*, 3 (July 2005), 426–438.

[28] Nagourney, A. California imposes first mandatory water restrictions to deal with drought. nytimes.com/2015/04/02/us/california-imposes-first-ever-water-restrictions-to-deal-with-drought.html, 2015.

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

[30] Schumm, S. A., and Lusby, G. C. Seasonal variation of infiltration capacity and runoff on hillslopes in western colorado. *Journal of Geophysical Research 68*, 12 (June 1963), 3655–3666.

[31] Shuman, L. M. Phosphorus and nitrate nitrogen in runoff following fertilizer application to turfgrass. *Journal of Environment Quality 31*, 5 (2002), 1710.

[32] Taebi, A., and Droste, R. L. Pollution loads in urban runoff and sanitary wastewater. *Science of The Total Environment 327*, 1-3 (July 2004), 175–184.

[33] Utilities, C. S. Water and sewer rates. http://www.cstx.gov/index.aspx?page=4005, 2019.

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

[35] Winkler, D. A., Wang, R., Blanchette, F., Carreira-Perpinan, M., and Cerpa, A. E. Magic: Model-based actuation for ground irrigation control. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)* (April 2016), pp. 1–12.

[36] Zhang, L., Walker, G. R., and Dawes, W. R. Water balance modelling water balance modelling: Concepts and applications.