Precision Agric (2018) 19:421–444
https://doi.org/10.1007/s11119-017-9527-4

Applying machine learning on sensor data for irrigation recommendations: revealing the agronomist's tacit knowledge

Anat Goldstein[1] · Lior Fink[1] · Amit Meitin[1] · Shiran Bohadana[1] · Oscar Lutenberg[2] · Gilad Ravid[1]

Published online: 31 May 2017

Abstract Jojoba Israel is a world-leading producer of Jojoba products, whose orchards are covered with sensors that collect soil moisture data for monitoring plant needs at real-time. Based on these data, the company's agronomist defines a weekly irrigation plan. In addition, data on weather, irrigation, and yield are recorded from other sources (e.g. meteorological station and irrigation-plan records). However, so far, there has been no attempt to use the entire set of collected data to reveal insights and interesting relationships between different variables, such as soil, weather, irrigation characteristics, and resulting yield. By integrating and utilizing data from different sources, our research aims at using the collected data not only for monitoring and controlling the crop, but also for predicting irrigation recommendations. In particular, a dataset was constructed by integrating data collected over almost two years from 22 soil-sensors spread in four major plots (which are divided into 28 subplots and eight irrigation groups), from a meteorological station, and from actual irrigation records. Different regression and classification algorithms were applied on this dataset to develop models that were able to predict the weekly irrigation plan as recommended by the agronomist. The models were developed using eight different subsets of variables to determine which variables consistently contributed to prediction accuracy. By comparing the resulting models, it was shown that the best regression model was Gradient Boosted Regression Trees, with 93% accuracy, and the best classification model was the Boosted Tree Classifier, with 95% accuracy (on the test-set). Data that were not contributing to the model prediction success rate were identified as well. The resulting model can significantly facilitate the agronomist's irrigation planning process. In addition, the potential of applying machine learning on the company data for yield and disease prediction is discussed.

✉ Anat Goldstein
goldanat@post.bgu.ac.il

[1] Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, P.O.B 653, 84105 Beer Sheba, Israel

[2] Jojoba Israel Ltd., Hatzerim, Israel

Keywords Precision agriculture · Data mining · Internet of things (IoT) · Machine learning · Irrigation planning · Gradient boosted regression trees

Introduction

Recent developments in the fields of wireless sensor networks, satellite imagery technology, smart agricultural machinery, and ubiquitous mobile communication, together forming the *Internet of things* (IoT), have led many farms to adopt precision agriculture (PA) and become ‘smart farms’ (Castle et al. 2015; Zhang et al. 2002). Nevertheless, it seems that still, in many of these smart farms, data gathered from different sources are analyzed separately and the potential of value extraction from existing data is not fully realized (Sørensen et al. 2010). In addition, collected data are mainly used for purposes of monitoring and controlling crops, e.g., by responding to soil moisture levels or disease symptoms (Damas et al. 2001; de Miranda 2003; Evans and Bergman 2007; Yunseop et al. 2008; Vellidis et al. 2016). Against this background, the present study shows that data integrated from different sources could be also exploited for predicting agronomist irrigation recommendations.

This study was conducted in collaboration with Jojoba Israel, one of the world’s leading producers of Jojoba oils, as well as a leader in the use of sensor-based agriculture. The orchards of Jojoba Israel are covered by sensors that collect soil and plant-related data, which are used for monitoring the state of plants at real-time. Additionally, the company’s agronomist (hereinafter, the agronomist) uses these data, while also considering meteorological data collected from a meteorological station and weather forecasts, to construct a weekly irrigation plan for each plot.

A common approach for deciding on irrigation amounts is to use the Penman–Monteith equation, which approximates net evapotranspiration (ET), given daily mean temperature, wind speed, relative humidity, and solar radiation, as defined by the Food and Agriculture Organization (FAO) in the FAO-56 standard (Allen 1998). However, since all the equation parameters are measured by the meteorological station for the entire plantation (rather than for each plot), the agronomist can calculate a single ET value for all plots, which would lead to a single irrigation recommendation for all plots. The agronomist combines this ET value with other measured parameters and plot-related characteristics to determine plot-specific irrigation quantities, resulting in different irrigation quantities for different subplots. The decision process is conducted ‘manually’, i.e. it is based on no explicit irrigation model but rather on the tacit knowledge of the agronomist, who examines the plant and soil conditions, as reflected in the sensor-based data, and decides accordingly. In fact, there is no recommended model for Jojoba irrigation in the literature and the irrigation decision process in Jojoba Israel has changed significantly over the years, resulting in a steady increase of yield.

The goal of this study was to develop a model that captures the unstructured decision process of the agronomist. This objective was achieved by applying machine learning (ML) on an integrated dataset, comprising sensor-based moisture data, meteorological data, and actual irrigation plans defined by the agronomist in the past, in order to learn his irrigation decision-making process.

Various models for irrigation decision support exist in the literature. These models are used for different purposes, such as optimizing the allocation of water under different constraints (e.g. Lilburne et al. 1998), determining irrigation needs (e.g. Thysen and Detlefsen 2006; Jensen et al. 2000), scheduling (e.g. Vellidis et al. 2013), and recording water consumption (e.g., Mateos et al. 2002). Only few of the irrigation decision-support models use ML. For example, Liu et al. (2006) used a genetic algorithm to optimize water allocation for a single crop in different growing periods and for numerous types of crops, given the water cost, market price of the crops, additional costs of irrigation, crop expenses, and expected yield reduction because of under-irrigation or over-irrigation in the entire growing period. The resulting optimization models were incorporated in a decision support system (DSS) for precision irrigation. These models, however, did not account for plot-specific characteristics. The literature also includes models for determining irrigation needs based on plot-specific data. For example, Hedley et al. (2013) used ML to predict soil water status and water table depth based on electromagnetic mapping. Smith and Peng (2009) used ML to classify soil textural composition based on soil-sensor data to inform a deficit irrigation control system. Another approach to infer plant water status is based on infographics. For example, Murase et al. (1995) used a layered neural network to identify plant water status based on the textural features of the pictorial information of the plant canopy.

In many of the existing models (whether they use ML or not), decision support is based on providing the farmer with simple water balance models (e.g. based on the Penman–Monteith equation) that are subjected to capacity constraints and allowing the farmer to simulate different irrigation alternatives and assess their implications. The support of these models is also based on displaying the farmer information on plant and soil status that is required for irrigation decision making.

A different approach, found in the literature, is to model the actual decision-making process of the farmer. For example, Labbé et al. (2000) modeled the irrigation decision making process of farmers for limited water allocation and irrigation scheduling for corn. The model consisted of irrigation management rules for different irrigation-related tasks that were derived from surveys of farmers and based on monitoring their irrigation practices during a period of 2 years. This model was incorporated in a simulator engine that, given the decision context, was able to predict irrigation schedules and irrigation volumes with average error ranging between 6 and 13 mm for different farmers, reflecting an error below 6.7%. Instead of manually developing a model that captures the farmer's decision process using surveys and observations, in this study, ML was used to capture the agronomist's irrigation decision process. The potential of using ML to forecast future farmer decisions was identified by Andriyas and McKee (2013). In particular, they tried to model the farmer's decision of whether or not to irrigate for three crops: alfalfa, barley, and corn. Whereas their work focused on whether to irrigate or not, our work is aimed at predicting the required irrigation quantities.

As mentioned above, a model that captures the unstructured decision process of the agronomist was developed in this study. To find the best model, various models, developed using different ML techniques and different subsets of the independent variables, were compared. Thus, the model can automatically recommend an irrigation plan (the dependent variable) and can be therefore used to support irrigation decisions and increase the efficiency of the irrigation planning process.

Achieving the objective of this study would potentially contribute to the literature in three ways. First, the study offers an irrigation recommendation model that is able to automatically recommend irrigation quantities similar to the expert agronomist, while accounting for plot-specific data. Automating the irrigation planning process of the agronomist can reduce his workload and increase his efficiency. Thus, the irrigation planning process can be effortlessly replicated to additional orchards or be carried out more often. In addition, the agronomist can use the model to fine-tune and validate his irrigation decisions. Second, the study may provide insights into the variables that contribute to the irrigation recommendation prediction model and should, therefore, be collected and accounted for in similar analyses. Third, the study may provide insights into the ML techniques that are most suitable for constructing irrigation recommendation models.

The rest of the paper is organized as follows. In the following section, the data sources and research method employed to develop the prediction models are presented. In addition, background on the relevant ML methods is provided. Then, the resulting models are presented, followed by a discussion of the results and comparison of the models. Finally, conclusions are provided and opportunities for developing other prediction models (for disease and yield prediction) based on an integrated dataset are discussed.

Materials and methods

The following subsections describe the data sources analyzed in this study, followed by the various steps comprising the research method.

The study followed a typical data mining process (Fayyad 1996), which included the stages of dataset definition, extraction-transformation-loading (ETL), application of ML algorithms to the dataset to train recommendation models, evaluation of the trained models, and selection of the best prediction model. Figure 1 presents the steps of the research method, which are elaborated in the following subsections.

Data sources

Data from various sources were integrated in this study. Soil moisture data were collected from uMANAGE™[1]—a real-time, web-based crop management system, connected to different types of sensors. As shown in Fig. 2, the study focused on four major plots, in which data had been collected since 2014. Each plot was characterized by the planting year (age), the type of planting (seeds or clones), whether the planting was organic or not, and the soil type. Each plot was further divided into 4–10 subplots that were separately irrigated. There were eight groups of subplots that were irrigated together, and each subplot was part of a specific group (the borders of each irrigation group are highlighted). Each subplot was also associated with one of eleven transmission stations, which transmitted data from two moisture sensors, located at two depths, shallow (less than 0.4 m underground) and deep (more than 0.4 m underground). In total, data were collected from 28 subplots of land, covered by 22 moisture sensors.

Apart from sensor-based moisture data, meteorological data were also gathered from a nearby meteorological station (marked by a blue star in Fig. 2), namely, data on air temperature, solar radiation, and air humidity. The plots were all within 1–2 km away from the meteorological station. Forecast archives were not available to provide the forecasting data available at the time of decision-making; instead, weekly average forecasts of temperature, humidity, and radiation were estimated by averaging the actual temperature, radiation, and humidity measured in the following week. Using actual data was considered

[1] https://www.netafim.com/product-category/umanage.

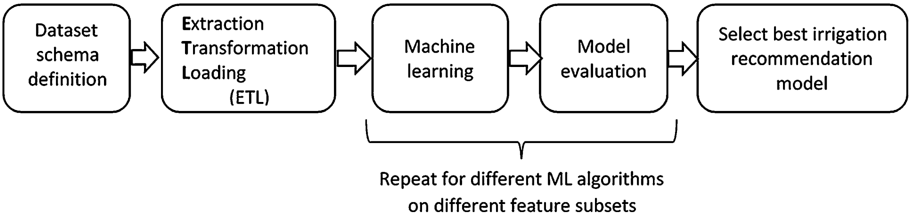
Dataset schema definition
Extraction Transformation Loading (ETL)
Machine learning
Model evaluation
Select best irrigation recommendation model
Repeat for different ML algorithms on different feature subsets

Fig. 1 Research process

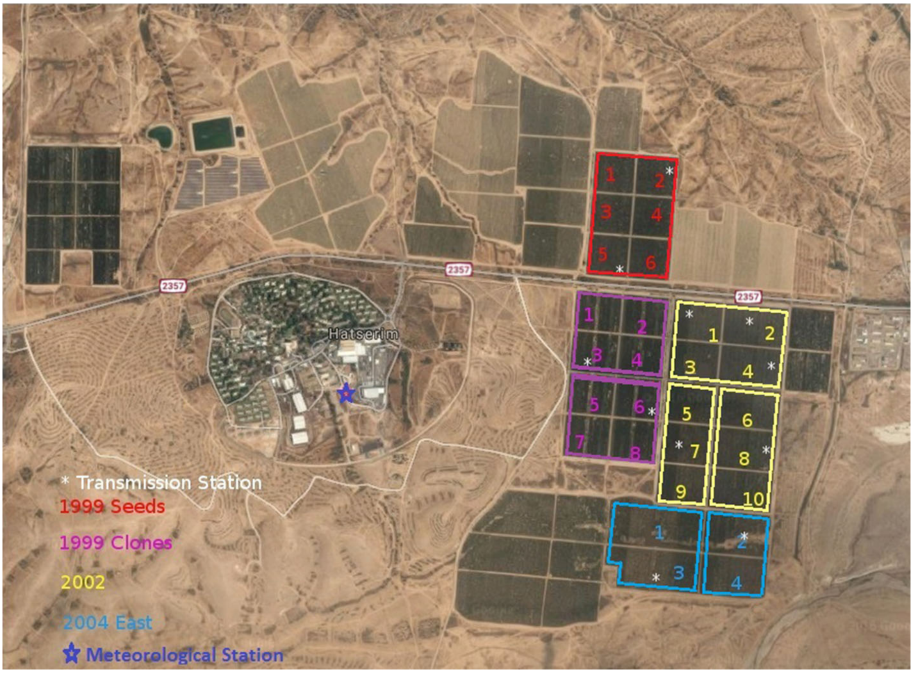

Fig. 2 Jojoba Israel - plots divided into irrigation groups and subplots, transmission stations (each with two sensors), and meteorological station

satisfactory for our purpose because weather forecasts in this area (the city of Be’er-Sheva) are considered accurate (e.g. the standard error in temperature forecasts is 1 °C, according to the Israel Meteorological Service). Weekly irrigation plans, created by the agronomist and recorded in excel sheets, were also used. The dataset contained data that were gathered from March 2014 until the end of 2015.

Dataset definition

In this step, the structure of the dataset for model training and testing was defined. The dataset combined data from the sources listed above. The variables characterizing the dataset observations, commonly referred to as ‘observation features’, are presented in Table 1. Features that were gathered from uMANAGE (every 15 min) and from the

Table 1 Dataset features

Variable	Description and time resolution	Time resolution of raw data	Relevant spatial unit	Source
Temperature	Average over the week preceding irrigation planning	Every 20 min	The entire land area	Meteorol. station
Solar radiation				
Relative air humidity				
Temperature forecast (Prediction_Avg_of_Air_Temperature)	Average weekly forecasts, estimated by averaging the actual values measured over the week following irrigation planning	Daily		
Solar radiation forecast (Prediction_Avg_of_Radiation)				
Air humidity forecast (Prediction_Avg_of_Humidity)				
Temperature IPPD[a] (Temp_IPPD)	Average of the values measured during the day preceding irrigation planning (Irrigation planning preceding day—IPPD)	Every 20 min		
Solar radiation IPPD (Radiation_IPPD)				
Air humidity IPPD (Hum_IPPD)				
Organic (Organic_plot)	Boolean: 1—organic 0—not organic		Plot (1-4)	Land data, maintained in excel files
Land name (Plot)	Categorical: 1999 seeds 1999 clones 2002 2004 east			
Irrigation group (Sub_plot)	Categorical: A, B, C, D, E, F, G, H		Subplots related to an irrigation group (1–8)	
Total drought-stress duration at depth d[b] (shallow_state_1, deep_state_1)	Duration (in days) in which the soil moisture was lower than 20% at depth d in the previous week	Every 15 min	Subplots related to a transmission station (1–11)	uMANAGE
Total saturation duration at depth d[b] (Shallow_state_2, deep_state_2)	Duration (in days) in which the soil moisture was higher than 27% at depth d in the previous week			
Soil moisture at depth d[b] (Moisture_sensor_deep, moisture_sensor_shallow)	Average soil moisture at depth d			

Table 1 continued

Variable	Description and time resolution	Time resolution of raw data	Relevant spatial unit	Source
Planned irrigation (millimeter) (Watering)	The dependent variable. Every week the agronomist plans irrigation volumes per plot	Once a week	Subplots related to an irrigation group (1–8)	Irrigation excel sheets

[a] *IPPD* irrigation planning preceding day

[b] *d* sensor depth, either shallow (less than 0.4 m underground) or deep (deeper than 0.4 m underground)

meteorological station (every 20 min) were averaged or summed to represent a granularity of one week.

Data extraction, transformation, and loading

After the dataset structure was defined, the relevant data were processed following the ETL process—they were extracted from their sources, cleaned and transformed into the required features, and finally were loaded to a flat table. This ETL process is summarized next.

Sensor data

When sensors are used, noisy data is often a problem. To clean the data, following guidelines provided by the agronomist, moisture values smaller than 17 or higher than 39%, considered abnormal, were filtered out. In the complete dataset, such anomalies, likely to be caused by a faulty or uncalibrated sensor or transmitter, occurred only twice. In these two instances, the values were replaced by using spatial mean imputation (Kennedy and Tobler 1983), i.e. by the average of the values gathered from the two nearest sensors.

Weekly soil moisture averages were then calculated for each irrigation group. Moisture sensors were located in two depths: deep (deeper than 0.4 m) and shallow (less than 0.4 m deep). Thus, two weekly moisture averages were calculated—one for each depth. In addition, for each irrigation group at each depth, the duration (in days), in which the soil was saturated (moisture level was higher than 27%, as determined by the agronomist) and drought-stressed (moisture level was lower than 20%, as determined by the agronomist) over the preceding week were calculated. These durations were represented as integers (1–7 days).

The reason for aggregating data on a weekly basis was that the irrigation planning process was carried out once a week, based on data collected over the previous week.

Meteorological data

Average air temperature, humidity, and radiation over the preceding week and during the preceding day were calculated based on data that were gathered from the nearby meteorological station (Fig. 2), which measured these values every 20 min. Averages were calculated over the entire day. Additionally, these data were used to estimate the weekly average forecasts of temperature, humidity, and radiation by averaging the actual temperature, radiation, and humidity that were measured in the week for which the irrigation plan was prepared.

Other attributes

Each row of the dataset captured the above week-level data for a particular irrigation group (there were eight groups) in a particular week and year. Each row also included the characteristics of the irrigation group: the plot it belonged to (determining the soil type), whether it was organic or not, and the actual irrigation quantity in millimeters (mm), which served as the dependent variable.

The resulting dataset included 695 records, representing a period of 98 weeks. This dataset was then segmented into training and validation sets: 70% of the records (randomly selected) were defined as the training-set and used to train the models, and the remaining records (30%) were defined as the test-set and used to evaluate the performance of the model.

Machine learning methods

The field of machine learning (ML) is concerned with the question of how to construct computer programs that automatically improve with experience (Mitchell 1997). A primary goal of ML is the construction of algorithms that can learn from data and make predictions using data. Usually, these algorithms use a large set of input observations, called a training-set, in order to tune the parameters of an adaptive model (Bishop 2006, p. 2).

ML algorithms aim at various tasks, including Regression—a process for estimating the relationships between a dependent variable and one or more independent variables; Classification—a process of identifying to which category, out of a given set of categories (classes), an observation belongs to, based on a training-set of previously classified observations; Clustering—a process of dividing a set of observations into a number of groups (clusters), such that observations grouped in the same cluster are more similar to each other than to those in other clusters (according to a given similarity function); and Association rule learning—a statistical process of identifying strong interesting relationships (rules) between variables based on large sets of observations (Liu 2011).

In the present study, regression and classification models were developed to predict the irrigation volumes recommended by the agronomist (in the case of classification models, observations were classified into five irrigation level classes). In the following subsections, these types of models are presented.

Regression models

A regression task can be defined as follows: Given a training set comprising N observations of an input variable X, written $X \equiv (X_1,\ldots, X_N)^T$, together with corresponding observations of the values of a target variable $t \equiv (t_1,\ldots, t_N)^T$, the goal is to exploit the training set in order to make predictions of the value t’ of the target variable for some new value X’ of the input variable. X is in fact a vector of D features $(x^1,\ldots,x^D)$ characterizing each observation.

A simple approach for handling this task is to use linear regression (Mitchell 1997, p. 237; Bishop 2006, p. 5), where a function $y(X, W) = w_0 + w_1x^1 + w_2x^2 + \ldots + w_Dx^D = \sum_{i=1}^{D} w_i x^i$ is fitted to the training set, by finding the W that minimizes a loss function that measures the misfit between the function y(X,W) and the training set data points. The root-mean-squared error (RSME) is commonly used to measure the misfit. The advantage of the linear regression model is that it can be used to explain the effect of different factors (feature values) on the target (dependent) variable.

A more advanced approach to regression is regression trees. Regression trees models (Loh 2011), similar to linear regression, are used to predict the value of the dependent variable, given a new data sample. However, different from linear regression, regression trees algorithms do not try to capture a linear relationship between the model features and the target variable. Instead, they are based on decision trees. A decision tree is a tree-graph, in which each internal node (i.e. a vertex with outgoing edges) is labeled with an input feature and the edges coming from the node are labeled with each of the possible values of that feature. Observations are sorted according to their feature values from the root of the tree downwards to the leaves. While in decision tree classifiers, each leaf (i.e. a vertex without outgoing edges) is labeled with a name of a class to which observations may belong, in regression trees, each leaf is labeled with a target value, which is the average of target-variable values of observations assigned to that leaf. For additional information on regression tree see (Loh 2011; Bishop 2006, p. 663).

In this study, a regression tree model, called *gradient boosted regression trees (GBRT)*, is used. The GBRT is a type of additive model that makes predictions by combining decisions from a sequence of base regression tree models (Elith et al. 2008). Unlike *random forest* (Breiman 2001), a ML approach that constructs a multitude of decision trees independently, each using a subsample of data, and returns the mean prediction of the individual trees, GBRT uses a particular model ensemble technique called gradient boosting, which iteratively builds a model, while improving the performance of the previous iteration model. This model ensemble can be written formally as $g(x) = f_0(x) + f_1(x) + f_2(x) + \ldots$, where regression tree $g(x)$ is a sum of base regression trees f_i.

The name Gradient Boosting comes from the association of this method with Gradient Descent optimization (Elith et al. 2008), commonly used to solve linear regression problems by finding a local minimum of the linear regression loss function (Bishop 2006). Similarly, let $g_t(x) = \sum_{i=0}^{t-1} f_i(x) = g_{t-1}(x) + f_t(x)$ be the regression tree trained at iteration t, $L[y_i, g(x_i)]$ be the loss function, and N be the number of observations, at each gradient boosting iteration, the algorithm finds a regression tree f_t, which moves g_t towards the negative gradient direction $-\partial L/\partial g$ by η (step-size) amount. Hence, f_t is chosen to be $f_t = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{N} \left\{ \frac{\partial L[y_i, g(x_i)]}{\partial g(x_i)} - f(x_i) \right\}^2$ and the algorithm sets $g_{t+1} = g_t + \eta f_t$. Since for regression problems with sum-squared loss function $\partial L/\partial g = y_i - g(x_i)$, f_t can be written as follows:

$$f_t = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{N} [y_i - g(x_i) - f(x_i)]^2.$$

GBRT is considered as one of the most effective and popular models for prediction (Cai et al. 2016). However, as opposed to linear regression or simple decision trees, which are easy to interpret, GBRT-based regression trees are more difficult to interpret, being a collection of many regression trees.

Classification models

Unlike regression tasks, in which the output variable is continuous, the output of classification tasks is one of a finite number of discrete categories. A classification task can be defined as follows: Given a training set comprising N observations of variable X, written $X \equiv (X_1,\ldots, X_N)^T$, where Xi is a vector of D features $(x^1,\ldots,x^D)$, together with corresponding observations of the values of a target variable $t \equiv (t_1,\ldots, t_N)^T$, where t_i may take one of $|C|$ discrete values $[C^1, C^2,.., C^{|C|}]$, our goal is to use the training set to train a model that correctly predicts the class of some new observation X'. There are many types of classification models, including decision trees (Mitchell 1997, pp. 52–59), support vector machines (SVM) (Liu 2011, pp. 109–116), logistic regression (Mitchell 1997, pp. 205–207), and Bayesian procedures (Mitchell 1997, pp. 154–164).

In this study, a particular classification algorithm, called boosted trees classifiers (BTC), is used. This algorithm is an extension of GBRT, discussed above, to classification problems (Mease et al. 2007). That is, instead of iteratively improving a regression tree model, a decision tree classifier is iteratively improved, by combining base-decision tree classifiers using the gradient boosting technique.

Development of ML models

Platform

For the data analysis and model development, *graphLab create* of the *Turi*[2] *machine learning platform* was used. GraphLab create is a python-based ML framework, which enables the development and deployment of intelligent applications and services. It includes libraries for data transformation and manipulation, as well as scalable ML toolkits for creating, evaluating, and visualizing ML models. It is built to handle large volumes of data (terabytes) efficiently and quickly.

Selection of ML models

For the development of the irrigation recommendation prediction model, different types of ML algorithms were examined: linear regression, regression trees, and classification algorithms. To select the particular algorithms for use in this study, a basic comparison process was carried out: First the integrated dataset was divided into training (70%) and test (30%) sets; Second, different ML algorithms were applied (with their default settings) on the training set to train different ML models; Finally, the performance of the resulting models were examined on the test-set and compared to one another.

For regression trees, the following models were considered: Decision Tree Regression, Random Forest Regression, and GBRT. The best model among these, in terms of RSME,

[2] https://turi.com.

was GBRT. Linear regression was selected as well due to its high interpretability and prevalence in the literature. For classification, the following classification models were compared: logistic classifier, decision tree classifier, random forest cassifier, and BTC. The best classier among these, in terms of classification accuracy, was BTC.

Thus, three ML models were developed: Linear regression, BTC, and GBRT. Each of these models was developed on eight different subsets of the feature-set in order to find the best model. Using the entire set of features to develop a single model may lead to overfitting (Mitchell 1997, p. 67) and to models that are more difficult to explain. In addition, because of correlations among many of the features, it was necessary to search for the best subset of features, as described in the following section.

Feature selection

In ML, feature selection is the process of selecting a subset of the relevant features (i.e. the variables listed in Table 1) for use in model construction. For feature selection, the correlation between each pair of variables was first estimated. As could be expected, correlations among the different meteorological features were very high (for example, weekly averages and preceding-day averages were correlated for each of the measures, with correlation values exceeding 0.85), as were the correlations among sensor-related features (e.g. the correlations between saturation/drought durations and soil moisture levels were higher than 0.7). By contrast, the correlations between sensor- and meteorology-related features were relatively low (lower than 0.1). Thus, three groups of optional variables were defined: (1) IPPD weather (including average temperature, radiation, and humidity measured on the day preceding irrigation planning); (2) weather forecast (predicted temperature, radiation, and humidity for the following week); and (3) saturation/drought durations over the preceding week. Because each of these three variable groups could be either included or not in the feature set, eight (2^3) feature subsets were defined, as shown in Table 2. Of the variables listed in Table 1, those that are not listed in Table 2 were included in all subsets. Each of the three ML models mentioned above was developed for each of the eight feature-subsets.

Model evaluation measurements

In the next section, the resulting models are presented and their ability to predict irrigation quantities (i.e. their ability to recommend a suitable irrigation plan) is evaluated. For the evaluation of regression models (linear regression and GBRT), successful predictions were defined as those in which the predicted value was not more than 0.2 mm below or above the actual value. To use classification (BTC), the dependent variable has to be categorical. The continuous irrigation variable was therefore binned into five classes, described in Table 3, which best captured the variance in this variable. These classes were then used as the predicted variable in BTC models. Successful prediction were defined as those in which classification was made to the correct irrigation class.

Table 2 Feature sets for which ML models were developed

Feature set	Drought (state_1)/ saturation (state_2) duration	Weather forecast (humidity, solar radiation, temperature)	Weather measured on the preceding day (humidity, solar radiation, temperature)
Feature names	shallow_state_1, shallow_state_2, deep_state_1, deep_state_2	Prediction_Avg_of_Humidity, Prediction_Avg_of_Radiation, Prediction_Avg_of_Air_Temperature	Temp_IPPD, Hum_IPPD, Radiation_IPPD
1	✓	✓	✓
2	✓	✓	
3	✓		✓
4	✓		
5		✓	✓
6		✓	
7			✓
8			

Table 3 Irrigation classes

Class	Irrigation (mm)
1	Less than 1.5
2	1.5–2.0
3	2.0–2.5
4	2.5–3.0
5	More than 3.0

Results

Linear regression

Linear regression was used with the eight datasets defined in Table 2. Each dataset was randomly divided into training and test sets, such that the training set contained 70% of the samples. Table 4 summarizes the performance of the eight linear regression models, showing for each the RSME, maximal error, maximal deviation rate (error/true value), and rate of successful predictions (those within a 0.2 mm deviation), in total and for each plot.

As can be seen in Table 4, Set 2 achieved the best results in terms of RSME (although RSME values for other models were quite similar), and Set 6 achieved the highest overall success rate (although a value of 52.3% success is considered relatively low). Table A1 (Appendix) presents the learned coefficients of the model developed for Set 2, which had the lowest RSME.

Due to the relatively low success rates of the simple linear regression models, regression models that aimed at capturing non-linear relationships between the independent variables and the dependent variable were also developed. For example, when plotting irrigation amounts against the temperature measured on the day preceding irrigation planning (Fig. 3), it can be seen that a square root function ($\sqrt{x}$) may better describe the relationship between irrigation and IPPD temperature. As another example, when plotting irrigation

Table 4 Performance of the simple linear regression models for the different datasets

Measure	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8
RSME	0.467	**0.466**	0.479	0.485	0.469	0.469	0.481	0.487
Maximal error	2.362	2.355	2.372	2.400	2.331	**2.320**	2.340	2.340
Maximal deviation rate	**131.2%**	134.1%	166.8%	179.0%	136.8%	140.1%	172.7%	187.8%
Success rate—total	49.5%	50.5%	48.2%	45.0%	50.9%	**52.3%**	50.9%	45.9%
Success rate—1999 seeds	46.4%	42.9%	28.6%	28.6%	**50.0%**	42.9%	35.7%	28.6%
Success rate—1999 clones	49.2%	50.8%	45.8%	49.2%	52.5%	**54.2%**	49.2%	49.2%
Success rate—2002	44.0%	45.3%	44.0%	40.0%	42.7%	46.7%	**48.0%**	45.3%
Success rate—2004 east	58.9%	60.7%	**66.1%**	55.4%	60.7%	62.5%	64.3%	51.8%

The best values for each measure are highlighted in bold

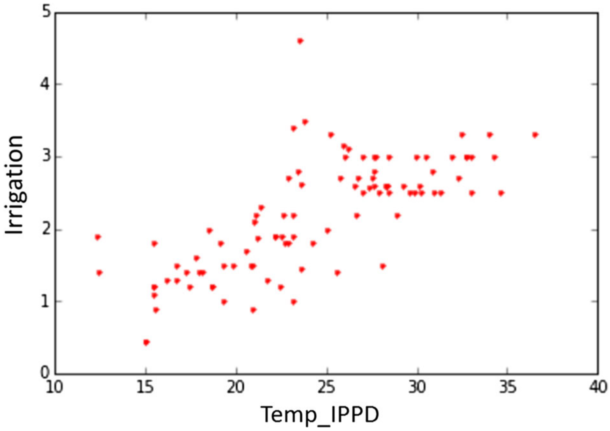

Fig. 3 Irrigation against temperature measured on IPPD

amounts against the week number, as shown in Fig. 4, it is easy to see that a quadratic function (x^2) would be more suitable to describe the relationship.

Therefore, for each numeric feature *x* in the dataset, four new features were added: $\sqrt{x}$, x^2, x^3, and x^4. The corresponding eight feature subsets were defined as before, but each subset was extended with the polynomial features. The linear regression algorithm was applied on each subset, resulting in eight new polynomial regression models. The performance measures of these models are summarized in Table 5.

Adding polynomial features improved the ability of the models to predict irrigation levels. The model for Set 5 (no drought/saturation data) achieved the lowest RSME value (0.38), which was superior to the lowest RSME value achieved without polynomial features (0.47 for Set 2). The highest total success rate with polynomial features was 62.8%, compared to 52.3% without them, and the smallest maximal error without polynomial features was 2.02, compared to 2.32 without them. While adding polynomial features improved the prediction ability of the models to some extent, it had the negative

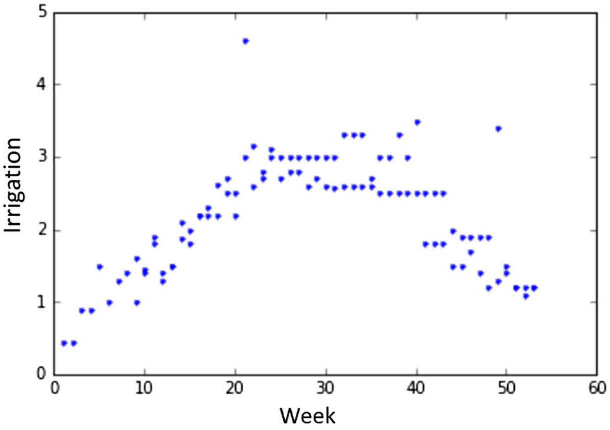

Fig. 4 Irrigation against week number

Table 5 Performance measures of the polynomial regression models

Measure	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8
RSME	0.380	0.406	0.394	0.413	**0.376**	0.402	0.393	0.409
Maximal error	2.102	2.191	**2.021**	2.148	2.113	2.198	2.065	2.158
Maximal deviation rate	54.0%	**48.6%**	64.8%	60.2%	58.9%	51.5%	58.8%	58.1%
Success rate—total	60.6%	59.6%	56.0%	53.7%	**62.8%**	61.0%	60.1%	55.0%
Success rate—1999 seeds	57.1%	57.1%	50.0%	46.4%	**67.9%**	64.3%	57.1%	42.9%
Success rate—1999 clones	**69.5%**	64.4%	64.4%	61.0%	66.1%	59.3%	67.8%	59.3%
Success rate—2002	56.0%	53.3%	50.7%	49.3%	**60.0%**	58.7%	56.0%	54.7%
Success rate—2004 east	58.9%	**64.3%**	57.1%	55.4%	60.7%	**64.3%**	58.9%	57.1%

The best values for each measure are highlighted in bold

implication of increasing the complexity of the model due to the addition of features (e.g. the model for Set 5 had 63 features).

LASSO (least absolute shrinkage and selection operator), a method that performs both feature selection and regularization to enhance both the prediction accuracy and interpretability of the regression model (Zou 2006), was employed to eliminate variables. Its use, however, did not improve the performance of any of the models in terms of RSME and success rate.

Gradient boosted regression trees

Eight GBRT models were trained on the eight feature-sets (Set1–Set8), defined in Table 2. Each dataset was randomly divided into training and test sets, such that the training set contained 70% of the samples. Each model comprised a sequence of decision tree models. An example of one such tree appears in Fig. 5, which depicts one sub-tree of a collection of 50 trees that were learned for Set 2. The performance measures of GBRT models are summarized in Table 6.

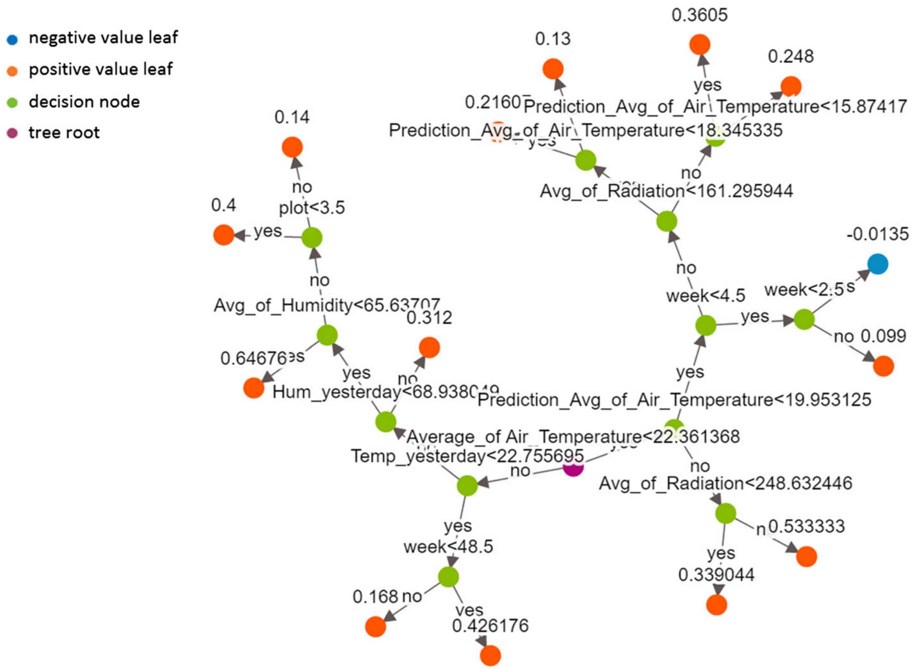

Fig. 5 An exemplary decision tree model, which was part of a GBRT model

Table 6 Performance measures of GBRT models

Measure	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8
RSME	**0.113**	0.150	0.139	0.165	0.116	0.138	0.149	0.129
Maximal error	**0.645**	0.768	0.765	0.985	0.675	0.673	0.870	0.693
Maximal deviation rate	21.5%	26.1%	**21.4%**	26.0%	22.5%	21.7%	31.3%	23.2%
Success rate—total	**92.7%**	88.1%	90.8%	91.3%	92.2%	**92.7%**	90.8%	90.4%
Success rate—1999 seeds	85.7%	82.1%	**89.3%**	82.1%	**89.3%**	85.7%	85.7%	85.7%
Success rate—1999 clones	**93.2%**	89.8%	89.8%	**93.2%**	89.8%	**93.2%**	89.8%	91.5%
Success rate—2002	**96.0%**	92.0%	94.7%	94.7%	94.7%	92.0%	94.7%	92.0%
Success rate—2004 east	91.1%	83.9%	87.5%	89.3%	92.9%	**96.4%**	89.3%	89.3%

The best values for each measure are highlighted in bold

Compared to the linear regression models, GBRT models achieved considerably better results: the model for Set 1 achieved the lowest RSME (0.11), and the models for Set 1 and Set 5 achieved the best success rate (92.7%). In other words, GBRT models were able predict the correct irrigation levels in 92.7% of the cases. The model for Set 1 also achieved the lowest maximal error, 0.64, with a maximal deviation rate of 21.5%. Set 2, which included no previous-day weather features, and Set 4, which included no weather forecast features, had the lowest success rates.

Boosted trees classifier

The BTC, provided by GraphLab Create, was used with the different feature sets. Again, the training set included 70% of the samples, which were randomly selected. Each of the resulting models comprised a collection of trees that were learned on the basis of the corresponding set of features. Five classes were defined, corresponding to the five irrigation classes in Table 3.

Table 7 summarizes the performance measures of the different classification models. The table includes measures that are different from those of the regression models: *accuracy*—the number of correct predictions out of the total number of predictions; *precision*—the average percentage of correctly classified items for all classes; *recall*—the average percentage of items classified as x out of the actual x items; *F1-Score*—a score that combines recall and precision, such that $F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$; and the precision and recall of each class separately (rate of correctly classified items).

The results show that the model for Set 1 (based on the entire feature-set) outperformed the other models in almost all performance measures.

Table 8 presents the performance of the classifier for this set, by showing the distribution of predicted labels against actual labels (confusion matrix).

Figure 6 depicts the average success rates of GBRT and BTC models in each week. In can be seen in this figure that between weeks 34 and 40, there was a period of five weeks in which the success rate dropped for both models. The agronomist confirmed that the land was deliberately and forcefully dried during that period, because the Jojoba fruits were being collected around that time by combines that were unable to operate efficiently unless the soil was dry.

The success rate values of linear regression (polynomial), GBRT and BTC models for each feature-set (Set 1–Set 8) are summarized in Fig. 7. The best success rates among the BTC models (95% for Set 1) and among the GBRT models (92.7% for both Set 1 and Set 6) were significantly higher than the best success rate among the linear regression models (62.8% for 'Set 5').

Discussion

Comparing the results for the three ML techniques reveals that the non-parametric techniques, GBRT and BTC, achieve significantly better performance measures than linear regression, even after the inclusion of polynomial features as predicting variables. While it may be more difficult to interpret the non-parametric techniques, they are found in this study to be better for predicting irrigation.

In general, tree-based models perform better than linear regression models when the relationship between the independent variables and the dependent variable is non-linear and non-monotonic. Therefore, it can be assumed that GBRT models would outperform linear regression models in predicting the recommended irrigation amounts of other crops, for which these amounts are affected by similar independent variables.

These results conform to previous results by Hedley et al. (2013), who showed that random forest models (similar to GBRT and BTC, these are ensembles of decision trees) achieved superior predictions of soil moisture and water table depth than linear regression models. The results received in this study also conform to results by Andriyas and McKee (2013), who used different tree-based models for predicting a farmer's decision of whether

Table 7 Performance measures of BTC models

Measure	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8
Accuracy	**95.0%**	94.0%	94.5%	93.6%	93.6%	94.5%	94.5%	93.6%
Accuracy—99 seeds	**89.3%**	**89.3%**	**89.3%**	**89.3%**	**89.3%**	**89.3%**	**89.3%**	**89.3%**
Accuracy—99 clones	**100%**	98.3%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
Accuracy—2002	96.0%	**97.3%**	**97.3%**	96.0%	96.0%	96.0%	96.0%	96.0%
Accuracy—2004 east	**91.1%**	87.5%	87.5%	85.7%	85.7%	89.3%	89.3%	85.7%
Precision macro	**95.3%**	94.5%	95.0%	94.0%	94.1%	94.9%	95.0%	94.1%
Recall macro	**95.4%**	94.5%	94.8%	93.8%	94.3%	95.1%	94.8%	93.8%
F1-score	**95.2%**	94.4%	94.8%	93.8%	94.0%	94.9%	94.8%	93.9%
Precision—class 1	**100%**	**100%**	**100%**	98.1%	**100%**	**100%**	**100%**	96.3%
Precision—class 2	**98.2%**	93.0%	97.6%	90.7%	93.0%	93.2%	97.6%	92.7%
Precision—class 3	85.1%	84.4%	82.6%	86.0%	81.3%	84.8%	82.6%	**86.4%**
Precision—class 4	93.0%	94.8%	94.8%	94.9%	96.4%	**96.4%**	94.8%	94.9%
Precision—class 5	**100%**	**100%**	**100%**	**100%**	**100%**	**100.0%**	**100%**	**100%**
Recall—class 1	98.1%	98.1%	**100%**	98.1%	98.1%	98.1%	**100%**	98.1%
Recall—class 2	97.6%	97.6%	97.6%	95.1%	97.6%	**100%**	97.6%	92.7%
Recall—class 3	**97.6%**	92.7%	92.7%	90.2%	95.1%	95.1%	92.7%	92.7%
Recall—class 4	88.7%	88.7%	88.7%	**90.3%**	85.5%	87.1%	88.7%	**90.3%**
Recall—class 5	**95.2%**	**95.2%**	**95.2%**	**95.2%**	**95.2%**	**95.2%**	**95.2%**	**95.2%**

The best values for each measure are highlighted in bold

Table 8 Confusion matrix of the model for Set 1

Predicted Target label	1	2	3	4	5
1	52	1	0	0	0
2	0	40	1	0	0
3	0	0	40	1	0
4	0	2	5	55	0
5	0	0	1	0	20

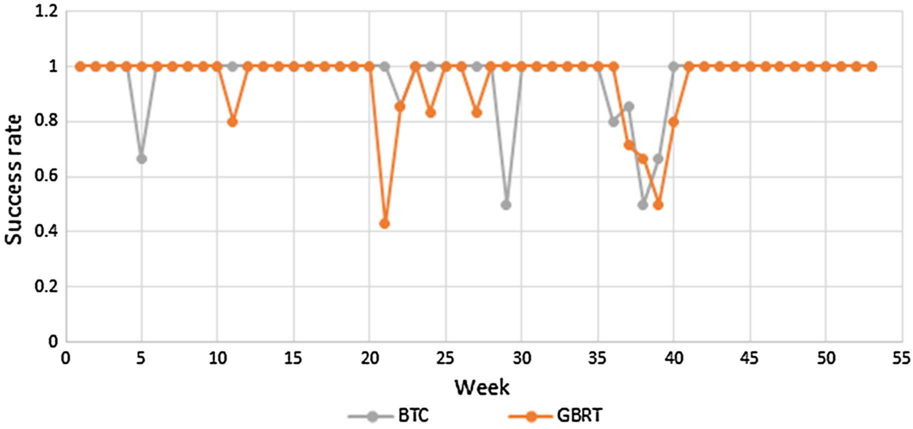

Fig. 6 Average success rate of GBRT and BTC models per week

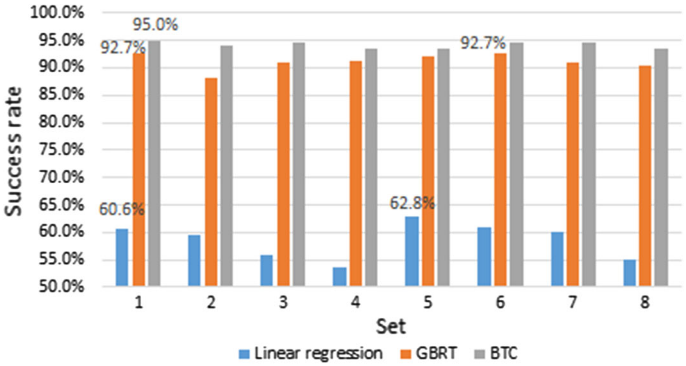

Fig. 7 Success rate of different models

to irrigate or not, and showed that tree-based models were suitable for prediction farmer's decisions of whether to irrigate or not with above 97% success rates for all decision tree models used.

While BTC models achieve slightly better results than GBRT models (95 compared to 92.7%, respectively), GBRT models are recommended for predicting irrigation amounts because of their ability to provide accurate values (BTC models can only classify into classes).

Reflecting on the results of GBRT models, it is interesting to see that while Set 1 achieved the best results in general (minimal RSME of 0.113, maximal error of 0.645, and success rate of 92.7%), Set 5, which included no saturation/drought data, achieved almost as good results (RSME of 0.116 and maximal error of 0.675). The same is true for the polynomial linear regression, where Set 5 achieved the best RSME. These findings raise questions about the importance of incorporating drought/saturation data in the model. When examining success rates, it is also interesting to see that Set 6, with included neither drought/saturation data nor IPPD weather data, achieved success rates that were similar to those of Set 1 (92.7%).

Questions on the importance of the different variables become even more interesting when considering the results for particular plots. For example, it can be seen in Table 6 that for plot '2004 east', which is organic, Set 6 achieved the best prediction success rate. Set 1 achieved the best results for plot '2002', and Sets 3 and 5 achieved the best results for plot '1999 seeds'. These findings imply that a contingency approach, by which different feature-sets are used for different plots, may prove to be the most valuable.

Examining the success rates of GBRT and BTC models for Set 1 along the weeks of the year (Fig. 6) reveals that while both models achieved 100% accuracy in most weeks (more than 40 weeks of the year), there were a few occasional weeks, in which the accuracy of one of the models dropped. This finding suggests that it is beneficial to consider the results of both models when deciding on irrigation. When the results of one model contradict those of the other, it is worthwhile to involve an agronomist in resolving the discrepancies. In addition, there was a period of time (5 weeks starting at the end of August) in which the accuracy of both models dropped to 50–80% due to deliberate and forceful soil drying, as noted above. It is expected that if data are collected over a longer period of time, this deliberate drying will be taken into account by the models.

The high prediction success rates of GBRT and BTC models (93 and 95%, respectively) suggest that they indeed can serve to facilitate and even automate irrigation decisions. The agronomist can use the models developed in this study instead of analyzing data from different plots and subplots, including weather data and forecasts, on a weekly basis. With time, as more data samples are collected, the prediction errors are expected to decrease and the models are expected to produce decisions that are identical to those of the agronomist.

For the agronomist to be able to effectively use these models for irrigation planning, they should be incorporated in a DSS that receives inputs from the uMANAGE system and from the meteorological services, that processes the inputs to fit the dataset structure described in Table 1 (this could be done on a daily basis—every day an observation that represents data of the previous seven days is added to the dataset), and that uses the models to predict the required irrigation quantities for the latest observation. Such a DSS would enable to increase the frequency of irrigation decisions so that decisions are based on more recent and accurate data and are able, for example, to better handle unexpected changes of weather.

As discussed in the " Introduction" sect., the literature includes various models for irrigation decision support. While many of the existing models support farmers' decisions by providing the farmer with simple water balance models (e.g. based on the Penman–Monteith equation) subjected to capacity constraints and by allowing the farmer to simulate different irrigation alternatives and assess their implications, the models developed in this study follow a different approach—they capture the actual decision-making process of the farmer. This approach has been used for example by Labbé et al. (2000), who modeled the decision making process of farmers for limited water allocation and irrigation scheduling for corn, based on surveys of farmers and based on monitoring their irrigation practices during a period of two years.

Instead of manually modeling the decision making process of farmers by surveying them and observing their behavior, in this study, ML was used to automatically learn their decision-making process from past irrigation decisions. The development of ML-based models requires considerably less effort than manually modeling the decision-making process, without compromising prediction accuracy. In fact, the GBRT models provided even better results in terms of average prediction error (the GBRT model for Set 1 achieved an RSME of 0.113 mm, which reflected an average error rate of 5.3%—less than an error rate of 6.7% achieved by Labbé et al. 2000). This study also advances the literature by taking into account farmer considerations that are plot-specific, suggesting that plot-specific data allows for plot-specific decisions. As noted above, a similar approach to ours was taken by Andriyas and McKee (2013), who used tree-based classification models for modeling the farmer's decision of whether or not to irrigate alfalfa, barley, and corn. The models developed in this study advance the state-of-the-art by predicting irrigation amounts and not only the binary irrigation decision.

This study demonstrated how in-field sensor data can be used not only for monitoring and control, but also for developing ML-based prediction models for Jojoba irrigation decisions. Irrigation prediction is just one area in which in-field sensor data can be used for prediction. Other areas are yield prediction and disease prediction, which, so far, have been mainly predicted using remote sensing and image processing. In the future, the development of a Jojoba yield prediction model is planned, using an approach similar to the one presented in this paper, but with yield as the target variable. To accomplish that, however, data describing subplots at a seasonal granularity, along with the yield per subplot, should be collected for at least ten years. In addition, application of ML techniques may reveal disease-potential situations and may assist in their prevention. For example, by analyzing in-field sensor data, it was already discovered that when the air temperature equals to the dew point temperature for several continuous hours, there is a high chance for fungal infection development in the following days. Thus, when such situations are identified they are immediately treated with pesticides to prevent disease development. The last two examples demonstrates the potential advantages of applying ML on in-field sensor data, together with additional data, for prediction purposes. As opposed to remote sensors, which can be seen as direct proxies of symptoms (e.g. plant disease or decrease in yield), soil and plant sensors serve as proxies of the actual state of the plant and soil, and they can reflect the causes for symptoms at real time. Consequently, models that are based on such data can predict diseases or yield decrease at earlier stages.

Conclusions

The objective of this study was to develop a model that captures the agronomist's irrigation planning process and that predicts his irrigation recommendations. This objective was achieved by applying ML on a dataset that integrated data from different sources. The dataset captured various sensor-based features, weather features, and features describing plots, in addition to irrigation levels that were determined by the agronomist. Based on this comprehensive dataset, different irrigation recommendation prediction models were developed, using three ML approaches: the traditional linear regression and two non-parametric approaches, GBRT and BTC. The models were trained against eight different feature subsets (as described in Table 2).

The results showed that non-parametric models, namely GBRT and BTC models, were more accurate in predicting irrigation decisions than linear regression models, with success rates of 93 and 95%, respectively. In addition, GBRT and BTC models, compared to linear regression, required fewer transformations to capture non-linear relations among variables. The models they provided, however, were more difficult to interpret, as they included fifty decision trees.

Although the GBRT and BTC models developed for the entire feature set (Set 1) provided the best results, it was shown that the GBRT model for Set 6 had the same success rate as that for Set 1, possibly indicating that drought/saturation and IPPD weather data could be excluded from the dataset with little loss in accuracy. In addition, it was shown that different sets of features achieved the best results for different plots, suggesting that more fine-grained data can improve decision making by providing the infrastructure for more fine-grained decisions on irrigation.

The contribution of this study is threefold. First, the study explicates the tacit knowledge that guides the abstract decision-making process of the agronomist. The resulting model can serve to automate the irrigation decision-making process. A common approach by farmers to decide on irrigation amounts is to use the Penman–Monteith equation (Allen 1998), which approximates net evapotranspiration (ET). However, since all the equation parameters are measured by the meteorological station for the entire land, the agronomist can calculate a single ET value for all plots, which would lead to a single irrigation recommendation for all plots. The agronomist of Jojoba Israel combines this ET value in his irrigation decision process with other measured parameters and plot-related features to define plot-specific irrigation quantities, resulting in different irrigation quantities for different subplots. This decision process is now captured by the developed models, which may increase the efficiency of the agronomist by automating or semi-automating the irrigation planning process and by obviating the need to analyze plant and soil data, plot by plot, on a weekly-basis. Modeling the decision-making process of the agronomist enables to replicate it to additional orchards or to perform it more frequently without additional effort.

Second, by developing models with different subsets of features, it was possible to derive insights about the different features and their contribution to the prediction model. In particular, it was shown that feature sets that included no data on drought/saturation and IPPD weather performed at least as well as feature sets that included all available data, calling into question the value of these data for predicting irrigation amounts. Additionally, it was shown that different feature sets were optimal for different plots, suggesting that it might be useful to develop unique models for different plots.

Third, this work generated insights on the suitability of different ML models for the irrigation planning problem, showing that non-parametric models, based on gradient boosting, achieved much better results than the traditional linear regression.

Altogether, these three contributions represent an approach that can be applied to additional crops, as well as to additional decision-making processes in agriculture (e.g. fertilization and pest-control decisions).

Acknowledgement This study has been funded by the Israeli Ministry of Agriculture and Rural Development, Grant Number 458-0603/13.

Appendix

Linear regression coefficients. See Table 9.

Table 9 The coefficients estimated by the simple linear regression model developed for Set 2

Term	Coefficient (standard error)
(Intercept)	6.47E + 01 (9.55E + 01)
Plot = P04	−3.39E-01*** (8.64E-02)
Plot = P99I	−1.13E-01 (6.87E-02)
Plot = P99Z	−5.05E-02 (6.37E-02)
Sub_plot = B	−2.49E-02 (7.07E-02)
Sub_plot = D	−1.24E-01. (7.14E-02)
Sub_plot = E	−2.92E-02 (9.27E-02)
Sub_plot = G	4.92E-03** (1.57E-03)
Year	−1.39E-02 (1.59E-02)
Week	−2.19E-02 (1.84E-02)
Moisture_sensor_deep	3.97E-02. (2.14E-02)
Moisture_sensor_shallow	2.02E-02 (2.04E-02)
Sallow_state_1	−2.24E-02 (1.67E-02)
Sallow_state_2	−1.25E-02*** (2.76E-03)
Deep_state_1	1.91E-03*** (5.51E-04)
Deep_state_2	1.65E-03 (8.56E-03)
Avg_of_humidity	1.11E-02*** (2.60E-03)
Avg_of_radiation	1.90E-03*** (4.79E-04)
Average_of_air_temperature	5.91E-02*** (9.48E-03)
Prediction_avg_of_humidity	−3.39E-01*** (8.64E-02)
Prediction_avg_of_radiation	−1.13E-01 (6.87E-02)
Prediction_avg_of_air_temperature	−5.05E-02 (6.37E-02)

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, $p < 0.1$; The following variables were excluded due to collinearity with other variables: sub_plot = C, sub_plot = F, sub_plot = H, organic_plot = TRUE

References

Allen, R. G. (1998). *Crop evapotranspiration: Guidelines for computing crop water requirements (FAO irrigation and drainage paper*. Rome: FAO.

Andriyas, S., & McKee, M. (2013). Recursive partitioning techniques for modeling irrigation behavior. *Environmental Modelling & Software, 47,* 207–217. doi:10.1016/j.envsoft.2013.05.011.

Bishop, C. M. (2006). *Pattern recognition and machine learning (Information science and statistics)*. New York: Springer.

Breiman, L. (2001). Random Forests. *Machine Learning, 45,* 5–32. doi:10.1023/A:1010933404324.

Cai, Q., Xue, Z., Mao, D., Li, H., & Cao, J. (2016). Bike-Sharing Prediction System. In A. El Rhalibi, F. Tian, Z. Pan, & B. Liu (Eds.), *E-Learning and Games Lecture Notes in Computer Science* (pp. 301–317). Cham: Springer.

Castle, M., Lubben, B. D., & Luck, J. (2015). *Precision agriculture usage and big agriculture data* (Cornhusker Economics). http://agecon.unl.edu/cornhusker-economics/2015/precision-agriculture-usage-and-big-agriculture-data. Accessed 24 Jan 2017.

Damas, M., Prados, A., Gómez, F., & Olivares, G. (2001). HidroBus® system: fieldbus for integrated management of extensive areas of irrigated land. *Microprocessors and Microsystems, 25,* 177–184. doi:10.1016/S0141-9331(01)00110-7.

de Miranda, F. R. (2003). A Site-Specific Irrigation Control System. *American Society of Agricultural and Biological Engineers*. doi:10.13031/2013.13740.

Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. *Journal of Animal Ecology, 77*(4), 802–813.

Evans, R., & Bergman, J. (2007). *Relationships between cropping sequences and irrigation frequency under self-propelled irrigation systems in the Northern Great Plains (NGP)* (5436-13210-003-02). http://www.reeis.usda.gov/web/crisprojectpages/0406839-relationships-between-cropping-sequences-and-irrigation-frequency-under-self-propelled-irrigation-systems-in-the-northern-great-plains-ngp.html. Accessed 8 Jul 2016.

Fayyad, U. M. (1996). *Advances in knowledge discovery and data mining*. Cambridge: MIT Press.

Hedley, C. B., Roudier, P., Yule, I. J., Ekanayake, J., & Bradbury, S. (2013). Soil water status and water table depth modelling using electromagnetic surveys for precision irrigation scheduling. *Geoderma, 199,* 22–29. doi:10.1016/j.geoderma.2012.07.018.

Jensen, A. L., Boll, P. S., Thysen, I., & Pathak, B. (2000). Pl@nteInfo® — a web-based system for personalised decision support in crop management. *Computers and Electronics in Agriculture, 25,* 271–293. doi:10.1016/S0168-1699(99)00074-5.

Kennedy, S., & Tobler, W. R. (1983). Geographic Interpolation. *Geographical Analysis, 15,* 151–156. doi:10.1111/j.1538-4632.1983.tb00776.x.

Labbé, F., Ruelle, P., Garin, P., & Leroy, P. (2000). Modelling irrigation scheduling to analyse water management at farm level, during water shortages. *European Journal of Agronomy, 12,* 55–67. doi:10.1016/S1161-0301(99)00043-X.

Lilburne, L., Watt, J., & Vincent, K. (1998). A prototype DSS to evaluate irrigation management plans. *Computers and Electronics in Agriculture, 21,* 195–205. doi:10.1016/S0168-1699(98)00035-0.

Liu, B. (2011). *Web data mining: Exploring hyperlinks, contents, and usage data*. Heidelberg, New York: Springer.

Liu, Y., Ren, Z.-h, Li, D.-m, Tian, X.-k, & Lu, Z.-n. (2006). The Research of Precision Irrigation Decision Support System Based on Genetic Algorithm. *International Conference on Machine Learning and Cybernetics*. doi:10.1109/ICMLC.2006.258403.

Loh, W.-Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews, 1,* 14–23. doi:10.1002/widm.8.

Mateos, L., López-Cortijo, I., & Sagardoy, J. A. (2002). SIMIS: The FAO decision support system for irrigation scheme management. *Agricultural Water Management, 56,* 193–206. doi:10.1016/S0378-3774(02)00035-5.

Mease, D., Wyner, A. J., & Buja, A. (2007). Boosted Classification Trees and Class Probability/Quantile Estimation. *The Journal of Machine Learning Research, 8*(5), 409–439.

Mitchell, T. M. (1997). *Machine Learning (McGraw-Hill series in computer science)*. New York, London: McGraw-Hill.

Murase, H., Honami, N., & Nishiura, Y. (1995). A nueral Netwerk Estimation Technique for Plant Water Status Using the Textural Features of Pictorial Data Plant Canopy. *Acta Horticulturae*. doi:10.17660/ActaHortic.399.30.

Smith, D., & Peng, W. (2009). Machine learning approaches for soil classification in a multi-agent deficit irrigation control system. *IEEE International Conference on Industrial Technology—- (ICIT), Churchill, Victoria, Australia*. doi:10.1109/ICIT.2009.4939641.

Sørensen, C. G., Fountas, S., Nash, E., Pesonen, L., Bochtis, D., Pedersen, S. M., et al. (2010). Conceptual model of a future farm management information system. *Computers and Electronics in Agriculture, 72,* 37–47. doi:10.1016/j.compag.2010.02.003.

Thysen, I., & Detlefsen, N. K. (2006). Online decision support for irrigation for farmers. *Agricultural Water Management, 86,* 269–276. doi:10.1016/j.agwat.2006.05.016.

Vellidis, G., Liakos, V., Porter, W., Tucker, M., & Liang, X. (2016). *A Dynamic Variable Rate Irrigation Control System*. St. Louis, Missouri: Academic Publishers.

Vellidis, G., Tucker, M., Perry, C., Reckford, D., Butts, C., Henry, H., et al. (2013). A soil moisture sensor-based variable rate irrigation scheduling system. In J. V. Stafford (Ed.), *Precision agriculture '13*. The Netherlands: Wageningen Academic Publishers.

Yunseop, K., Evans, R. G., & Iversen, W. M. (2008). Remote Sensing and Control of an Irrigation System Using a Distributed Wireless Sensor Network. *IEEE Transactions on Instrumentation and Measurement, 57,* 1379–1387. doi:10.1109/TIM.2008.917198.

Zhang, N., Wang, M., & Wang, N. (2002). Precision agriculture—a worldwide overview. *Computers and Electronics in Agriculture, 36,* 113–132. doi:10.1016/S0168-1699(02)00096-0.

Zou, H. (2006). The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association, 101,* 1418–1429. doi:10.1198/016214506000000735.