



HAL
open science

Short term soil moisture forecasts for potato crop farming: a machine learning approach

Amaury Dubois, Fabien Teytaud, Sébastien Verel

► **To cite this version:**

Amaury Dubois, Fabien Teytaud, Sébastien Verel. Short term soil moisture forecasts for potato crop farming: a machine learning approach. *Computers and Electronics in Agriculture*, inPress, 180, 10.1016/j.compag.2020.105902 . hal-03081945

HAL Id: hal-03081945

<https://hal.science/hal-03081945v1>

Submitted on 18 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Short term soil moisture forecasts for potato crop farming: a machine learning approach

Amaury Dubois^{a,b}, Fabien Teytaud^a, Sébastien Verel^a

^a*Univ. Littoral Côte d'Opale, UR 4491, LISIC, Laboratoire d'Informatique Signal et Image de la Côte d'Opale, F-62100 Calais, France*

^b*Weenat, Technocampus Alimentation 2 impasse Thérèse Bertrand-Fontaine 44300 Nantes*

Abstract

Agricultural decision-making is crucial for future yields. In the context of smart farming, grower combine information from sensors located close to their crops with agronomic models to help them to better understand their crops. Irrigation management is therefore based on extrapolation of data and/or agronomic model responses. This problem can be seen as a learning task for which machine learning techniques have proven their relevance in many and diverse applications. In this paper we place ourselves in the context of potato farming, a crop for which irrigation plays a crucial role. We model the problem of soil water potential prediction as a learning problem solved by supervised learning algorithms. The problem appears to be difficult since there are several potential inputs, and several outputs to predict. Experiments are conducted on several scenarios with data acquired during 3 years. We demonstrate the possibility of applying feature selection method to automatically design models with features relevant to the problem at hand while having good performances. We have also demonstrated the relevance of the machine learning for this kind of problem, since the methods are able to correctly predict the next water potential values.

Keywords: smart irrigation, feature selection, machine learning

Email addresses: amaury.dubois62@gmail.com (Amaury Dubois),
fabien.teytaud@univ-littoral.fr (Fabien Teytaud),
sebastien.verel@univ-littoral.fr (Sébastien Verel)

URL: <https://amaury-dubois.herokuapp.com/> (Amaury Dubois),
<http://www-lisic.univ-littoral.fr/~teytaud/> (Fabien Teytaud),
<http://www-lisic.univ-littoral.fr/~verel/> (Sébastien Verel)

1. Introduction

Developments in agricultural technologies are leading to its fourth revolution. The latter puts forward four essential requirements: increasing productivity, allocating resources reasonably, adapting to climate change, and avoiding food waste [1, 2].

One of the main challenges is to improve water resources management [3]. Smart irrigation directly affects crop performances and can lead to qualitative and quantitative improvements in vegetable production [4]. In the context of large scale agriculture, sprinkler irrigation (water cannon) remains the most widespread method of irrigation [5]. Unfortunately, this technique is expensive, time-consuming and water-consuming. Nowadays, growers need to capitalize as much information as possible in order to better understand their environment, and, to be able to be advised in their decision-making process. Two aspects are essential: first, knowledge of the current soil moisture is the most critical element in this process [6]. This value is dynamic, which means that daily monitoring is required to analyze global trends. One of the indicators used to conduct a cultivation campaign is the Soil Water Potential (SWP). This metric corresponds to the effort that the plant must develop in order to extract water from the soil. The higher this value is, the more difficult it is for the plant to extract water and therefore the risk of hydric stress occurs. Cultivators can directly use this information to manage their campaign [7] or can be transformed into Soil Water Content (SWC) using Van Genuchten's equation [8], if the soil characterisation is well known. This value is acquired via a tensiometer implant at the depth where you wish to observe. Generally sensors are located at the root level. From this value, thresholds can be established and characterize the soil water status (saturation, comfort, vigilance and stress) depending on the soil type. From there a grower can have a knowledge about the water reserves currently present in his soil. Second, being able to predict short-term evolutions of this information with a weekly vision is a major benefit for planters, since they can plan more efficiently their irrigations in terms of water input quantity or number of irrigations. Concerning soil moisture data acquisition, numerous methods have been proposed to acquire these information:

- Direct measurement or gravimetric method [9] remains the most reliable way to obtain these data and it is therefore essential for calibrat-

ing instruments used in the indirect methods. However these methods are extremely expensive, require numerous batteries of laboratory tests and destructive measurements taken in the field which lead to a high manpower cost, time consuming and potential handling problems.

- Indirect or sensor measurement methods are good alternatives, allowing measurements without sampling [9]. Many sensor models exist (tensiometer, capacitive, resistive, hygrometric, ...), their measurement methods differ but also allow to access to the knowledge of the soil moisture. However, the quality of the measurements is highly dependent on the quality of the sensors used. Indeed the sensors are subjected to climatic conditions (wind, rain, temperature) for a period of several months in the open field, which can lead to a degradation in their accuracy and reliability [10]. Tensiometers are often preferred to other types of substrate moisture sensors due to their low cost, simplicity of use, and direct measurement of matric potential, and also, they are not influenced by temperature and soil osmotic potential [1, 11].

In a context of connected agriculture, the methods of data acquisition by sensor benefit from the advantages of IoT and can therefore upload data frequently in order to allow a monitoring up to a quarter of an hour. The proposed methodologies to make forecasts are based on the previously collected data. Latter are used as inputs into mathematical models, for instance: Jhonson [12], STIC [13], AQUACROP [14], WEEDRIQ [15], water balance [16], Penman [17]. This approach has proved its relevance and robustness, unfortunately mathematical models introduce biases and biological simplifications. Moreover, they are parametric which means they have to be calibrated in order to use them optimally [18, 19, 20].

The natural or biological phenomena under study involves features with a relationship among them that may be non-linear. Moreover, when the conditions of the phenomenon change, the relationships may also be modified. However, collected data may include similarities, natural symmetries, patterns and thus; designing a data-driven model able of approximating the relationships between the different elements is suitable. One way to deal with the data-driven approach is the Machine Learning (ML). In recent years, Machine Learning methods have been successfully applied to many real world applications, and in particular [21, 22, 23] demonstrate the relevance of modeling complex biological phenomena such as evapotranspiration by transformation into a supervised learning problem solved by algorithms

as Neural Network (NN), Random Forest (RF) or Support Vector Machine (SVM). [24] pointed out that ML approaches, in most cases, outperformed parametric approaches for the prediction of biomass and soil moisture from satellite data.

In the context of agriculture, [25] successfully applied machine learning approach to predict weekly evapotranspiration for orchards crop and therefore the water needs. On the subject of potato cultivation in farmland, various topics are highlighted and well-documented. Methods using machine learning algorithms are successfully applied to provide a solution on these topics such as predicting leaf water potential [26], modeling root development [27], tuber growth [28], daily evapotranspiration prediction [29, 30, 31], etc. The issue of evapotranspiration prediction is well documented, however concerning the SWP, to our knowledge, a methodology for weekly prediction of this indicator using a connected sensor-powered ML approach has not been proposed. In most studies involving sensors inserted in the ground, only one depth is considered. However, observing at several depths allows us to capture information relating to water exchanges between the different layers, for instance: percolation, capillary upwelling, etc.

In most cases, the proposed resolution methods are developed with small experimental plots, and because of this, they are given for a particular climatic, geographical and soil type context. Thus they may not be applied directly in commercial agricultural fields [32]. Thus, it is still necessary to develop simple models that are executable in the practice of irrigated agriculture and at the same time are developed for large areas of agricultural crops [32].

There are many approaches but 2 main approaches are used:

- model-based, in this case a reference model (e.g. Penman-Monteith [31, 33] or AquaCrop [14, 34, 27],...) are used, generally an optimization is performed to improve the quality of predictions.
- Data-driven approach uses, as described above, machines learning algorithms to generate models from observations obtained directly on fields [30]. These approaches have shown their relevance and robustness.

The literature is very well documented on the methods and possible approaches for solving the above mentioned problems. In this study we have chosen to follow one of the methodologies commonly used to solve learning

processes. Initially, features are defined either by experts in the field or by biological model equations. These characteristics represent the data necessary for learning the phenomenon or for its use in the equations. Then, data are collected through the use of sensors, and sometimes data sampling, allowing the collection of several historical campaigns aggregated together in order to generate the observation base. In the learning stage, the latter is then divided into 2 parts: learning and validation database. A strong constraint on the creation of the bases is the non division of a campaign. Indeed, if data from a campaign in the validation database are available for learning, then the model has a knowledge of the validation data and therefore loses its relevance. Eventually, the validation of the method consists in comparing the predictions made with a model (already existing reference model or generated by a learning algorithm previously learned on the learning database) with the response of a comparison model or with a validation database (unknown observations) made with the collected data.

In this paper, we are interested in potato cultivation, for which irrigation plays a crucial role to avoid water stress (negative balance of water supply compared to plant transpiration). This can lead to a collapse of yields [35]. Using quality sensors can be a limiting factor for a large majority of farmers due to their expensive prices. In order to democratize these techniques we propose in this paper the use of low-cost sensor sets. Using affordable tensiometers is a good compromise between the accuracy of the measurements and the feasibility of being used by a large number of farmers. Moreover, it becomes possible in the long term to generate a database that is certainly inaccurate, but which can cover different types of soil, crops, climatic conditions, etc, which is crucial for a production start-up. Finally, it is possible to install sensors at different depths and thus modeling the evolution of the SWP at several layers as well as possible capture exchanges between them. We model the problem of predicting water potential as a learning problem. Since real data are difficult to obtain, we propose to aggregate the values of several type of sensors (tensiometer, rain gauge, temperature) into inputs. We take into account the values of the tensiometers at 3 depths. We design the inputs for use in the learning process. The water potential value will be used as a label (output) in order to solve this problem by supervised learning algorithms like NN, RF and SVM. The learning process does not necessarily need all the features to find a relationship that best explains the phenomenon. Thus, judiciously choosing the most proper features leads to a model that may be easier and faster to learn, more robust and/or more expli-

cable. In addition, later on, if other features are used to improve the existing model, the feature selection method will be able to keep them or reject them automatically if they are not relevant. The novelty of this work is to propose a methodology to forecast SWP trends at several depths up to week (which is a important metric for crop management and decision-making), thanks to a learning process in a potato growing context fed by remote sensors. we also propose to use feature selection method to automatically design models. To our knowledge there is no similar work. Contributions of this work can be summarized as follows:

- We model the soil water potential prediction problem as a learning problem solved by supervised learning approach and low-cost sensors for data collection.
- We predict the water potential of a potato crop farmland for up to a week at 3 different depths .
- We use feature selection methods, in order to automatically design model with the most relevant features.
- We compare performances of models (with and without feature selection) of different machine learning algorithms under different climatic conditions.

The scope of this contribution is limited for the moment to one type of soil: "sandy clay loam". The goal is indeed to put aside, for this study, the influence of the soil composition on the evolution of the soil water potential.

The rest of the paper is organized as follow: Section 2 detail the study area and data and explains the different supervised learning methods used in this work. Section 3 explains the feature selection used in this work. Section 4 details the methodology used to address the issue. Then, we present performance evaluation and experimental results in Section 5. Finally we conclude and present our future work in Section 6.

2. Materials and methods

2.1. Study area and data

The predictive model is built using sensor data from historical potato growing campaigns. The test crops used are close to each other and are

located in the north of France, on the seafront. The climate is oceanic, which characterizes variable, windy weather, moderate summer temperatures, prominent humidity, and abundant but low precipitation. We assume from experts that crops have the same type of soil.

Soil moisture acquisition is carried out by tensiometers. In this study 3 depths are considered: 20, 30 and 40 cm depth. As, in general, potato’s roots depth do not exceed 35-40 cm in this region. These depths are appropriate in order to predict the water potential, since it is at these depths that the plant collect the water it needs. We use 3 measurement points per depth (i.e. 9 measurements) since the sensors are not accurate (because we used cheap sensors, see section 1), the average of 3 measurement points is an acceptable compromise. The weather is provided by weather stations which are located less than 5 km away from the crops. The measurement campaign covers a period of 3 years (2016, 2018 and 2019) over the months of April to September. The year 2016 is characterized by heavy precipitation and 2019 by low precipitation. Over these 3 years, we have data from 3 or 4 crops per year, resulting in 10 time series in total. Each day of measurement is considered as an independent observation and are our learning database. The growing time for potato cultivation is short, averaging 90 days, however we had to remove the days where sensor values are not usable. There are different reasons why values are not correct, for instance, during the sensor calibration period, or because of technical problems or sensor displaced due to tractors passing.... After this preprocessing, the number of measurement points is about 750. A measurement point corresponds to the information provided by the sensors for a given day. All the characteristics to be taken into account in this phenomenon have been provided by the agronomists and are listed in Tab. 1.

For each example, we have the weather of the past 3 days and the weather forecast of the next 7 days. The total number of features is then 45. The database includes the daily average values of the tensiometer at 3 depths of the different crops, from planting to defoliating, as well as water inputs, irrigation, rainfall and temperature for the weather.

the input data are normalized in order to remove the scaling effect of raw data. The normalization process can be done by applying Eq. 1 on the raw data.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Table 1: The 5 features we used in the experiments presented in this paper.

Feature	Description
Water_Pressure	Water pressure applied to the sensor in <i>Kpa</i> (3 depths in this work)
Mean_Temperature	Current average temperature in <i>Celcius</i> of the day
Rainfall	Current rainfall and irrigation in <i>mm</i> of the day
AGE	Age of a plant expressed in degree <i>Celcius</i> per day resulting from the sum of degrees above a threshold (for potato 6 degree <i>Celcius</i>)
Dry	Indicate if Drought, i.e if water pressure is above 140 <i>Kpa</i>

where x_{norm} is the normalized data, x is the raw data, x_{max} is the maximum value of input vector and x_{min} is the minimum value of the input vector. Fig. 1 shows the correlation matrix of the different features at our disposal. We have the 5 previously defined features among the time range studied:

- Water pressure: P1, P2, P3 and P_day_* or all previous and next days.
- Mean temperature: T_mean_* for all previous and next days.
- Rainfall: Rainfall_* for all previous and next days.
- Age: Age_day_* or all previous and next days
- Dry: Dry_P1, Dry_P2, Dry_P3 binary value that states whether the sensor is out-of-threshold.

Some features are more correlated than others: rainfalls and irrigations forecast (i.e. water inputs) have a serious negative correlation with the water Pressure (P1, P2 and P3), which is expected since water inputs is in this study the only, a priori, possible way for this value to decrease. However, these features have a stronger influence on the upper ground layer than the others, which seems expected, since the water input will reach the first layer first. Age and Temperature also seems to have an impact, which is also expected.

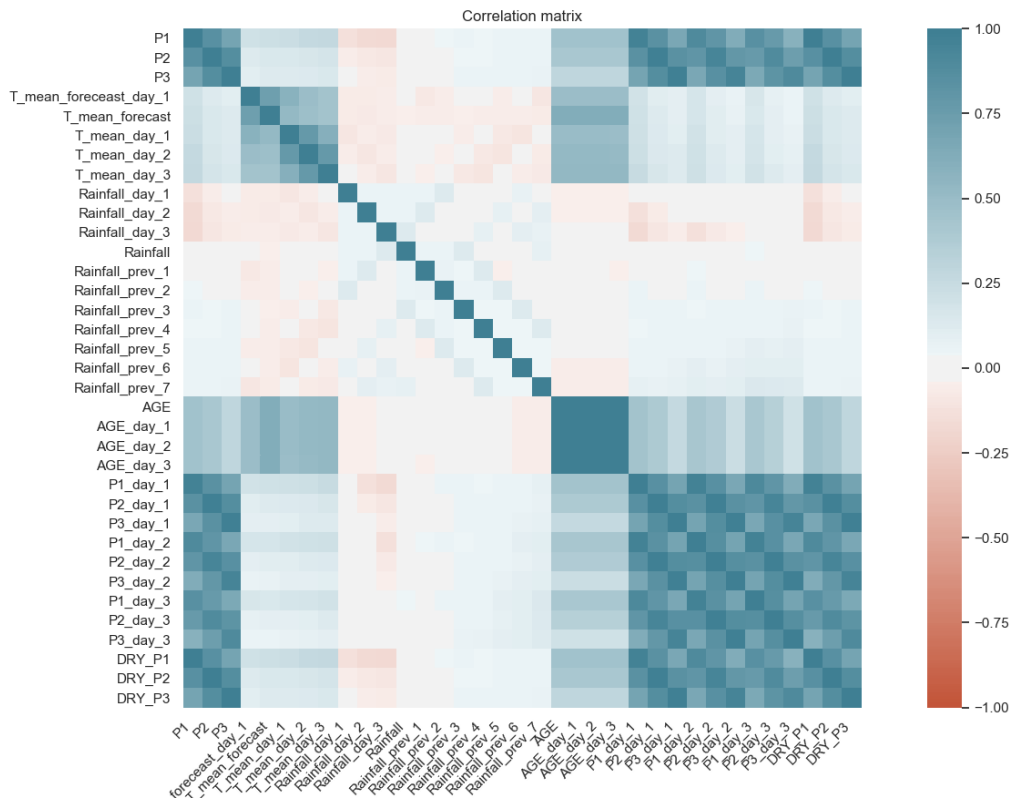
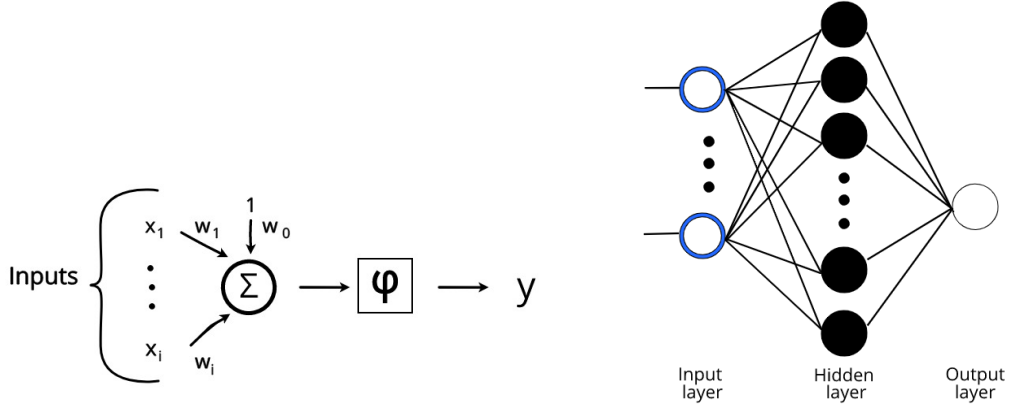


Figure 1: Correlation matrix of the 5 features used in this study, among the time period studied (see text).



(a) Simplified neural schema, with inputs x_i , bias w_0 , a transfer function φ and an output y .

(b) Example of a feed-forward neural network fully connected with one hidden layer and one output unit.

2.2. Neural Network

A neural network [36] is a computational system inspired by biological neural networks. It is able to learn, without the need of prior knowledge, specific tasks using examples. Fig. 2a represents a neural schema. A network is made up of neurons. Each neuron is defined by several weighted inputs, and a bias. Equation 2 explains the neuron computation. The sum of inputs x_i multiplied by their corresponding signals w_i , including the bias w_0 , is given to a transfer function φ which can be for example a sigmoid, a relu, or a hyperbolic tangent function. Formally, the final output value y of the neuron with p inputs is computed following:

$$y = \varphi \left(\sum_{i=0}^p x_i w_i \right) \quad (2)$$

Neurons are often grouped into layers: *an input layer, one or more hidden layers and an output layer*. Thus, neurons of the same layer share the same inputs but there are not connected to each other. The output values of the neurons in one layer are used as inputs to the neurons in the next layer. In this paper, the considered neural network is a feed-forward fully connected, which means that every unit belonging to each layer is connected to every unit belonging to the adjacent layer without any loop or cycle. Thus the information is transferred from input layer to output layer through each hidden layer

(see Fig. 2b). The output layer consists of several neurons representing the different desired predictions (see Section 2.1), and the transfer function used for this layer is a simple linear function. Predicting with a neural network involves a learning phase. During this phase, the learning process is supervised. Thus, thanks to a set of examples with expected output information (predictions), the model can be trained using an optimization process. The more various examples there are, the better the prediction of the model may be. In our case study, we predict the soil moisture, that is a continuous value. This problem is known as a regression problem. During the learning stage, its ability to generalize is tested on a test base, i.e. an unknown set of examples (see Section 5). We use a standard backpropagation algorithm [37]. The error between predicted value and the sensor value is back propagated to the input layer adjusting the weights and biases in each layer thanks to an optimization algorithm. In the context of low data there is a risk of overfitting: the model can learn the statistical noise of small data set, which leads to poor performance in generalization (i.e. on unseen examples). Combining many networks with different topologies on the same database reduces this problem but in practice it is quite difficult to use. One regularization method to prevent overfitting is the dropout [38]. The idea behind this approach is to simulate many network topologies from a single network.

2.3. *Random Forest*

A Decision Tree is a machine learning algorithm which is invariant under scaling and various other transformations of feature values. It is robust to inclusion of irrelevant features, and produces understandable models. However, they are rarely accurate, and if the decision tree grows too much and there is a risk of overfitting. The first designation of the Random Forest [39] is a combination of Breiman's bagging idea and random selection of features introduced by Ho [40]. Random Forest is an ensemble learning which combines the concepts of random subspaces and bagging. The random forest algorithm performs training on multiple decision trees trained on different subsets of data in parallel. The construction of decision trees is done using the bootstrap method to randomize the use of variables and data. In the context of regression, the prediction of unseen samples is made by the mean of the output. This approach improves the performance of decision trees because it decreases the variance of the model, without increasing the bias. RF algorithm can be defined as described in algorithm 1. This algo-

Algorithm 1: Random Forest

Input: S : training set $(x_1, y_1), \dots, (x_n, y_n)$; F : set of features; B : number of trees in forest.

Output: H : set of trees (random Forest regressor)

```
1 begin
2    $H \leftarrow \{\}$ 
3   for  $i \in 1 \dots B$  do
4      $S_i \leftarrow$  A bootstrap sample from  $S$ 
5      $h_i \leftarrow$  RandomizedTreeLearn( $S_i, F$ )
6      $H \leftarrow H \cup \{h_i\}$ 
7   return  $H$ 
```

rithm has several parameters. The 3 main parameters free to be tuned in our implementation are:

- Minimum sample leaf
- Number of decision trees
- Maximum feature attributes

In this study we use the scikit-learn [41] library to implement RandomForest.

2.4. Support Vector Machine

Originally developed for classification tasks [42], this algorithm is used for regression [43] (called Support Vector Regression or SVR). This algorithm can be applied for many machine learning problems and can, depending on the data, have performance similar or superior to the Neural Network and RandomForest. Its advantage is to be able to work with large data sizes while having a low number of hyperparameters to tinned and it is known to have good results in practice. The concept of its approximation function is based on the Statistical Learning Theory [44] plus a set of normalized input/output examples. In context of regression, let be a training sample $X = \{(x_1, y_1), \dots, (x_n, y_n)\}$, the goal is to find an $f(X)$ as close as possible using the form given in Equation 3 [45]:

$$f(X) = W.\phi(X) + b \quad (3)$$

Where $\phi(X)$ is a matrix with high-dimensional space. To estimate coefficients W and b a regularised risk function is used [46]. Although we consider that if the loss is zero it implies that $f(x_i)$ explains y_i , $\forall i \in X$. SVR assumes that we can accept a maximum ϵ deviation between $f(x)$ and y . The underlying idea is to avoid overfitting by accepting an error's margin. Thus, SVR calculates the loss when the absolute value of the difference between $f(x)$ and y is greater than ϵ . During the training, if a point is predicted within an area centered on y and with width of ϵ then the point is considered correct. The impact of the error margin is therefore a crucial parameter to tune in order to have good performance in generalization: a too small margin allows a better learning at the risk of over-learning while a too large margin can have poor performance in generalization. To implement the SVR, we use the scikit-learn library¹. 2 parameters are free for adjustment:

- ϵ , which is the width of the acceptance band.
- C , a regularization parameter. The strength of the regularization is inversely proportional to C . It must be strictly positive.

3. Feature Selection

Feature selection is a preprocess method of dimensionality reduction . Its objective is to generate a lighter model by excluding some of the features that contribute the least to explaining a phenomenon while limiting the degradation of model performance. In context of several outputs to predict, each output must be considered independently, since it is impossible that the same features contribute the best to all outputs. In this case the resulting lightweight models can be seen as a unique collection of features from the list of selected features for each output (possibility of features redundancy).

From a set of examples containing inputs and outputs, the algorithm uses a metric using the correlations/relationships between features to establish a ranking order based on their contribution to explain the output. There are several methods of feature selection with different strategies such as Recursive Feature Elimination (RFE) or Filter Method (FM), Forward Selection (FS), Each method needs a metric to classify the features in order of importance. Thus a estimator is used to provide them with this information. All algorithms that can generate this metric can be used, for instance,

¹<https://scikit-learn.org/stable/>

Random Forest (RF), Support Vector Machine (SVM) with a linear kernel, LASSO, One of the advantages of these methods is that they do not require any prior knowledge to select the features that best explain the problem. However, a major drawback of this approach is the weighting issue. Indeed, the ranking of features is established prior to the training based on the available data (train database). If an unknown observation disagrees with the learning base, then the ranking can be questioned and may lead the algorithms to exclude one or more relevant features and thus penalize the generalization of the light model. In this study, 2 methods of features selection are studied:

- Filter method [47, 48] : A estimator is used to assign a score for each feature and rank them in order of importance. The selection of the best *a priori* parameters is made according to a threshold. All features with a score greater than or equal to the threshold are preserved. The value of the threshold is a hyper parameter which is provided by the user. Higher the threshold is, the fewer features are kept.
- Recursive Elimination Feature [49]: From the available set of features and a Given external estimator that assigns weights to features (e.g., the coefficients of a linear model), the goal of recursive feature elimination , is to recursively pruned a feature that contributes the least to the explanation of the output. First, the estimator is trained on the initial set of features and the importance of each feature is made. The feature which contribute the least is pruned and this procedure is recursively repeated on the pruned set until K best features is eventually reached. The value of the hyper parameter K is provided by the user. Feature selection strategies can therefore be very different. For example, RFE uses an iterative process while FM uses a one-shot approach. When the number of features to be estimated is important, RFE can take more time to calculate than FM. In case of several output to consider, the process cannot be directly applied to all of them. Indeed the methods are used to capture the relationships between the inputs and 1 output. In order to tackle this problem, it is necessary to decompose the problem with X outputs into X problems with 1 output. Each of them has the same inputs. In this way the feature selection methods are able to select specific features for each output.

Algo 2 and Algo. 3 describe the feature selection methods used in this study.

Here again we have used the scikit-learn library for both these approaches.

Algorithm 2: Filter Method

Input: α : acceptance limit ; X : training set $\{(x_1, y_1), \dots, (x_n, y_n)\}$;
 F : set of features $\{f_1, \dots, f_n\}$; R : regression method.

Output: I : a set of most important features

```

1 begin
2    $M \leftarrow \text{fit}(R, X)$ 
3   foreach  $f \in F$  do
4     if  $\text{ScoreImportance}_M(f) > \alpha$  then
5        $I \leftarrow I \cup \{f\}$ 
6   return  $I$ 
7 .

```

4. Processing modeling

The problem of soil water potential prediction is similar to the problems described in section 1, and recently in Yamacc [31] which proposes a comparison of different learning algorithms to estimate the daily evapotranspiration value for the potato crop over 4 scenarios against an FAO equation. Thus the methodology used in this work follows the same principles:

- All available features used for the learning process are given by agronomists.
- Data collection is carried out using sensors.
- Historical campaigns are divided into 2 databases (learning and validation) without introducing knowledge of one into the other.
- Models are generated using supervised learning algorithms.
- Validation is carried out by measuring the model prediction error on test campaigns.

Moreover, like [31, 50, 33] we compare different learning algorithms, in our case NN, RF and SVR which have proven their relevance for solving similar problems (see section 1). However, we choose to predict SWP and not evapotranspiration and those over a larger time window, namely 7 days (like in [25])

Algorithm 3: Recursive Feature Elimination.

Input: K : number of features to select $\neq 0$; X : training set $\{(x_1, y_1), \dots, (x_n, y_n)\}$, R : regression method; F : a set of all features

Output: F : a set of most important features

```
1 Function RFE( $K, F, X$ )
2   if  $K \neq 0$  then
3      $M \leftarrow \text{fit}(R, X)$ 
4      $f^* \leftarrow \arg \min_F \text{ScoreImportance}_M$ 
5      $F \leftarrow F - \{f^*\}$ 
6     return RFE( $K - 1, F, X$ )
7   else
8     return  $F$ 
```

in the future in our study compared to 1 day proposed in most of the studies mentioned above, which is important to note. Compared to other studies, we assume that the sensors used here are not necessarily very accurate, because the objective is to be able to test the relevance of their use to collect data to learn the phenomena in a production context (*i.e.* with all sources of uncertainty, technical problems, ...). Moreover, the choice of features is an important point to take into account in learning problems [51, 52, 53]. Therefore, we add to our methodology a feature selection step, with the intention to automatically generate lighter or robust models but also to select the most suitable features and thus make the models resistant to the insertion of non-essential features as they will be automatically excluded. Indeed, it is possible that not all features are necessarily useful to the algorithms to explain the observations. Thus, we investigate 2 approaches of feature selection: Recursive Feature Elimination (RFE) [54, 50] and Filter Method (FM). To the best of our knowledge, this work is new in the field of potato cultivation. In this study, we assume the weather forecasts are correct, and with a perspective to a production release, we limit the forecast to 7 days due to the reliability of our current weather models. The data set is divided into 2 parts:

- the training set, which is the available data to design a predictive model.

- the test set, that corresponds to an entire growing campaign on a crop that is not in the training set. This is the cleanest way to have a independent test set, and to be sure not to introduce any bias. Indeed, since the data are from a few time series, using random sampling on all of them may lead to testing the forecasting on points close to those it has already learned (i.e. not be fully independent).

We perform a leave-two-out cross validation: two crop are removed from the available data set to serve as the test set and the others 8 crops corresponds to the current the training set, and then, we loop on all fields. This design choice is a compromise between the low amount of learning data, and the statistical accuracy to be able to interpret the results. Thus, the number of test folds in cross-validation method becomes $N = 45$ where $N = (numberOfCrop * (numberOfCrop - 1))/2$ since we have $numberOfCrop = 10$ which is large enough to be statistically significant. Our problem is not trivial as there are 21 outputs to predict, *i.e* 3 depths to predict over 7 days.

Thus, we design 2 sets of experiments:

- For the first experiment, we are interested in the impact of the feature selection method in the creation of lighter model as well as their performances. As these methods need a estimator to generate a metric for scoring the contribution to the output, we are also interested in the influence of the algorithm used. Finally we study the importance of the threshold value.
- For the 2nd experiment, we compare both the best lightweight models for each algorithm and models with all available features. We are interested in the global performances as well as in the evolution of the error according to the *i*-th day to be predicted in front of different metrics.

Additionally, we present the predictions of our decision support tool on particular test campaigns. The interest here is to provide readers with a visual of what will be proposed for tiller as well as difficulties related to this issue. The weather (in particular the precipitation) and the irrigations have a strong impact on the soil moisture between the depths of 0 and 20 cm, corresponding to the area above the upper sensor. It is important to keep in mind that it is difficult to model what happen between the ground surface and the measure of the upper sensor (the same difficulty appears between where the water is directly usable by the roots. The unit used by sensors is the Kpa and the

range of values stands between 0 kpa (rainoff soil) and 200 kpa (drought). It should be noted that values between 0 and 20 kpa (soil saturated) and above 150 kpa (soil very dry), can not be measured accurately by sensors. In this work, the proposed neural network consists of a feed forward neural network developed thanks to the Keras library². The network is composed of 1 hidden layer of 600 neurons using a sigmoid activation function and 1 output layer of 21 outputs using a linear activation function. The reason for the number of output is explained in the section 2.1. In addition, we use a dropout at 0.2 in order to avoid overfitting. The proposed topology is the result of an empirical study of different topologies. We present here the one that has obtained the best results at the moment. For Random Forest, after a grid search study we set the number of decision trees to 500 and the number of maximum attributes for a tree equal to the number of available features, which gives experimentally the best results. All other parameters are the default ones in the scikit library. For SVR, we use the 'linear' kernel, other kernels were tested but this one had best results the C -value is set to 5, and ϵ -value to 10^{-1} which gives experimentally the best results.

5. Results and discussion

This section presents the different experiences that have been made as well as an interpretation of the results.

5.1. Performances evaluation

To evaluate the accuracy of our models, we consider the *Mean Absolute Error* (MAE) (see eq 4), the *Root Mean Squared Error* (RMSE) (see eq 5) and the *R squared* (R^2) (see eq 6).

$$\text{MAE}_{\text{depth}(j)} = \frac{1}{n} \sum_{i=1}^n |y_{i,j} - \hat{y}_{i,j}| \quad (4)$$

$$\text{RMSE}_{\text{depth}(j)} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_{i,j} - \hat{y}_{i,j})^2} \quad (5)$$

$$R_{\text{depth}(j)}^2 = 1 - \frac{\sum_{i=1}^n (y_{i,j} - \hat{y}_{i,j})^2}{\sum_{i=1}^n (y_{i,j} - \bar{y}_{i,j})^2} \quad (6)$$

²<https://keras.io/>

where n is the number of examples, $y_{i,j}$ the real tensiometer value for the day i at depth j , $\bar{y}_{i,j}$ average of the measurements and $\hat{y}_{i,j}$ the model prediction for the day i at depth j . Since we have 3 depths to consider each day, for a given day the MAE, RMSE and R^2 criteria to minimize, is then average the errors over the 3 depths:

$$\begin{aligned} \text{MAE} &= \frac{1}{3} \sum_{j=1}^3 \text{MAE}_{\text{depth}(j)} \\ \text{RMSE} &= \frac{1}{3} \sum_{j=1}^3 \text{RMSE}_{\text{depth}(j)} \\ R^2 &= \frac{1}{3} \sum_{j=1}^3 R^2_{\text{depth}(j)} \end{aligned}$$

Please note that for the models resulting from the feature selection stage are grouped together and their performance is thus measured in the same way as for global models, i.e. with all outputs to be considered at the same time

5.2. Feature selection comparison

For the first experience, Figs 2, 3, and 4 show the impact of the hyper parameter of the two feature selection methods (*Threshold* for Filter method (FM) and k , the number of features to keep, for RFE) in front of different metrics defined by Eqs 5.1 after a leave-two-out cross-validation has been performed. As explained in section 4 using a leave-two-out cross validation allows us to have statistically significant performances and so we can compare the means.

For the FM, on the x-axis, high threshold values mean that a small number of features are selected. On the opposite, for the RFE, high values mean that a large number of features are retained. On these graphs are represented the various estimators used to select the features: SVR with a linear kernel (SVRL), LASSO with cross validation (LASSO) and Random Forest Regressor (RF). Other estimators can be considered. Concerning the results, first, the performances of the feature selection methods are quite similar between the learning algorithms. The rapid decrease of the error (MAE and RMSE) when the hyper parameter is low (up to 10 for LASSO and 15 for RF and SVRL) for RFE indicates that from this value the models have enough feature to be able to learn the phenomenon properly. According to the results,

RF

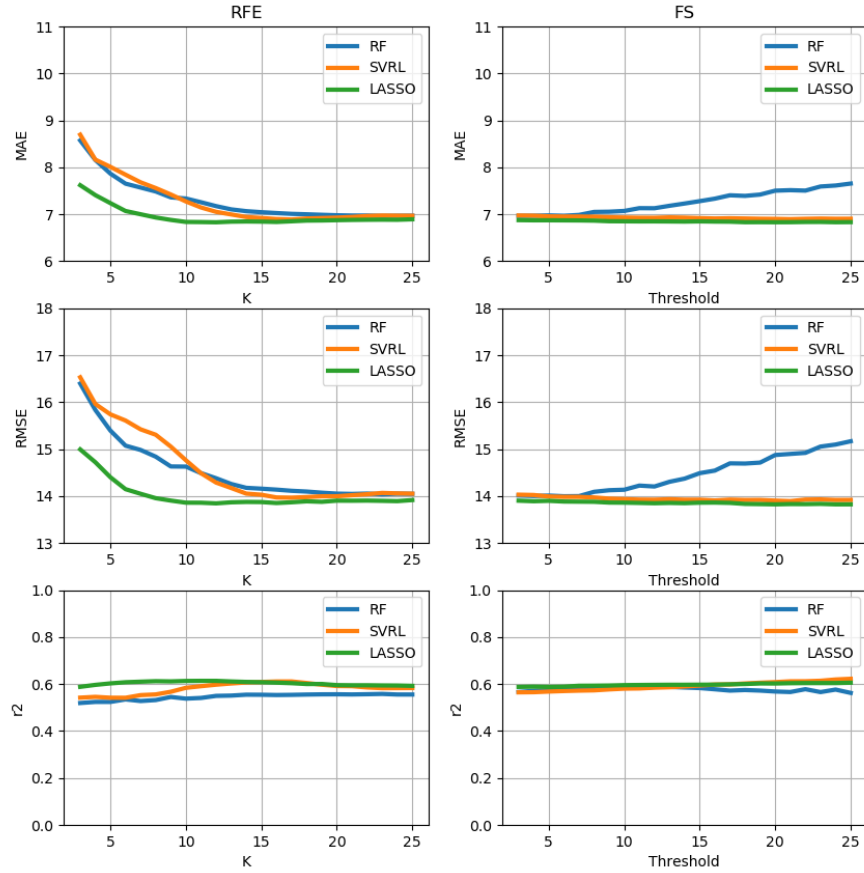


Figure 2: Hyper parameter tuning of the feature selection methods with Random Forest Regressor learning algorithm according to several metrics. The higher the k parameter (RFE), the more parameters per output are retained. A high *Threshold* (FM) indicates that only parameters with a significant contribution score per output will be retained. Except for R^2 , lower is better.

SVR

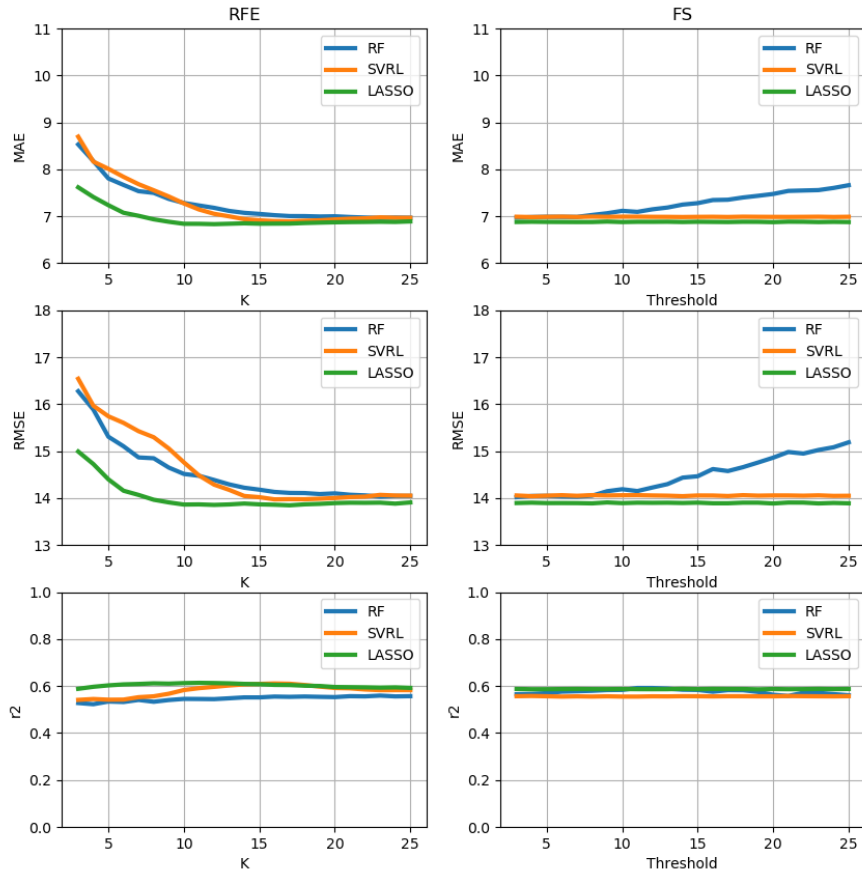


Figure 3: Hyper parameter tuning of the feature selection methods with Support Vector Regressor learning algorithm according to several metrics. The higher the k parameter (RFE), the more parameters per output are retained. A high *Threshold* (FM) indicates that only parameters with a high contribution score per output will be retained. Except for R^2 , lower is better.

NN

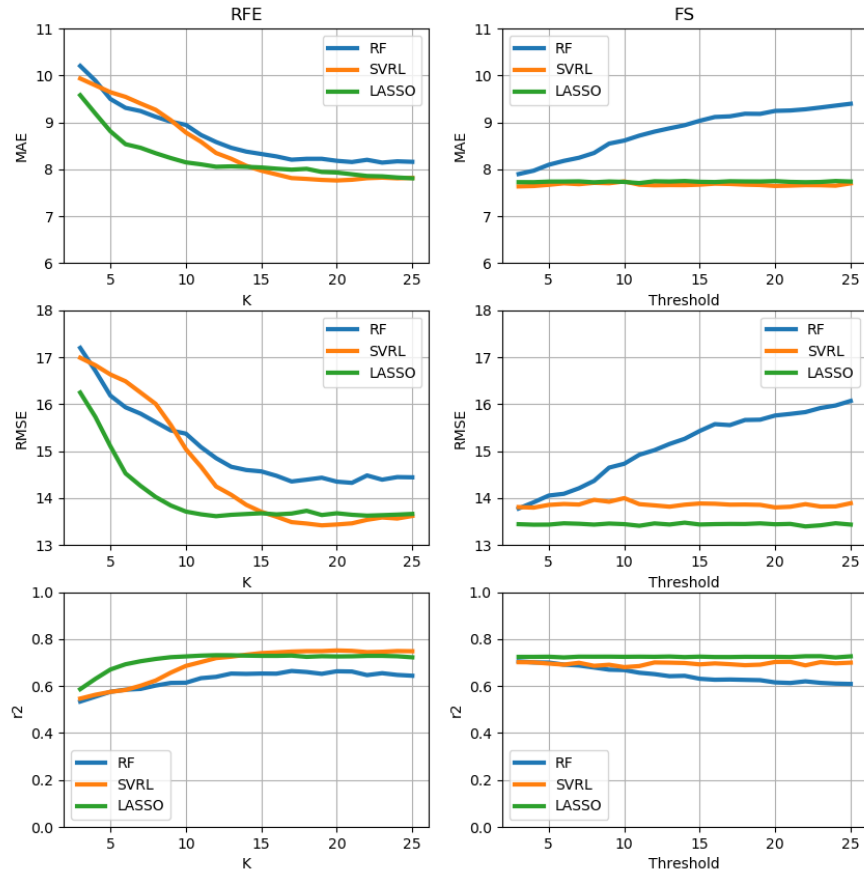


Figure 4: Hyper parameter tuning of the feature selection methods with Neural Network learning algorithm according to several metrics. The higher the k parameter (RFE), the more parameters per output are retained. A high *Threshold* (FM) indicates that only parameters with a high contribution score per output will be retained. Except for R^2 , lower is better.

LASSO is able to select the most relevant parameters when the hyper parameter is very low whatever the learning algorithm. Then its performance is comparable to SVRL. about RF, for the Feature Selection, According to the results, for RFE, higher the k parameter is, the better the prediction is. Indeed, both the MAE and the RMSE are decreasing while R^2 slightly increase which is expected. Concerning the estimator, although the performances of RF and SVRL are equivalent, Lasso with cross validation has a better dynamic. We can notice that LASSO find a consistent set of parameter with $k = 10$ which is much better than RF and SVM whatever the metric. Beyond this value the MAE and RMSE performances converge but the R^2 score remains slightly higher than SVRL and higher than RF. This suggests that LASSO has succeeded in selecting the most important features. It can also be argued that the resulting model has lower extreme deviations (lower RMSE) than the other methods, which should also explain the higher correlation coefficient. As explained in section 3, since several outputs have to be taken into account, there are k features selected per output, but since these outputs are not identical, the combination of all these outputs forms a larger set. For instance, for RF using a RFE method with LASSO as an estimator with hyper parameter $k = 10$. the resulting models so use 10 features by output but the overall unique collection set is 30 features, i.e. about 25% less than the model with all features. On his side, for FM, the strategy is different. Higher the threshold is, lower the number of feature is kept. But Only those with an importance score greater than or equal to the threshold are retained. Thus there is no guarantee about a constant number of features per output. Here, LASSO and SVRL have similar performance while RF shows a degradation from $Threshold = 7$. This suggests that RF score less effectively the features. Here, the best setting of the hyper parameter is reached for LASSO and SVRL when $Threshold = 16$ for these value, the number of unique feature is then 43 (up to 33 features by output).

As SVR's performances are identical, the conclusions are therefore the same. The number of features per output depending on the selection method and the estimators are the same. Finally for NN, for RFE, according to the errors, MAE is higher than for RF or SVR. But RMSE and R^2 are better, with respective values of 13.7 and 0.78 achieved with SVRL as estimator and $K = 20$. The resulting models use 20 features by output and the unique collection set of features is then 33. These results indicate that on average NN has lower extreme errors. About the estimators, RF has the worst performances whatever the feature selection method. This echoes what has

been said before. For the FM, Apart from RF, SVRL and LASSO also have similar performances. Since the impact of the threshold and estimator is low, if we look at $threshold = 25$, then the number of features per output is at most 29 for a total of 35 unique features.

For feature selection method, first, at that point, reformulate a problem with several outputs into several problems with 1 output using a feature selection method allows us to automatically design models with the most pertinent features for each output. The estimator used to score features have a significant impact on the resulting model in terms of performances, while the feature selection method impact the number of features kept. For RF, SVR using RFE as feature selection with the LASSO method with cross validation as estimator offers the most interesting models, a priori, since it has a low error while maintaining a higher R^2 than the other methods. For NN, usign RFE with SVRL provides the best results in terms of R^2 and RMSE.

Common features between the different methods tested and the different estimators used. The values of the tensiometers and the future precipitation are always selected. This seems obvious, but it is interesting to note that these feature selection methods validate these features as values that are crucial for learning process. Then, come the temperature values and the age of the plant. The Age feature was not necessarily expected, but indeed, there is a correlation between the age of the plant and the increase in Water Pressure (the greater the age, the more developed the plant is and the greater its water needs). These observations matches with the correlation matrix 1.

Performance comparison. The Fig. 5 shows the boxplots colorized according to the machine learning algorithm used (RF, SVR and NN) and grouped according feature selection method. Each boxplot of a group corresponds respectively to :

- Without a selection method, *i.e.* using all available features (None).
- Recursive Feature Elimination method (RFE).
- Filter Method (FM).

The best set of parameters of each feature selection method according to the metrics has been used. For RF and SVR for RFE, the k parameter is 10 and for FM the *Threshold* parameter is 16. Both of them use LASSO as estimator. For NN, for RFE the k parameter is 20 and for FM the *Threshold* parameter is 25. The estimator used for RFE and FM is SVRL First of all, within a learning method (with or without feature selection), apart from SVR, the means are quite similar whatever the metrics. For MAE and RMSE, maximum errors are lower (which is especially the case for NN) when the feature selection is used. In contrast, the minimum values of R^2 are higher. This is an interesting point since it shows that the lighter models are able to do as well than the model with all feature. In addition, SVR's performance is significantly improved when the feature selection is used. Now, If we compare machine learning algorithms with each other, when no feature selection is used RF has the best average performance. NN, however, has a higher minimum R^2 value. But when the feature selection is used, SVR has similar performance to RF. NN has a higher mean error, but lower maximum values for MAE and RMSE. Dispersion is also lower. For R^2 , the minimums are higher for NN, than RF.

Table 2: Lower is better. Details of the performance of the leave-two-out cross validation for Neural Network in front of different feature selection method.

	NN	NN_FM	NN_RFE
Mean	7.94	8.14	7.73
Minimum	1.15	1.15	1.30
Maximum	27.64	27.64	27.13
25 %	4.32	4.11	4.54
50 %	6.65	7.02	6.40
75 %	10.01	10.18	9.78

Tables 2, 3 and 4 details respectively the performance of Figs 4, 2 and 3. The results confirm the analysis of the boxplots.

Now if we disaggregate the overall performance in order to detail the evolution of the error according to the number of days to be predicted for each depth (P1, P2 and P3), Figs 6, 7 and 8 show this evolution according the same metrics (MAE, RMSE and R^2). Here, The same hyper parameters are kept. Thus the graphs show another point of view. For RF, SVR and

Table 3: Lower is better. Details of the performance of the leave-two-out cross validation for Random Forest in front of different feature selection method.

	RF	RF_FM	RF_RFE
Mean	6.84	6.86	6.84
Minimum	0.28	0.29	0.28
Maximum	32.15	32.04	32.15
25%	2.25	2.28	2.24
50%	4.35	4.29	4.39
75%	9.18	9.08	9.36

Table 4: Lower is better. Details of the performance of the leave-two-out cross validation for Support Vector Regressor in front of different feature selection method.

	SVM	SVM_FM	SVM_RFE
Mean	6.86	6.88	6.85
Minimum	0.27	0.28	0.28
Maximum	32.76	32.10	32.76
25%	2.24	2.23	2.25
50%	4.33	4.40	4.36
75%	9.27	9.33	9.03

Performance comparison

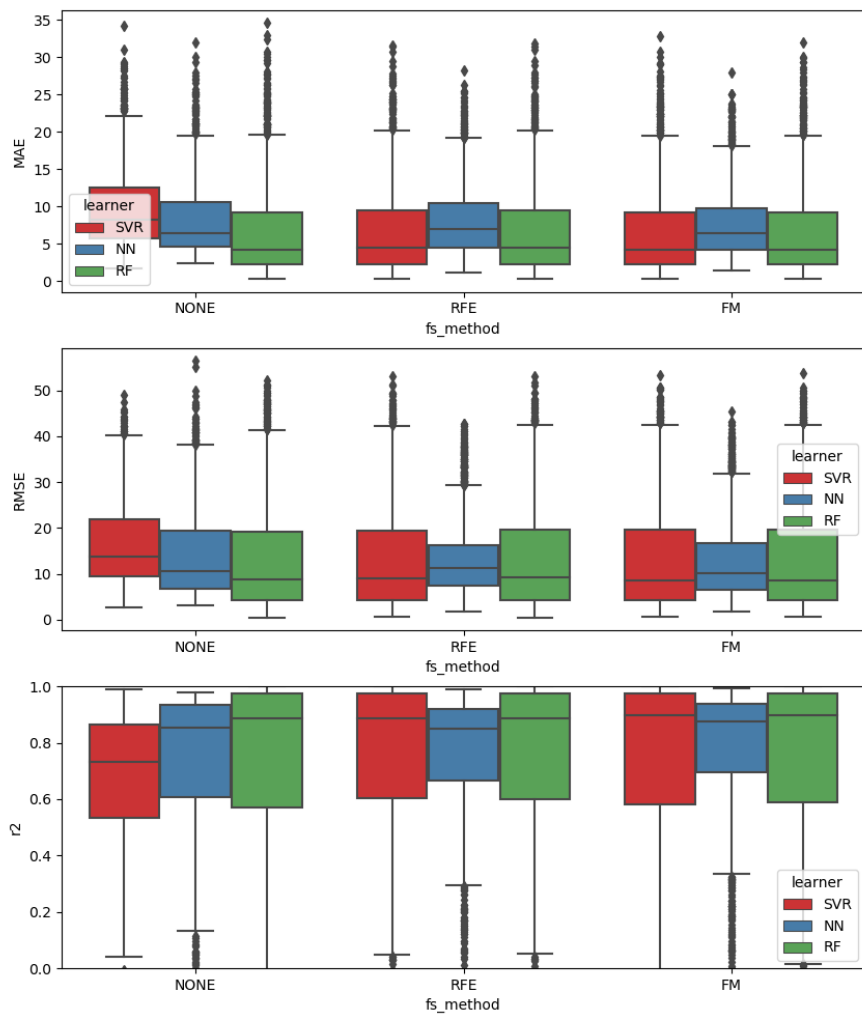


Figure 5: Cross validation error for each feature selection approach. Learning is done using the best set of hyper parameters for each method. Except for R^2 , lower is better.

RF

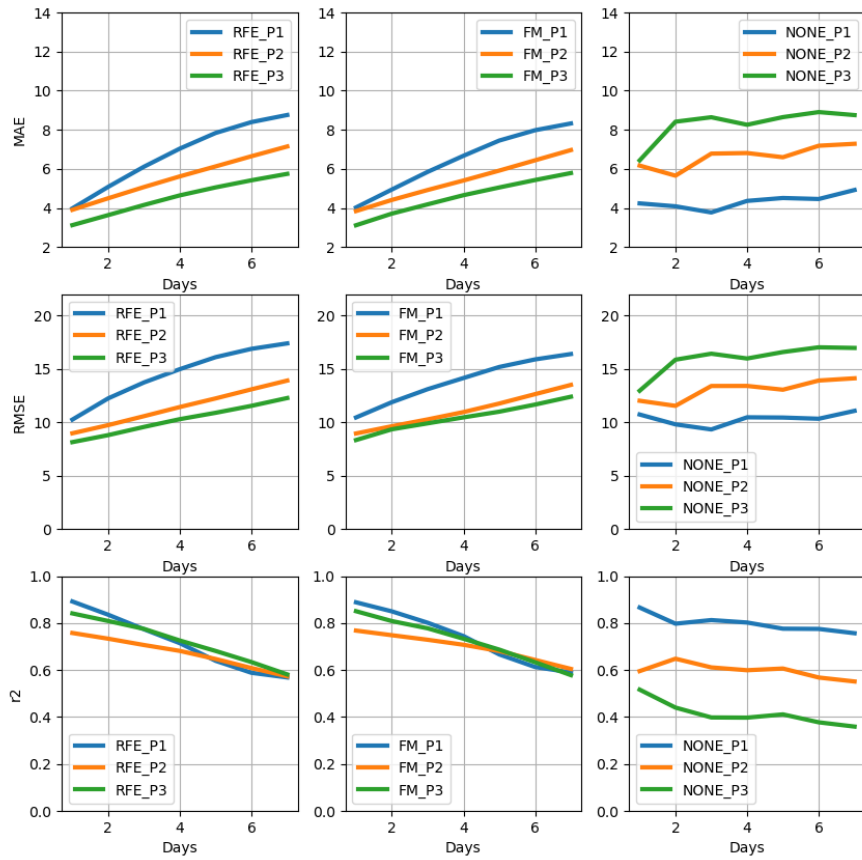


Figure 6: Evolution of the metrics according the number of day to predict in front of different method of feature selection method or not. Except for R^2 lower is better.

SVR

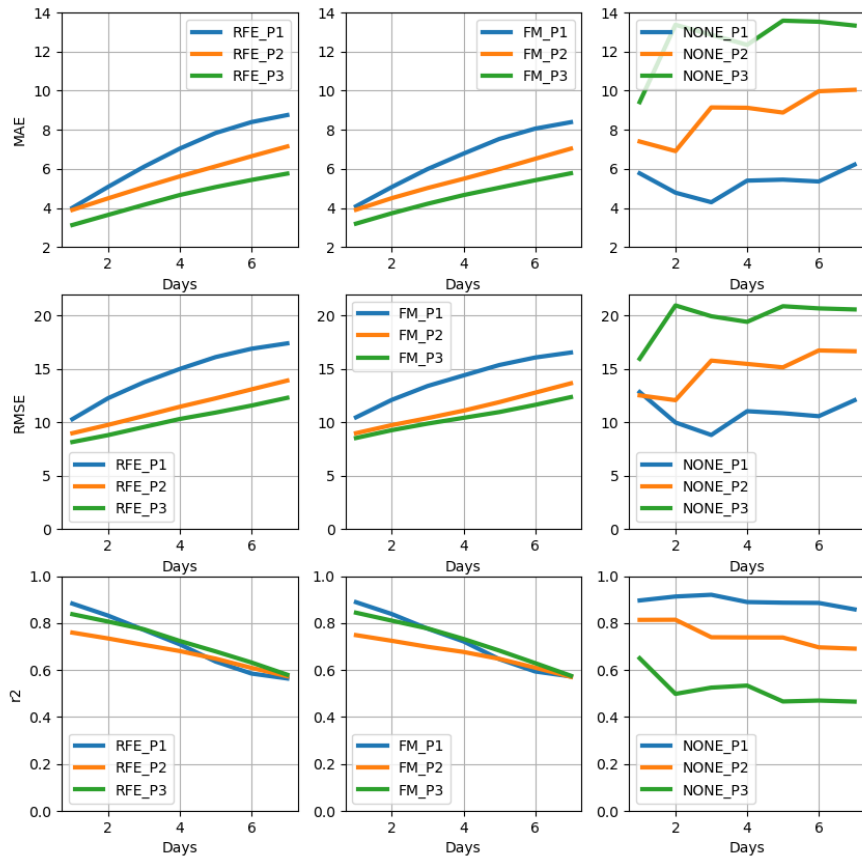


Figure 7: Evolution of the metrics according the number of day to predict in front of different method of feature selection method or not. Except for R^2 lower is better.

NN

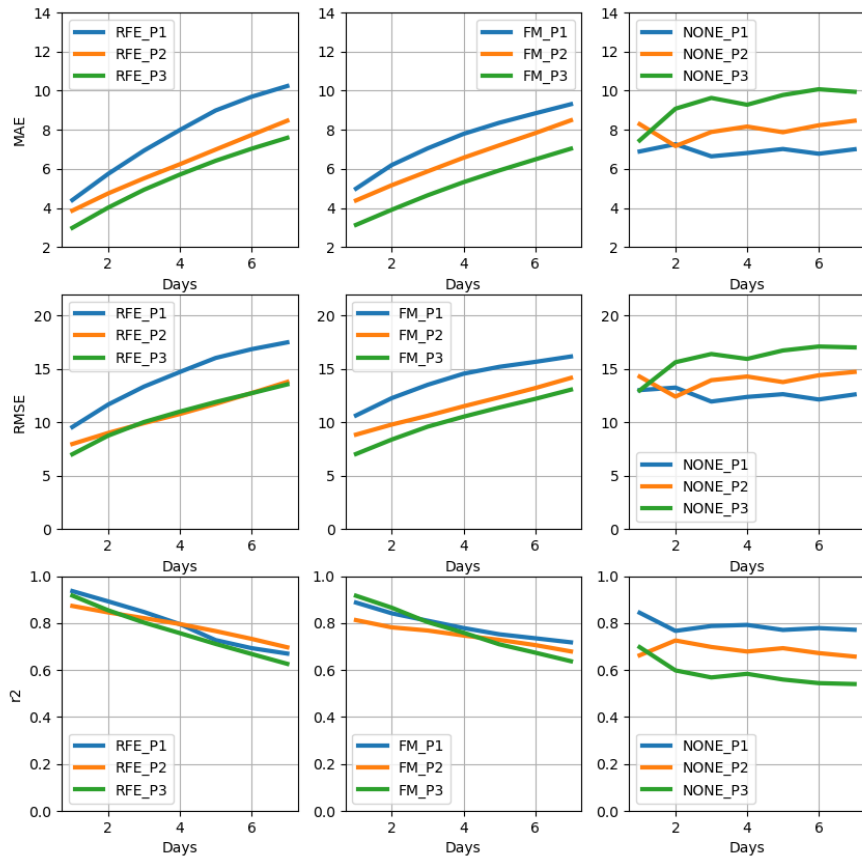


Figure 8: Evolution of the metrics according the number of day to predict in front of different method of feature selection method or not. Except for R^2 lower is better.

NN, predict 1 day ahead achieve great performance with R^2 score above 0.92 whatever the feature selection method used. This accuracy is equivalent to the performance of other studies on evapotranspiration (Soil Water Potential here). When the number of days to be predicted increases the performance remains correct up to 4 days and then deteriorates beyond that. At 7 days the error increases sharply, but remains correct. However the R^2 score collapses, which indicate that the prediction seems more uncertain. If we focus on the different depths, we notice that the evolution of the error is not the same according to the depth. At 20 cm the error increases faster than for 30 and 40 cm. This can be explained by a greater water dynamic at 20 cm than at 30 and 40 cm. Another possible explanation is that both the learning and test base has few number of observation with severe tensiometer values (close to 200 Kpa) at these depths. Eventually if we focus on feature selection method, the different algorithms have similar performances when the feature selection is used but different from the model with all features (NONE). It is noticeable that the latter learns the first depth (P1) better than the other two (P2 and P3) regardless of the number of days to predict, unlike models with feature selection which learns the different depths more equally. The R^2 scores are significantly less spread with the selection method (and better for P2 and P3) than with the none feature selection. This behaviour is confirmed with the other metrics. So if we take into account all depths, up to 5 days, it is preferable to use the model with feature selection (RFE or FM). Beyond that, if we are interest only by P1, it is more appropriate to use the all feature (NONE). These results are interesting since it highlights the fact that the global model (NONE) can focus on 1 depth (rather than the others) while the feature selection is able to efficiently select the most important features for each output since each output have the most relevant ones.

Discussion. The results are encouraging as they show that feature selection methods are suitable to generate such high-performance models. More, the automatic lightweight model design process is relevant in view of the encouraging results since the differences are (at least) quite small compare to the model with all feature, more the spread out is lower. It is also interesting to point out that the models have less discrepancies regarding performance between depths (whatever the metrics).

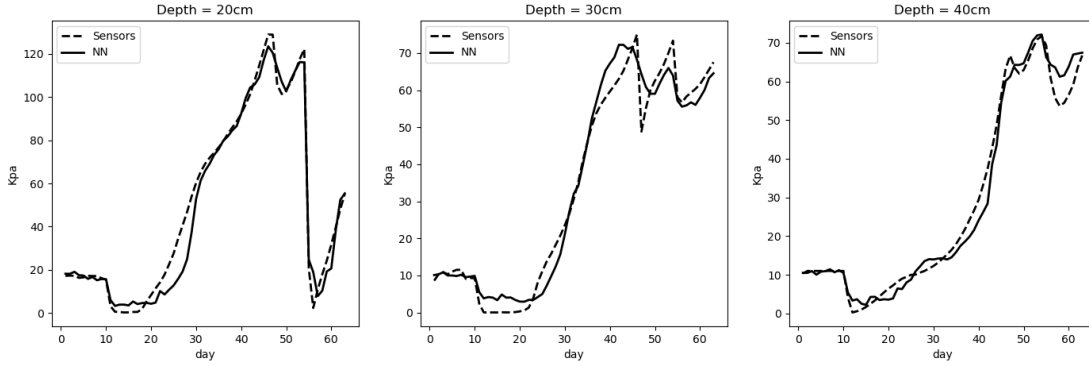


Figure 9: 7th-day forecasting over a cloudy campaign

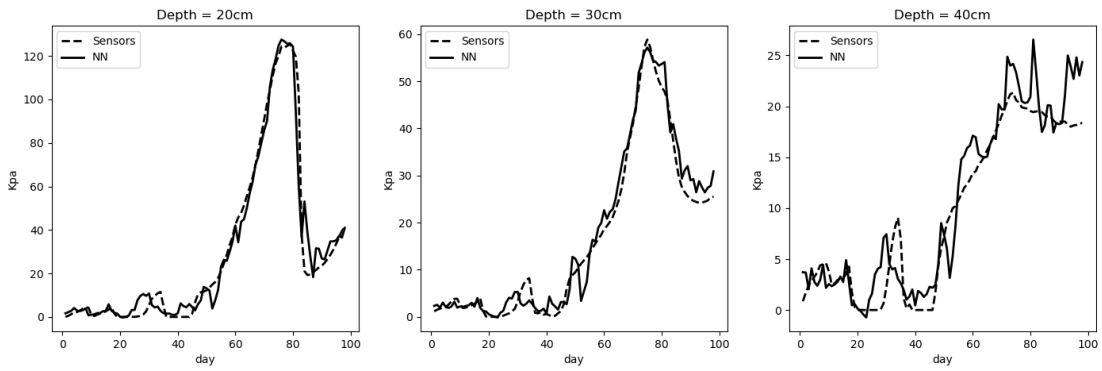


Figure 10: 7th-day forecasting over a rainy test campaign

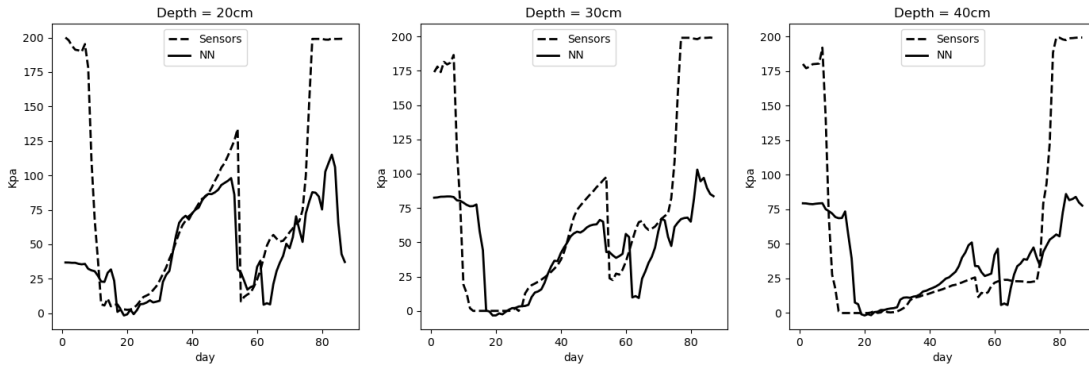


Figure 11: 7th-day forecasting over a hard drought test campaign

5.3. Short-term weekly soil moisture predictions

Irrigation tool. Since NN is the learning method with the lowest extreme errors (max bound), we will select it. Indeed, on land of several hectares, the variation of the SWP between 2 areas of the plot can then be higher than 10 kpa (i.e higher than the error made by the algorithms). Thus it is preferable to choose an algorithm that behaves better when the prediction conditions are unfavourable i.e. to predict global trends as accurately as possible and with reduced extreme errors. Figs. 9, 10, and 11 show some output of our predictive tool, representing a complete test campaigns using neural network learning algorithm. These curves represent a solution proposed to grower to address the problem of weekly SWP prediction. The black line illustrates the tensiometer values while the black dotted line represents the 7th-day forecast values by our neural network, in other words the most difficult long range prediction. Fig. 9 is the field corresponds to the campaign on which our model had the best performances. Here, visually the 7-th Day forecast over the whole campaign is very close to the ground truth. Fig. 10 represents a campaign where rainfall was abundant. On the first 2 depths (20 and 30 cm) the performances are also more than correct. The predictions at the last depth seem more chaotic but one must take into account the scale of the graph. Indeed here, the error is about 5 Kpa, which is not significant for field areas of several hectares. Eventually, fig. 11 correspond to the field with the worst performance. These campaigns are affected by dry spell and the soil was often drought. The learning databases include few samples covering these conditions. This explain the relative poor forecasting, in extreme conditions, in particular at the beginning (before 20 days) and at the end (after 70 days) of the campaign. The fall of Kpa that occurred around day 64 was due to the rain that the rain gauge measured but the sensors in the ground did not see it. One possible explanation is that the rain sensor was located at a certain distance from the plot, so rain could have really happened but outside the plot.

Discussion. As report on figures, the model (neural network here) is able to learn the trends of each campaign, even if the prediction quality can be heterogeneous. These figures provide a visualization of the prediction model. Eventually the proposed tool will present this type of information to the planter in order to give them a trend for the coming days.

6. Conclusion

In this paper, we model the problem of Soil Water Potential into a supervised learning problem. The problem is not trivial since there 21 outputs to predict (3 depths for each day and 7 days to predict) In addition, we use feature selection methodology in order to automatically design lighter models using the most suitable features. This work is a step forward, as we directly work with farmers, and the goal is to help them in decision making process, in an economical and ecological points of view. We use the information provided by low-cost tensiometers in order to make its use as democratic as possible.

As a result, the leave-two-out cross validation, allows us to present statistically significant results. The similar performance between the feature selection methods indicates that a lighter model can achieve close performance, which is an important point. In addition, if new features are introduced this method will automatically reject the least significant features and therefore be more robust. However, these methods require the tuning of the hyper parameters. About the methods, using a machine learning approach is relevant. With a larger database we will be able to improve the performance of algorithms.

For future work, we will increase the number of fields monitored and thus increase the current database. Few shot learning algorithms are also a way to explore since they are able to learn from a (very) limited number of examples. The interest of this approach is a good compromise compared to parametric models, or expensive destructive sampling. However, there is still room for improvements, with more data we could test current and new approaches on other type of soil, kind of crops or other weather. Since feature selection methods have an important impact on the model's design process, it would be pertinent to explore other approaches. Using a time series approach such as recurrent networks (RNN) or Long Short Time Memory (LSTM) could also be convenient too [55]. Their ability to learn from natural symmetries, patterns or other similarities related to temporal phenomena should improve current performance. In these case, a performance comparison with other temporal approach like ARIMA (the next value is predicted using previous values an autoregressive model, and a moving average model) method could also be investigated.

Declaration of Competing Interest

This work was supported by WEENAT company, who's funding Amaury DUBOIS's thesis.

The other authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank the WEENAT company for the use of their data which made this article possible. Experiments presented in this paper were carried out using the CALCULCO computing platform, supported by SCOSI/ULCO (Service COMMun du Système d'Information de l'Université du Littoral Côte d'Opale).

References

- [1] F. F. Montesano, F. Serio, M. Carlo, S. Angelo, P. Angelo, S. Pietro, Tensiometer-based irrigation management of subirrigated soil-less tomato: Effects of substrate matric potential control on crop performance, *Frontiers in Plant Science* 6 (2015) 1150.
- [2] Z. Zhai, J. F. Martínez, V. Beltran, N. L. Martínez, Decision support systems for agriculture 4.0: Survey and challenges, *Computers and Electronics in Agriculture* 170 (2020) 105256.
- [3] M. Francesco, Parente, L. Nicola, T. Mladen, Trotta, Modern technologies, strategies and tools for sustainable irrigation management and governance in Mediterranean Agriculture, 2015.
- [4] M. D. Dukes, L. Zotarelli, K. T. Morgan, Use of irrigation technologies for vegetable crops in florida, *HortTechnology* 20 (1) (2010) 133–142.
- [5] FAO, Sprinkler irrigation.
- [6] eos, soil moisture control is an essential farming constituent (2019).

- [7] K. T. Kassaye, J. Boulange, V. T. Lam, H. Saito, H. Watanabe, Monitoring soil water content for decision supporting in agricultural water management based on critical threshold values adopted for andosol in the temperate monsoon climate, *Agricultural Water Management* 229 (2020) 105930. doi:<https://doi.org/10.1016/j.agwat.2019.105930>.
- [8] M. T. Van Genuchten, A closed-form equation for predicting the hydraulic conductivity of unsaturated soils 1, *Soil science society of America journal* 44 (5) (1980) 892–898.
- [9] D. C. Erbach, Measurement of soil bulk density and moisture, *Transactions of the ASAE* 30 (4) (1987) 922–0931.
- [10] F. S. Zazueta, J. Xin, Soil moisture sensors, *Soil Sci* 73 (1994) 391–401.
- [11] M. Thalheimer, Tensiometer modification for diminishing errors due to the fluctuating inner water column, *Soil Science Society of America Journal* 67 (3) (2003) 737–739.
- [12] K. Johnson, S. Johnson, P. Teng, Development of a simple potato growth model for use in crop-pest management, *Agricultural Systems* 19 (3) (1986) 189 – 209.
- [13] N. Brisson, C. Gary, E. Justes, R. Roche, B. Mary, D. Ripoche, D. Zimmer, J. Sierra, P. Bertuzzi, P. Burger, F. Bussi ere, Y. Cabidoche, P. Cellier, P. Debaeke, J. Gaudill ere, C. H enault, F. Maraux, B. Seguin, H. Sinoquet, An overview of the crop model stics, *European Journal of Agronomy* 18 (3) (2003) 309 – 332, modelling Cropping Systems: Science, Software and Applications.
- [14] D. Raes, P. Steduto, T. C. Hsiao, E. Fereres, Aquacrop—the fao crop model to simulate yield response to water: Ii. main algorithms and software description, *Agronomy Journal* 101 (3) (2009) 438–447.
- [15] E. Ramat, B. Vandoorne, Plant growth model for decision making support, Tech. rep., Universit e du Littoral C ote d’Opale, and ISA Lille (2002).
- [16] R. G. Allen, L. S. Pereira, D. Raes, M. Smith, et al., Crop evapotranspiration-guidelines for computing crop water requirements-fao irrigation and drainage paper 56, Fao, Rome 300 (9) (1998) D05109.

- [17] T. Howell, S. Evett, The penman-monteith method (01 2004).
- [18] D. Chen, Y. Ma, Optimized algorithm for estimating parameters by solving van genuchten equation based on stochastic particle swarm optimization, *Nongye Gongcheng Xuebao*(Transactions of the Chinese Society of Agricultural Engineering) 22 (12) (2006) 82–85.
- [19] S. Sorooshian, Q. Duan, V. K. Gupta, Calibration of rainfall-runoff models: Application of global optimization to the sacramento soil moisture accounting model, *Water Resources Research* 29 (4) (1993) 1185–1194. arXiv:<https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/92WR02617>.
- [20] A. Dubois, F. Teytaud, E. Ramat, S. Verel, Automatic calibration of a farm irrigation model: a multi-modal optimization approach, in: L. Idoumghar, P. Legrand, A. Liefoghe, E. Lutton, N. Monmarché, M. Schoenauer (Eds.), *Artificial Evolution*, Springer International Publishing, Mulhouse, France, 2019, pp. 192–204.
- [21] H. Chen, J. J. Huang, E. McBean, Partitioning of daily evapotranspiration using a modified shuttleworth-wallace model, random forest and support vector regression, for a cabbage farmland, *Agricultural Water Management* 228 (2020) 105923.
- [22] Y. Feng, N. Cui, D. Gong, Q. Zhang, L. Zhao, Evaluation of random forests and generalized regression neural networks for daily reference evapotranspiration modelling, *Agricultural Water Management* 193 (2017) 163 – 173.
- [23] M. Kumar, N. Raghuwanshi, R. Singh, W. Wallender, W. Pruitt, Estimating evapotranspiration using artificial neural network, *Journal of Irrigation and Drainage Engineering* 128 (4) (2002) 224–233.
- [24] I. Ali, F. Greifeneder, J. Stamenkovic, M. Neumann, C. Notarnicola, Review of machine learning approaches for biomass and soil moisture retrievals from remote sensing data, *Remote Sensing* 7 (12) (2015) 16398–16421. doi:10.3390/rs71215841.
- [25] R. Torres-Sanchez, H. Navarro-Hellin, A. Guillamon-Frutos, R. San-Segundo, M. C. Ruiz-Abellón, R. Domingo-Miguel, A decision support system for irrigation management: Analysis and implementation of different learning techniques, *Water* 12 (2) (2020). doi:10.3390/w12020548.

- [26] R. Zakaluk, R. S. Ranjan, Artificial neural network modelling of leaf water potential for potatoes using rgb digital images: a greenhouse study, *Potato Research* 49 (4) (2006) 255–272.
- [27] D. Delgoda, S. K. Saleem, H. Malano, M. N. Halgamuge, Root zone soil moisture prediction models based on system identification: Formulation of the theory and validation using field and aquacrop data, *Agricultural Water Management* 163 (2016) 344 – 353.
- [28] J. G. Fortin, F. Anctil, L.-É. Parent, M. A. Bolinder, A neural network experiment on the site-specific simulation of potato tuber growth in eastern canada, *Computers and Electronics in Agriculture* 73 (2) (2010) 126–132.
- [29] A. Sabziparvar, H. Tabari, Comparison of artificial neural network models and non-linear regression methods for estimation of potato crop evapotranspiration in a semi-arid region of iran, in: *The international conference on intelligent network and computing*, Nov, 2010, pp. 26–28.
- [30] H. Tabari, C. Martinez, A. Ezani, P. H. Talaei, Applicability of support vector machines and adaptive neurofuzzy inference system for modeling potato crop evapotranspiration, *Irrigation science* 31 (4) (2013) 575–588.
- [31] S. S. Yamaç, M. Todorovic, Estimation of daily potato crop evapotranspiration using three different machine learning algorithms and four scenarios of available meteorological data, *Agricultural Water Management* 228 (2020) 105875.
- [32] R. Filgueiras, T. S. Almeida, E. C. Mantovani, S. H. B. Dias, E. I. Fernandes-Filho, F. F. da Cunha, L. P. Venancio, Soil water content and actual evapotranspiration predictions using regression algorithms and remote sensing data, *Agricultural Water Management* 241 (2020) 106346. doi:<https://doi.org/10.1016/j.agwat.2020.106346>.
- [33] M. Y. Chia, Y. F. Huang, C. H. Koo, Support vector machine enhanced empirical reference evapotranspiration estimation with limited meteorological parameters, *Computers and Electronics in Agriculture* 175 (2020) 105577.
- [34] R. Linker, I. Ioslovich, G. Sylaios, F. Plauborg, A. Battilani, Optimal model-based deficit irrigation scheduling using aquacrop: A simulation

- study with cotton, potato and tomato, *Agricultural Water Management* 163 (2016) 236–243.
- [35] C. D. van Loon, The effect of water stress on potato growth, development, and yield, *American Potato Journal* 58 (1981).
 - [36] P. J. Werbos, *The roots of backpropagation: from ordered derivatives to neural networks and political forecasting*, Vol. 1, John Wiley & Sons, 1994.
 - [37] P. Werbos, *Beyond regression:” new tools for prediction and analysis in the behavioral sciences*, Ph. D. dissertation, Harvard University (1974).
 - [38] Nitish Srivastava and Geoffrey Hinton and Alex Krizhevsky and Ilya Sutskever and Ruslan Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research* 15 (2014) 1929–1958.
 - [39] L. Breiman, J. Friedman, R. Olshen, C. Stone, *Classification and regression trees* (1984).
 - [40] T. K. Ho, Random decision forests, in: *Proceedings of 3rd international conference on document analysis and recognition*, Vol. 1, IEEE, 1995, pp. 278–282.
 - [41] scikit learn, random forest regressor (2020).
 - [42] V. N. Vapnik, *The nature of statistical learning theory* (1995).
 - [43] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, V. Vapnik, Support vector regression machines, in: *Advances in neural information processing systems*, 1997, pp. 155–161.
 - [44] V. Vapnik, *The nature of statistical learning theory*, Springer science & business media, 2013.
 - [45] J.-Y. Lin, C.-T. Cheng, K.-W. Chau, Using support vector machines for long-term discharge prediction, *Hydrological Sciences Journal* 51 (4) (2006) 599–612.

- [46] Q. Li, Q. Meng, J. Cai, H. Yoshino, A. Mochida, Applying support vector machine to predict hourly cooling load in the building, *Applied Energy* 86 (10) (2009) 2249–2256.
- [47] scikit learn, Feature selection (2020).
- [48] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, *Journal of machine learning research* 3 (Mar) (2003) 1157–1182.
- [49] scikit learn, Recursive feature elimination (2020).
- [50] J. You, S. A. [van der Klein], E. Lou, M. J. Zuidhof, Application of random forest classification to predict daily oviposition events in broiler breeders fed by precision feeding system, *Computers and Electronics in Agriculture* 175 (2020) 105526.
- [51] F. Löw, U. Michel, S. Dech, C. Conrad, Impact of feature selection on the accuracy and spatial uncertainty of per-field crop classification using support vector machines, *ISPRS journal of photogrammetry and remote sensing* 85 (2013) 102–119.
- [52] C. Chu, A.-L. Hsu, K.-H. Chou, P. Bandettini, C. Lin, A. D. N. Initiative, et al., Does feature selection improve classification accuracy? impact of sample size and feature selection on classification using anatomical magnetic resonance images, *Neuroimage* 60 (1) (2012) 59–70.
- [53] H. X. Zhao, F. Magoules, Feature selection for support vector regression in the application of building energy prediction, in: 9th IEEE International Symposium on Applied Machine Intelligence and Informatics (SAMI 2011), IEEE Computer Society, Smolenice, Slovakia, 2011, pp. –.
- [54] P. M. Granitto, C. Furlanello, F. Biasioli, F. Gasperi, Recursive feature elimination with random forest for ptr-ms analysis of agroindustrial products, *Chemometrics and Intelligent Laboratory Systems* 83 (2) (2006) 83 – 90.
- [55] A. Kamilaris, F. X. Prenafeta-Boldú, Deep learning in agriculture: A survey, *Computers and Electronics in Agriculture* 147 (2018) 70 – 90.