Model-Based Simulations for Turfgrass Irrigation Scheduling

Lucas Major, Logan Gall
Advisor: Dr. Josh Friell

Introduction

Golf courses require significant amounts of water for irrigation to maintain healthy grass. A lack of water can introduce drought stress and reduced turf quality, but over-irrigation can also lead to waterlogging and excessive water use. As a result, properly balanced irrigation management is essential to maintaining high quality turfgrass while staying environmentally sustainable. Traditionally, practitioners schedule irrigation based on fixed intervals, visual observations, or evapotranspiration (ET) measurements. ET is an estimate of the amount of water lost in a system, and it is dependent on the system's temperature, humidity, solar radiation, and wind speed. While ET provides an estimate of the water lost, soil moisture sensors (SMS) can give an exact measurement of the water content in the soil. When used in conjunction with ET estimates, SMS measurements provide the ability to model turf water conditions.

Building off of initial research from USGA study #2021-15-739, this project models the future volumetric water content (VWC) in a plot of turfgrass.[1] Then, the model is integrated into a simulation program to help a user determine when irrigation of a plot should occur in order to optimize water usage according to a VWC threshold of their preference. To do this, we utilized existing soil moisture and ET time history datasets to train three types of models: linear, random forest, and neural network. A final best model was chosen based on its performance of making VWC predictions over a validation time frame. The final model was then adapted to allow user inputs. A pipeline was created where a user can input their turf's current VWC measurement and forecasted ET & rainfall of the area. These inputs will be used to predict VWC given multiple irrigation threshold values. Using outputting data and plots, the user can decide on a best threshold to optimize water usage or hit a target VWC value. With this tool, golf course superintendents will be able to sustainably manage their courses. Reducing unnecessary irrigation events leads to water, cost, and labor savings. Optimal turf health leads to improved playability, better catering to the golfers using it every day. The findings of this project have the ability to make a significant impact on enhancing the sustainability of the game to golf courses wanting an environmentally and economically sound future.

Goals

For this project, we had two primary goals. First, we wanted to model turf volumetric water content (VWC) using data provided from a previous USGA turfgrass irrigation study.[1] Being able to predict future VWC allows us to know how a grass may grow and react with upcoming weather.

Our second goal was to apply these VWC models to create a simple threshold-based irrigation management program. We want this to be an accurate program that a course superintendent could easily use by inputting as few values as needed. This program can be used to forecast VWC and help the user decide the best time to irrigate in order to optimize total water usage while still maintaining healthy turf grass.

Input Data

We utilized two data sets in this predictive analysis. First is soil moisture data. These data were previously collected in the mentioned irrigation study. In summary, a set of 30 turfgrass plots were analyzed in Saint Paul, MN collecting data on grass water usage and health during dry down between July 1 and September 30, 2022. These plots were kept in open conditions and maintained like a typical golf fairway. They were set up in a randomized complete block design, with three replications, two grass species, and had irrigation schedules based on five different Plant Available Water (PAW) thresholds. The five PAW values (15%, 30%, 45%, 60%, and 75% PAW) were tested and scheduled irrigation events during dry down. If the PAW value of the plant (converted to VWC) drops below the threshold value, a 0.25" irrigation event occurs after logging data values. Three replications of two turfgrass species (Kentucky Bluegrass and Creeping Bentgrass) were used on top of native Waukegan silt loam soil.

Mean soil moisture (VWC) readings were reported every 1-4 days using Campbell Scientific CS655 soil moisture sensors. While the sensors recorded VWC every 15 minutes, our data contained the average VWC through an entire report day. This was the main response for our project. Other plant response measurements, such as NDVI and photochemical efficiency, were also collected but do not fall within the scope of this report. These additional measurements are functions of VWC, and are typically time-delayed from a quick VWC response.

The second data set, Weather & ET, contains daily weather observations values collected from an on site Spectrum Technologies WatchDog 2900ET weather station. These weather observations, overviewed in **Table 1**, are used to calculate a day's net reference evapotranspiration (ET) through a set of equations.[2] ET is a known useful measure for estimating water lost due to evaporation in a system and can be used in this project to help model the total water balance of a given plot. This data set also includes net water balance calculations equal to ET + Rainfall each day.

Table 1: Weather data measured variables.

Variable(s)	Description
Date	Measurement date
Low, Mean, High Temperature (°F)	Temperature measurements
Low Relative Humidity (%), High Relative Humidity (%), Solar Radiation (W/m^2), Windspeed (mph), Net Rainfall (in.)	Additional weather measurements, most used for ET calculation.
Reference ET (in.)	Calculated ET
Net Water (in.)	Net water balance calculations

Methods

Our methodology has four parts. Early on, we focused on cleaning the data and preparing it for further analysis. We then explored the data, looking to identify key factors for predictive analysis. Once we understood the data, we worked with 3 types of models to narrow on key variables. Finally, we picked the best model to implement into a simulation program that predicts VWC values given user input. This was further verified by inputting testing data into the simulation program. Each of these steps is described in detail below.

Data Cleaning

To begin cleaning the data, we excluded all observations containing 'NA' for the variable *Mean.VWC* in the soil moisture dataset. *Mean.VWC* is the average VWC reading through that day of measurement. We want to perform an inner join of the soil moisture dataset with the ET data. This allows us to add ET information to each observation of soil moisture. To do this, we joined via the 'date' string. The format mismatched slightly between the sets, so the date values were rewritten to match the date notation of the ET dataset so an inner join could be completed.

Now both datasets are combined. A new variable titled *next VWC Observation* was created containing the *mean.VWC* of the plot's next recorded observation. Additionally, the variable *days Until Next Observation* was calculated indicating the number of days until the current plot's next observation. This number ranges from 1-4 days. A variable *cumulative Water* was calculated, which is the cumulative water balance gained or lost from irrigation, ET, and rainfall between the current and next observations. For the *cumulative Water* variable, an irrigation event adds 0.25 inches of water.

Finally, the rows from the last day of the study were removed as they have no next observation. In the end, we had a total of 1407 observations.

Data Exploration

Our goal is to predict the next observed VWC value for a given turf plot. As we planned our analysis, we wanted to make this model as simple as possible while maintaining accuracy. As mentioned above, most weather variables (Solar Radiation, Humidity, Temp, ect.) are used to directly calculate ET. We looked at using ET to keep a simple model rather than inputting multiple other values. We also chose to explore using the new *cumulative Water* variable, which combines irrigation, precipitation, and ET. All three values are important for the simulation as a user will input forecasted values to estimate water gain/loss and also see effects with varied irrigation thresholds.

We created plots visualizing these variables against the change in VWC value from the next and current observations. We noted factors that increased or decreased the VWC value leading to the next observation.

Model Creation

Given our goal to predict the next VWC observation, we chose to evaluate three types of regression models for our analysis: Linear, Random Forest, and Neural Networks.

For all our model evaluations, we chose to cross validate using a 60/40 train/test data split. 60% of the data is independently used for training the model, and the other 40% is used for predictions and testing accuracy. The soil moisture data has anywhere from 1 to 4 days between observations. For a useful and precise user program we want to work only in single day steps. This means we had to create a specialized cross validation script that applies the model multiple times to predict multiple days out.

The cross validation script first splits the data to a train/test set and trains a specified model. After that, it iterates through the testing set, reading the current *Mean VWC* value, and *n* number of *days Until Next Observation*. It then pulls a single day of weather values (from the Weather and ET dataset). These values are directly inputted into the model for prediction, creating an output based on a single day's data. This prediction is repeated in a loop *n* times, updating the weather and predicted *Mean VWC* values. The final output *Mean VWC* prediction can be compared to the given *next VWC Observation* so we see how the models perform when working in single day steps.

For each model we manually tuned the models using different variable combinations (including polynomial and interaction relationships). We picked the best model based on the average mean squared error (MSE) and R^2 values over 100 repetitions of cross validation. The simple repetition is quick to implement and reduces doubt caused by poor training/testing splits.

The linear model is a simple combination of weighted data inputs. This type of model is quick to calculate and can accurately evaluate linear relationships between a variable and the responding *next VWC*. Due to its simplicity, this type of model is inflexible and tends to have lower accuracy when there are non-linear relationships in the data that are not accounted for.

The random forest model is a machine learning model that creates randomized subsets of the data, and then creates multiple decision trees to model the next VWC response. In a decision tree, the data is split into subsets based on the value of one of the input features, with the goal of separating the data into groups with similar output values. The quality of the split is measured by the information gain, which takes into account the distribution of output values in each subset of data. This process of splitting is continually repeated for the tree, with each subset of data being further split into smaller subsets based on the values of input features. The splitting ends and a leaf node is created once a stopping criterion has been met, such as reaching either the node size parameter or the maximum depth. Additionally, each decision tree is built on a random subset of the input features and a random subset of the training data. The final prediction is made by averaging the predictions of all trees in the forest. This model works well with linear and nonlinear data as it makes no assumptions about the underlying data distributions. This is good for our data, as some responses may have a nonlinear relationship with the next VWC response. The random forest R function (from the 'randomforest' library) has two main parameters: 'trees' and 'node size.'[3] The trees parameter decides how many randomized decision trees are built and combined for the model. Node size decides the bin size of a leaf of the decision tree. A larger number averages more observations at an endpoint and creates a more simple model. A lower number creates a more complex model, but can lead to overfitting. We plotted and manually tuned the number of trees and nodes that the R random forest function uses in order to optimize a testing R^2 value.

A neural network is a highly complex model that works by optimizing a set of 'neurons' that interact with each other through 'hidden layers'. In the model, each neuron takes an input and applies a weight to produce an output that is passed onto the next layer, until the output layer is reached. Neural networks are capable of recognizing complex patterns in data without having to manually define polynomials or interactions, which we decided may be useful with the factors being evaluated. The number of neurons and hidden layers can be adjusted, with a higher number leading to a more complex model, but also leading to a tendency of overfitting. We plotted and manually tuned the number of neurons and hidden layers that the R neural network function (from the 'neuralnet' library) uses to optimize a testing R^2 value.[4]

Simulation

To make the models into a useful real-world application, we created a simulation program similar to the above cross validation script. This program starts by taking user input of a plot's current VWC value. The user is able to define how many days into the future they would like to predict, and input a few days of ET and weather forecasts. With this data, the program will recursively apply the best model (chosen above), estimating the VWC value of the plot day by day.

Just estimating VWC alone is good but given the purpose of our analysis we want to create a tool to assist in optimizing water usage. So, the recursive prediction is repeated 14 total times. In the first iteration, the VWC across X days is predicted with no intervention. For the following 13 iterations, it tests an irrigation PAW threshold between 15% and 75% (by 5% steps). These thresholds are converted to VWC using the following equation specific to the soil and plots used in the USGA irrigation study:

$$VWC_T = LL + \%PAW * (UL - LL)$$

VWC_T is the VWC threshold conversion. LL and UL are the lower and upper limits of VWC field capacity for soil plots. %PAW is the given PAW threshold. Putting together for this study's plots, the equation is:

$$0.0437+\%PAW*(0.4366-0.0437)$$

If on any given day, the ground's water content is predicted to drop below the threshold, an irrigation event is applied.

At the end of the simulations, the program outputs all of the VWC predictions for every day at each threshold level. It also notes on which day an irrigation event should be applied. These can also be plotted by the user at the conclusion of the simulation program. This comparison allows the user to decide on a threshold that allows them to reduce total irrigation events and maintain healthy VWC values.

Results

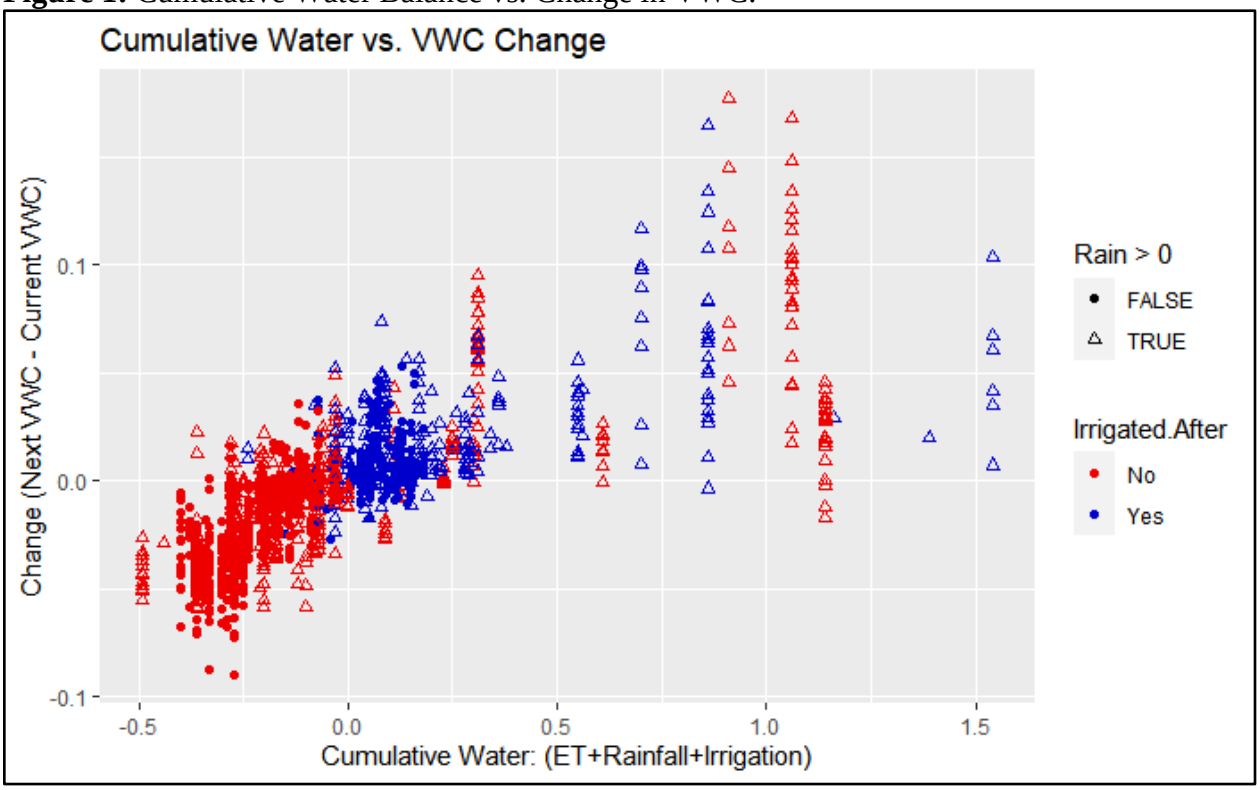
Data Exploration

We found the strongest factor affecting VWC to be overall water balance. With that, we chose to eliminate weather factors that directly calculate ET to simplify our modeling process.

Figure 1 shows the relationship between cumulative water balance and the change in VWC between two observations. The x-axis is the cumulative water balance between the current and next observations, and the y-axis is the change in VWC between the current and next observations. Values above zero on the y-axis mean the VWC increased, and values above 0 on the x-axis means the plot gained water. Right away we can see a positive relationship between the two. As water balance increases, the change in VWC also tends to increase. As we may expect, at low cumulative water balance, there is little rain or irrigation. Similarly, most observations with a positive water balance either had an irrigation event or positive rainfall. Generally, the main takeaway from this plot is that we are able to predict VWC because there is a relationship between it and cumulative water.

We do see some outlying values in this plot, however. At high cumulative water balance values, there is a high spread of change. This could be due to the high soil saturation not accepting any more water. Also, there are points near zero cumulative water balance with either positive or negative change. These differences may be due to odd timing of rain events or other factors that change soil moisture but can't be seen with the single combined value.

Figure 1: Cumulative Water Balance vs. Change in VWC.



After our data exploration, we eliminated many redundant variables, allowing us to create simple models useful for a real-world application. Remaining variables are: *meanVWC*, *daysUntilNextObservation*, *Rainfall*, *ET*, *Irrigation*, and *cumulativeWaterBalance* (calculated from *ET + Rainfall + Irrigation*).

Model Creation

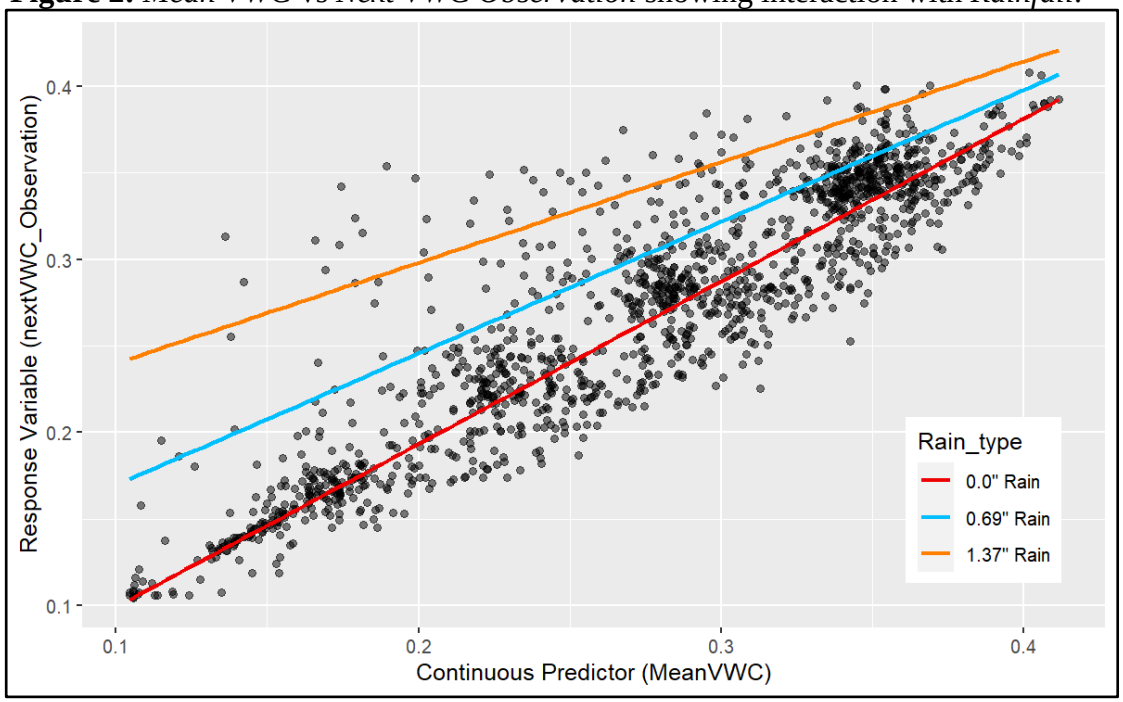
Linear Model

With narrowed down variables, we began testing different combinations within our models. We found the best linear model to contain the following factors:

*Next VWC Observation ~ Mean VWC * Irrigation * Rainfall * ET*

The * in the model terms indicates interaction effects between the variables. An interaction means as one variable changes, another tends to change along with it, causing different effects on the predictor variable. One of the most significant interactions was between the current *Mean VWC* and *Rainfall*. **Figure 2** shows this interaction between the two variables. We can see that when VWC is very low, any rain will cause a significant change. But, when VWC is high, the rain does not change the next observation nearly as much. This is likely due to the ground getting saturated at high VWC values and can't take on any more water.

Figure 2: *Mean VWC* vs *Next VWC Observation* showing interaction with *Rainfall*.



This linear model ends up fitting the data very well, with an average MSE across the 100 iterations of cross validation to be 0.00024. This means for any prediction, we would expect an average error of about plus or minus 0.015 VWC. The R^2 is also strong at 0.95.

Random Forest

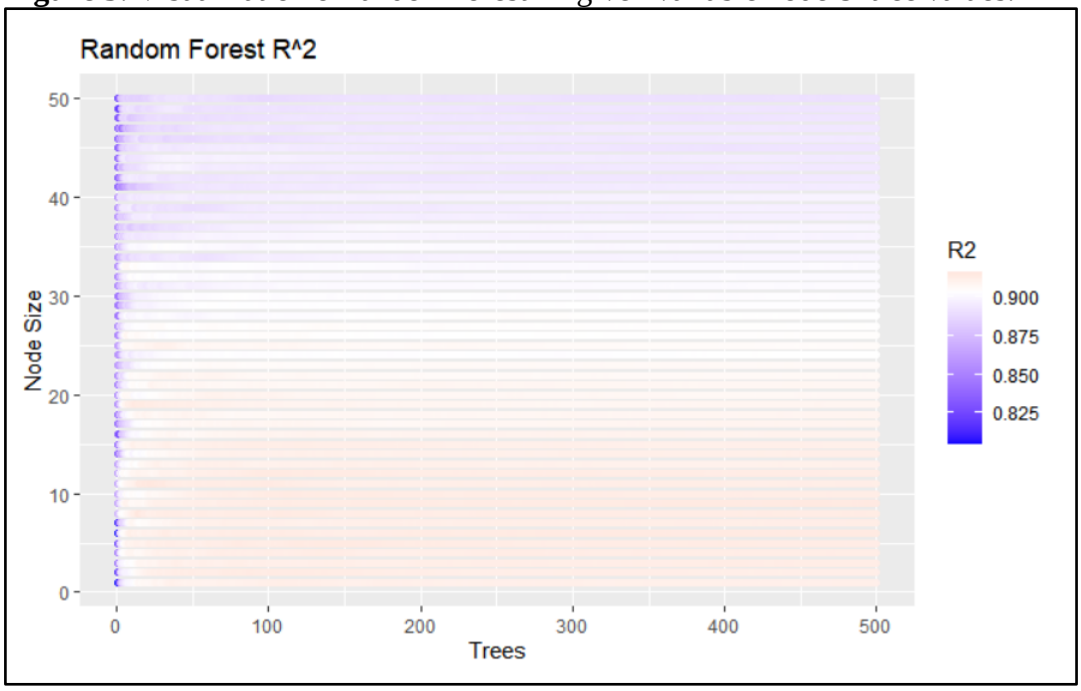
The best random forest model contained the following predictors:

Next VWC Observation ~ Mean VWC + Cumulative Water Balance

Unlike the linear model, the random forest's highest accuracy occurred with only two predictors. This is most likely because random forests are much better at capturing non-linearity between the predictors without the need for interaction effects.

Figure 3 displays the effects on accuracy of changing the random forest model parameters. Immediately, we can see when the number of trees is close to zero, the accuracy suffers the greatest. However, beyond those values close to zero there is only a slight positive increase in the accuracy when raising the number of trees. On the other hand, the node size has a significant impact on the model accuracy. From this plot, a lower node size corresponds to higher accuracy, which makes sense as low node size values create more complex decision trees. Based on the information from this plot there are many valid tree and node combinations. We chose 10 nodes and 200 trees in the final model as an overall balance between accuracy, complexity, and runtime efficiency.

Figure 3: Visualization of random forest R^2 given variable node & tree values.



With the selected parameters, the random forest model had an MSE of 0.00049 and an R^2 of 0.91.

Neural Network

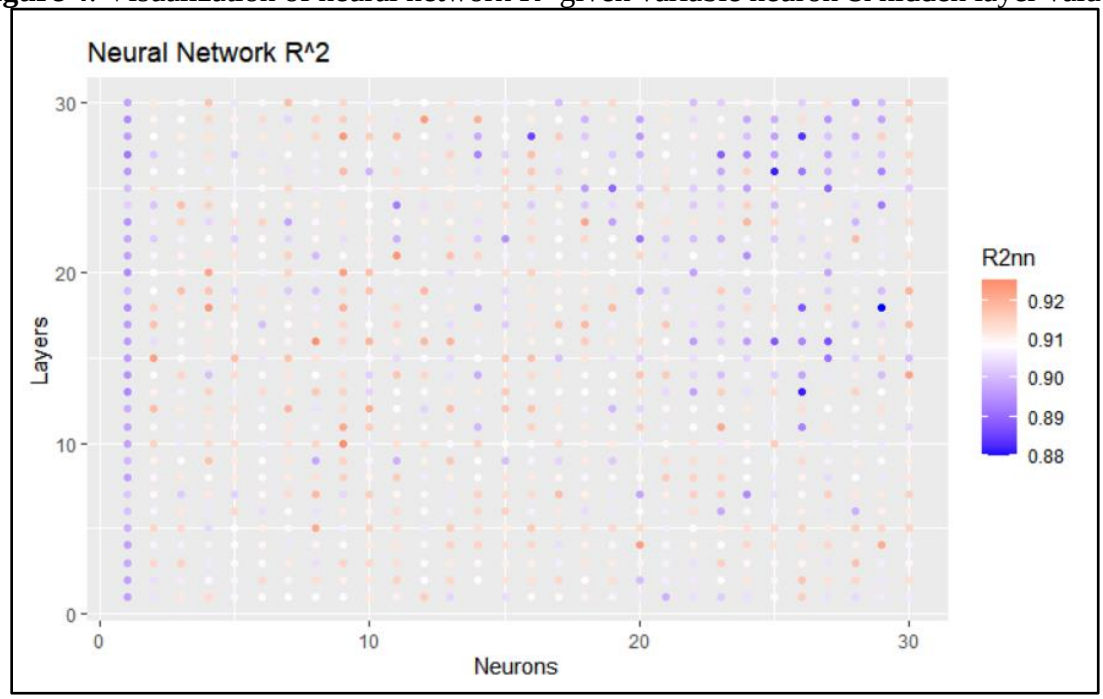
The best neural network model contained the following predictors:

Next VWC Observation ~ Mean VWC + Cumulative Water Balance

Similar to the random forest, the best neural network model consists of only two predictors. Again, this is because neural networks excel in capturing non-linearity between predictors.

Figure 4 models accuracy of the neural network with respect to the number of neurons and hidden layers. The R^2 is low when the number of neurons is close to zero, which makes sense as the model probably is not capturing all of the relationships in the data. Similarly, the R^2 lowers when the number of neurons and hidden layers increase. In this case, the model is most likely overfitting due to overcomplexity. Other than that, however, the trends in this graph aren't as strong compared to the random forest. Neural networks are very abstract, so random variance within the data skews results. Despite this, the range of R^2 values in this plot is narrow, meaning the neural network generally remains consistent in its accuracy even with varying parameter combinations. Neural networks are quite abstract, so adjusting the parameters doesn't nearly have as strong of an effect on accuracy compared to the random forest, but we chose 9 neurons and 10 hidden layers for simplicity and a lower likelihood of overfitting the data.

Figure 4: Visualization of neural network R^2 given variable neuron & hidden layer values.



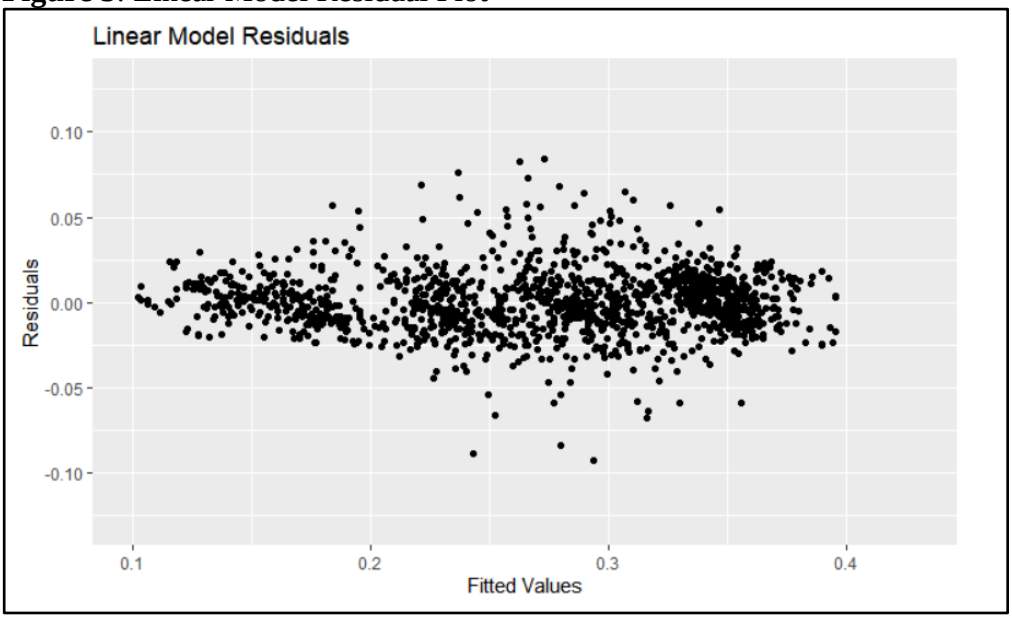
With the selected parameters, the neural network model had an MSE of 0.00049 and an R^2 of 0.91.

Model Comparisons

A residual plot is a graphical representation of the residuals, which are the differences between the observed values and the predicted values. The residual plot provides insight into the quality of the model fit by revealing any trends. If the model fits the data well, the residuals should be randomly scattered around the horizontal line at zero. However, if there is a trend in the scatter, it suggests the model may not adequately capture underlying relationships in the data. Adding higher order terms, interaction effects, considering different transformations, or testing other models are all ways to improve a model that isn't fitting the data. Our linear model performed

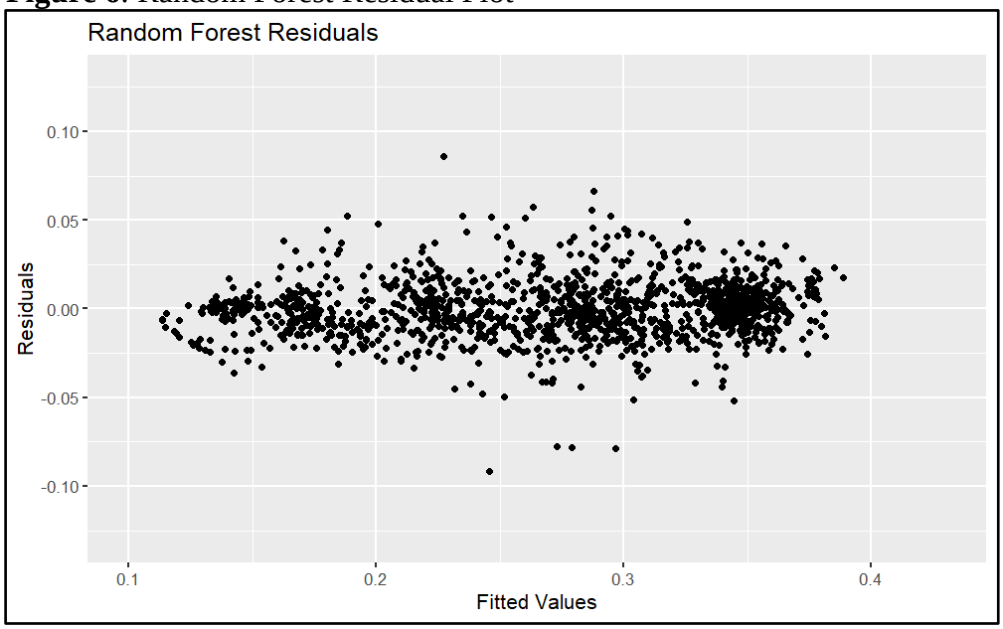
the best after adding interaction effects, and the resulting residual plot in **Figure 5** displays those residuals. The plot supports a good model fit as there are not any strong trends. There is a slight bulge in the middle of the scatter but this is not cause for concern because there is a higher concentration of data points in this region.

Figure 5: Linear Model Residual Plot



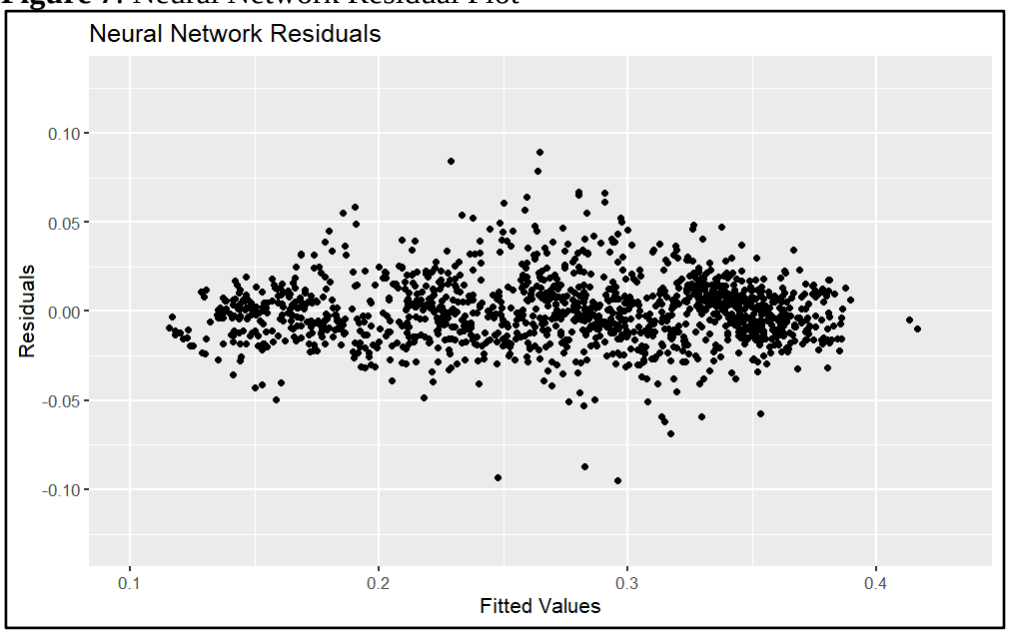
The random forest residual plot in **Figure 6** shows no signs of non-linearity, demonstrating a good fit. Generally, the scatter is slightly more precise compared to the linear residuals, with a few strong outliers.

Figure 6: Random Forest Residual Plot



The neural network residual plot in **Figure 7** appears very similar to the linear model residual plot, indicating the fit is generally good. However, similar to the random forest, some of the outliers in the neural network plot are quite strong.

Figure 7: Neural Network Residual Plot



Simulation Program

The linear model ended with the highest accuracy in our tests, so we chose to implement that into our simulation program. The user inputs a current VWC value, how many days out they would like to predict, a set irrigation amount, and forecasted ET and Rainfall. The weather data can found easily through the National Weather Service or other tools that give these forecasts.[5,6]

To illustrate a use of the simulation, let's use an example using real data. In this example, the user has a current VWC reading of 0.227, wants to predict 3 days into the future, and uses 0.25 inches of water per irrigation. Between day 0 and 1, there is -0.19 ET water loss and 0.0 Rainfall water gain. Between days 1 and 2, there is -0.15 ET and 0.23 Rain. And between days 2 and 3, -0.18 ET and 0.01 Rain.

After the simulation is run, the final predicted VWC is 0.213. The actual result from the data was 0.201. This prediction error across multiple days is smaller than even the single iteration expected error of 0.015.

Tables 2 and **3** show the output of the example data. **Table 2** shows predicted VWC values at each day step within the simulation. **Table 3** similarly shows whether an irrigation event is applied at each day step. Between 0% and 40% PAW thresholds, we can see no irrigation events occur, so the predicted VWC ends up the exact same. As the threshold gets higher, we can see more irrigation and thus more water usage.

Table 2: Example simulation output of VWC predictions.

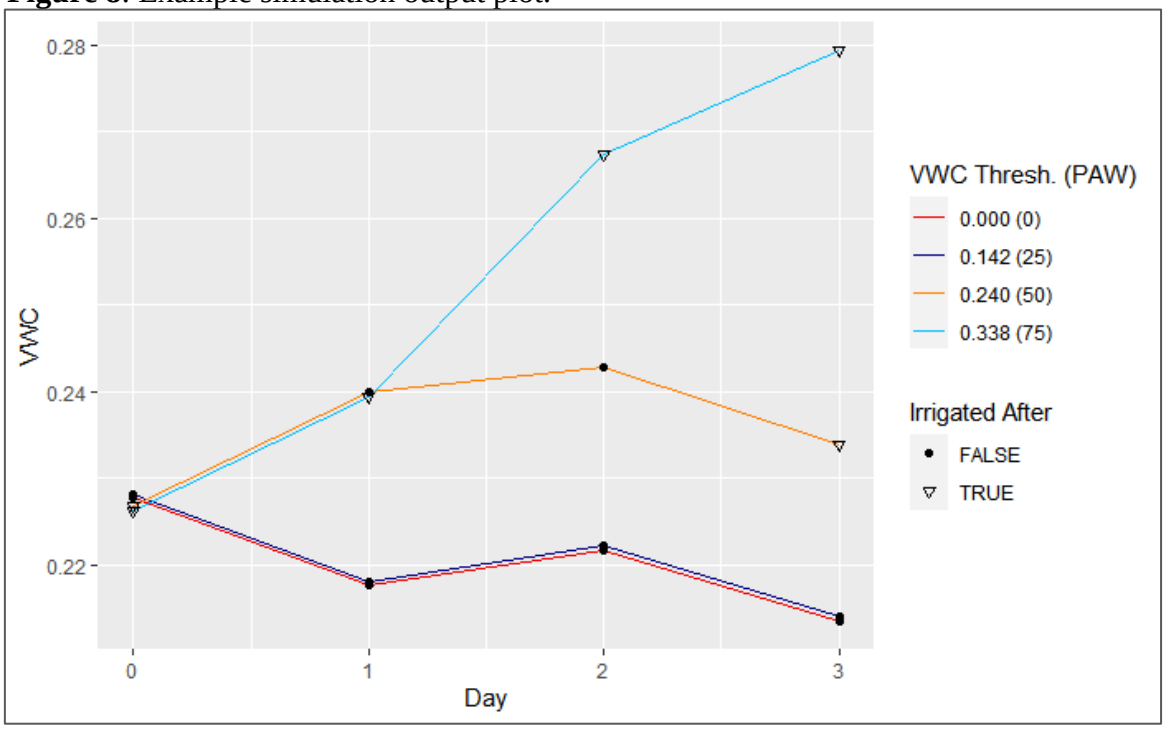
VWC Thresh (PAW)	Day 0	Day 1	Day 2	Day 3
0.00 (0)	0.227	0.217	0.221	0.213
0.103 (15)	0.227	0.217	0.221	0.213
0.122 (20)	0.227	0.217	0.221	0.213
0.142 (25)	0.227	0.217	0.221	0.213
0.162 (30)	0.227	0.217	0.221	0.213
0.181 (35)	0.227	0.217	0.221	0.213
0.201 (40)	0.227	0.217	0.221	0.213
0.221 (45)	0.227	0.217	0.248	0.239
0.24 (50)	0.227	0.24	0.243	0.234
0.26 (55)	0.227	0.24	0.268	0.258
0.279 (60)	0.227	0.24	0.268	0.28
0.299 (65)	0.227	0.24	0.268	0.28
0.319 (70)	0.227	0.24	0.268	0.28
0.338 (75)	0.227	0.24	0.268	0.28

Table 3: Example simulation output of irrigation events. (1 = True, 0 = False)

VWC Thresh (PAW)	Day 0-1	Day 1-2	Day 2-3	Day 3-4
0 (0)	0	0	0	0
0.103 (15)	0	0	0	0
0.122 (20)	0	0	0	0
0.142 (25)	0	0	0	0
0.162 (30)	0	0	0	0
0.181 (35)	0	0	0	0
0.201 (40)	0	0	0	0
0.221 (45)	0	1	0	0
0.24 (50)	1	0	0	1
0.26 (55)	1	1	0	1
0.279 (60)	1	1	1	1
0.299 (65)	1	1	1	1
0.319 (70)	1	1	1	1
0.338 (75)	1	1	1	1

Since the tables contain a lot of information to process, we chose to allow a user to plot a subset, shown in **Figure 8**. This figure displays the VWC values for four different VWC thresholds over a simulation period. Each color is a different threshold, and the symbols indicate if an irrigation event is called for. Initially, we can see lower threshold values have low VWC and less irrigation events while high thresholds may call for irrigation events multiple days in a row. For this plot, we included the threshold of 50 since that is a general industry standard. This plot is an important tool for the user because they can easily visualize what their soil moisture content will be and when they need to irrigate for each day of the simulation at their specific threshold. They could choose to plot up to six days in advance instead of three if they wanted to plan the whole week of irrigation out, and they can also select different thresholds to display on the plot.

Figure 8: Example simulation output plot.



Discussion and Conclusion

Overall, we were able to accurately predict VWC by using only water balance and weather features. As a result, we achieved our goal of keeping the model as simple as possible. The linear model gave us the best accuracy due to the interaction effects of the model fitting the data really well. The neural network and random forest still gave great accuracy, however they most likely picked up extra noise from outlier data that the linear model resisted.

The simulation program provides useful and actionable information for golf course superintendents, as it can help optimize water usage while catering to the user's VWC threshold preference. Using our simulation, it would be easy for a golf course superintendent to plan a week's worth of irrigation at their threshold preference by inputting only the forecasted ET & precipitation in addition to their current soil moisture measurements. The weather forecasts and soil moisture measurements are accessible for a user of this program, as forecasts are available to the public through the National Weather Service, and soil moisture sensors are owned by many golf courses.

We believe this program has strong potential to work well in real-world applications, though we can't say there aren't any limitations. For one, we must make a lot of assumptions about the ground we're working with. Different soil, grass, or even off-calibrated moisture sensors will completely change how our models react and predict. If we apply these exact models in a new setting, we cannot be sure that the VWC reacts the exact same way for any given rainfall, ET, or irrigation event. It would be best to train a new model specific to the location and application we are working on. Also, we can only predict so far into the future without significantly lower

accuracy. This program is best designed to schedule about a week out given that the National Weather Service only has ET forecasts up to one week. Any long-term predictions would need further research to find best parameters.

In the future, we think this project could be improved in many ways. One can always test with more data to reduce overfitting. We believe the neural network and random forest can especially be improved if they have a larger dataset to be able to compensate for outliers that easily affect the output. Extra data can also be collected in new environments. This can allow for more types of models to be tested, applied, and eventually able to generalize the VWC predictions to be accurate anywhere. We also think Time Series analysis can be used to better model uncertainty and different changes that happen over time. The current model takes each day independent of the previous or next, though there could be some long-term effects that change the end result.

A final key expansion could also be to automate the optimization between water usage and other turf features. This program focuses on predicting VWC, and then lets a user decide an irrigation threshold that best fits their application. If these simulations were combined with, say, a target turf health NDVI value, we could predict and pick an irrigation threshold value that will automatically hover the VWC to keep the grass in optimal health conditions. This would be very useful for any superintendent to simplify the decision process and still have a balance between the overall quality of the grass and minimized total water usage.

References

[1] Friell et al., *Determining Irrigation Thresholds to Optimize Water Use, Turf Health, and Playability.* USGA ID#: 2021-15-739 .

[2] Allen et al., (1998), Chapter 4 - Determination of ET_0. *FAO Irrigation and drainage paper 56.* ISBN 92-5-104219-5. www.fao.org/3/x0490e/x0490e00.htm .

[3] Comprehensive R Archive Network (CRAN). (n.d.). *Package randomforest*. CRAN. cran.r-project.org/web/packages/randomForest/ .

[4] Comprehensive R Archive Network (CRAN). (2019, February 7). *Training of neural networks [R package neuralnet version 1.44.2]*. The Comprehensive R Archive Network. cran.microsoft.com/snapshot/2022-03-24/web/packages/neuralnet/ .

[5] US Department of Commerce, N. (n.d.). National Weather Service. https://www.weather.gov/

[6] University of Kentucky. (n.d.). *UKAWC PointAgCast/Severe Weather Info*. UK Ag Weather Center. weather.uky.edu/ukawc2.php .