Comparing machine-learning models of different levels of complexity for crop protection: A look into the complexity-accuracy tradeoff

Olivier Gauriau, Luis Galárraga, François Brun, Alexandre Termier, Loïc Davadan, François Joudelat

Smart Agricultural Technology 7 (2024) 100380

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

Comparing machine-learning models of different levels of complexity for crop protection: A look into the complexity-accuracy tradeoff

Olivier Gauriau [a,*], Luis Galárraga [a], François Brun [b], Alexandre Termier [a], Loïc Davadan [c], François Joudelat [d]

[a] Univ. Rennes, Inria, Irisa, France
[b] ACTA, INRAE, UMR AGIR, Castanet Tolosan, France
[c] IFV, UMT SEVEN, Cognac, France
[d] ITB, Paris, France

ARTICLE INFO

Editor: Spyros Fountas

ABSTRACT

Crop diseases and pests constitute significant causes of yield losses for crops. To limit the harm incurred by those events, farmers resort to plant protection products. Such products are known to have adverse effects both on the environment and on human health. Agronomists make continuous efforts to limit the usage of plant protection products to situations where those products are strictly necessary. To determine such situations, agronomists and policy-makers often rely on decision support tools to model and predict the dynamics of plant diseases. Decision support tools are based either on mechanistic models or on statistical approaches learned from large datasets of biotic (e.g., disease incidence, plant phenological stage) and abiotic (meteorological, soil characteristics) observations in cultures. The surge of powerful machine learning (ML) methods in the last decade makes such approaches a natural pathway to model the dynamics of plant diseases.

Machine learning models can reveal the factors that contribute the most to disease and pests outbreaks, provided that those models are simple enough for human inspection. Simplicity, however, may come at the price of lower prediction performances when compared to more complex models.

In this paper, we offer a deep look at the performance of ML models of different complexity when used on two use cases of crop disease prediction: downy mildew in the grapevine, and Cercospora leaf spot in the sugar beet. We compare model accuracy and complexity using a year-based cross-validation approach. Our results suggest that interannual meteorological variations are a very important factor in plant disease prediction. Moreover, in line with the observations of the research community in interpretable ML, model complexity stands in clear trade-off with accuracy. This makes models of intermediate complexity appealing for predicting the dynamics of crop diseases as they can provide explicit insights about the rationale of their predictions.

1. Introduction

Crop protection against plant diseases is crucial to secure crop yields. To this end, farmers and agronomists make use of plant protection products, i.e., pesticides, to combat plant diseases and pests in cultures. It is well-known, however, that the usage of such products has multiple downsides. Besides their impact on farmers' health, and their polluting effect on the environment, such products incur an economic cost on both farmers and consumers, not to mention their role in the development of pesticide-resistant breeds [1] and the indirect contamination in other stages of the food supply chain [2]. It follows that minimizing the usage of pesticides in cultures incurs countless benefits. One way to reduce our dependence on such products is to adapt their usage to local factors [3] such as the climate/weather, the soil type, or the farming practices. This can be achieved through the deployment of models that can predict disease incidence or risk of outbreak. Such tools help farmers and agronomists avoid the usage of pesticides when they are not necessary.

There have been multiple efforts to model and predict the risk of outbreak and the incidence of plant diseases in cultures [4]. Existing methods can be categorized into two families. On the one hand, mechanistic models are constructed based on prior observations and knowledge of the diseases or pests' life cycles. These models require

* Corresponding author.
E-mail address: olivier.gauriau@irisa.fr (O. Gauriau).

https://doi.org/10.1016/j.atech.2023.100380
Received 28 August 2023; Received in revised form 6 December 2023; Accepted 9 December 2023
Available online 18 December 2023

extensive agronomical studies and experts intervention, and were the preferred approach to model plant diseases for long time. An inflexion point arose with the emergence of large amounts of data including past observations of diseases in cultures – as human annotations or as images –, but also information about abiotic factors such as the characteristics of the soil and meteorological data. This data abundance has made statistical models, in particular machine learning models, more appealing in the last decade, and has nurtured their steady increase in accuracy and sophistication.

ML models used in crop protection are usually trained for a single type of crop and disease. This is due to the fact that different crops develop in different ways, and so do diseases and pests. It is also known that models are typically trained for a given region, and are less accurate when used on data from other regions [5]. Some approaches rely on image classification with deep learning [6–9] for disease diagnosis. Other models are designed to predict or forecast the incidence of a disease at a particular period of the year, e.g., before harvest, based on human annotations. This forecast can take the form of an incidence prediction (regression) or a risk of outbreak (classification) [3]. In those cases the models' outcomes help agronomists decide whether to apply or not plant protection products in their cultures.

But besides forecasting the incidence of crop diseases, ML models can also help agronomists understand which factors contribute to the development of those diseases. This is possible, however, if the model is simple and interpretable enough to be understood by humans. Examples of interpretable ML models are linear functions and shallow decision trees.

A simple, yet effective proxy to model interpretability is model complexity [10]. Complexity is usually measured as the number of relevant parameters that play a role in the model's answers, and it is known to be correlated with interpretability. To see why, it suffices to compare the effort of interpreting a linear model with 5 variables versus a linear model with 300 variables.

While complex models such as neural networks or gradient boosting tend to be less interpretable than transparent simple methods such as linear regression or shallow trees, in some cases this complexity pays off in terms of prediction performance [11,12,10].[1] This trade-off between complexity and prediction accuracy can happen because more parameters or weaker assumptions endow models with more expressiveness and flexibility to capture subtle interactions. Simpler models make assumptions that may not encapsulate the complexity of real data. For example linear models assume there exists a linear relationship between the input features and the target variable i.e., the variable we want to model or predict. This, for instance, excludes any potential interactions between the input features as predictors for the target variable. Between complex approaches and simple models lie pattern-based models [10,15] that strike an interesting trade-off because they remain relatively simple white boxes that exhibit higher predictive power than linear regression or decision trees.

Existing works that use ML methods for crop protection have paid little attention to the potential needs for interpretability and the complexity-interpretability trade-off [16,7,9]. We therefore contribute to the state of the art by studying this trade-off in the context of crop protection. We train different popular machine learning models of varied complexity for two typical crop protection tasks: (i) disease incidence prediction, and prediction of the symptoms appearance date. We predict these target variables for the downy mildew in grapevine cultures, and for the Cercospora leaf spot in sugar beet crops, both in France. In both cases we resort to biotic (e.g., past disease incidences) and abiotic (e.g., meteorological data) predictors. Our tasks are classical regression problems, therefore the studied models include (i) black-box ensemble methods such as random forests and gradient boosting trees; (ii) white boxes such as linear regression; and (iii) HiPaR [10], a pattern-based regression method of intermediate complexity. Our experiments confirm a clear complexity-accuracy trade-off in our use cases, and also show different techniques to distill agronomical insights from both white- and black-box ML models. Our results suggest that despite the difference in prediction accuracy and model architecture, the models agree on some common insights. Moreover, interannual effects play a very important role, which makes very difficult to have a single model that can predict disease incidence for any arbitrary year.

Section 2 describes the datasets used for our study, the methods trained on those datasets as well as their performance. This is followed by a discussion of the different agronomical findings we extracted from the trained ML models in Section 3. Section 4 concludes the paper with avenues for future research in the prediction of disease incidence in cultures.

2. Material and methods

We now describe the agronomical datasets used in our study as well as the machine learning models trained on those datasets.

2.1. Data

Our study case builds upon four datasets covering two major plant diseases observed in French cultures: Grape downy mildew and Sugar beet Cercosporia.

2.1.1. Sugar beet Cercosporia epidemiologic data

Sugar beet Cercosporia (SBC) incidences were observed in several vineyards located in France by different extension services, including the ITB (Institut Technique de la Betterave). The experimental observations have been collected from 2009 to 2020 in different regions in France.

For each monitored site, a specific part of the area, further referred to as the "plot", was observed throughout a specific year. Weekly visual inspections were performed on leaves covering one hundred plants in order to assess disease incidence. The incidence was calculated as the proportion of sugar beet leaves displaying symptoms of Cercosporia leaf spot (*Cercospora beticola*). Weekly inspections were conducted in each plot from leaf emergence (which happens in mid-May) until harvest (after mid-September). The collected dataset adds up to 1235 individual plots. We highlight that no plot was observed every year, and that conversely, not all plots can be monitored in a single year.

For each plot, we define the date of SBC onset (yearly symptoms apparitions date) as the first day in which the proportion of infected leaf exceeded 10%. The end of season incidence for SBC was defined as the maximum incidence for the period going from the 25th of August to the 15th of September.

2.1.2. Grape downy mildew epidemiologic data

Grape downy mildew (GDM) incidence were observed in several vineyards located in France by different wine extension services including the IFV (Institut Français de la Vigne et du Vin). The data have been collected from 2010 to 2017.

For each considered plot, an untreated row of vines was observed. Each untreated row was surrounded by two other untreated rows to ensure that they were not unintentionally sprayed with fungicides. In the monitored central row, weekly visual inspections were performed on leaves in order to measure disease incidence. The incidence was calculated as the proportion of vine leaves displaying downy mildew symptoms caused by *Plasmopara viticola*. Weekly inspections were conducted in each vineyard from budburst (early March) until at least bunch closing (mid-late July) or stopped when the incidence was close to 100%. The observations consist of around 9407 weekly datapoints corresponding to 713 plots.

[1] As shown by [13,14], the accuracy-interpretability trade-off is not necessarily observed in every application domain and depends on multiple factors such as the quality of the data.

Table 1
Description of the meteorological variables used to model the dynamics of the Sugar beet Cercosporia (SBC). Temperatures are considered as *inhibiting* below 10 °C or above 38 °C.

Name	Feature
RHmX	Mean Relative Humidity lower than X (X = {60, 65, 80, 90})
H87	Humidity index equals to 87
H87Y	Humidity index equals to 87 for at least (Y = {6, 10}) hours
TmX	Mean Temperature higher than (X = {15, 20})
TmXTinfYZ	Mean Temperature higher than (X = {15}) but lower than (Y = {10}) for at least (Z = {3}) hours
TblоX	Number of days where temperatures were defined as *inhibiting* to SBC growth for more than (X = {3,6}) hours.

For each plot, date of GDM onset (yearly symptoms apparitions date) was defined as the first week in which the proportion of infected vines leaf exceeded 1%. The end of season incidence for GDM was defined as the maximum incidence for each plot.

2.1.3. Meteorological data

Meteorological variables were provided by the SAFRAN weather database constructed and maintained by the French national meteorological service (Météo-France). SAFRAN organizes the French territory into a grid of size 8×8 Km and stores meteorological data for each cell in the grid [17]. Daily observations on humidity, mean temperature, wind, amount of rainfall, and solar radiation were used to compute different meteorological variables for both diseases.

For SBC, each meteorological variable covers a period of half a month (15 days) from January to June. Features in the dataset follow a given convention. The first part describes the temporal characteristics of the feature with the first three letters of the corresponding month, followed by an 'A' for the first half of a month or a 'B' for the second half. The second part describes the climatic nature of the feature and how this information was calculated. The feature suffixes are described in Table 1. For example, the variable named *JanA-ndRHm60* corresponds to the number of days (**nd**) such that the relative humidity was higher than 60 percent (**RHm60**) during the first half (**A**) of January (**Jan**).

For GDM, features either describe meteorological conditions at the date of recording or its sum for the four previous weeks before recording. For example, the predictive variable ETP gives us the evapotranspiration at the time of recording. ETP-4w is the sum of evapotranspiration for the four previous weeks. Two exceptions are the number of rainy and dry days, which are counted from the beginning of January. This length of four weeks was chosen based on expert insights about the growth speed of downy mildew.

2.1.4. Four prediction targets

From both diseases data and associated climatic variables, we finally obtained 4 data sets corresponding to our 4 prediction targets.

- Sugar beet Cercosporia (SBC) end of season incidence (% of leaves with diseases) with 1235 plots and 367 variables including one categorical variable and 366 numeric ones. The categorical feature is the *risk-exposure*, an indicator defined by agronomists based on their own knowledge of each plot's sensitivity to SBC. The numerical variables correspond to the one described in Subsection 2.1.3.
- Sugar beet Cercosporia (SBC) symptoms appearance date (day number of year) with 1235 plots and 367 variables.
- Grape downy mildew (GDM) end of season incidence (% of sick leaves) with 359 plots and 22 variables including two categorical and 20 numeric.
- Grape downy mildew (GDM) symptoms appearance date (week number of year) with the same 359 plots and 22 variables.

Thus, the target variables are numerical. We are thus confronted to a regression problem in all cases.

2.2. Regression methods

We assume that the goal is to predict the values of a real variable, that we call the *target variable*, using observations from another set of variables that we call the *predictive variables*. Examples of target variables are given in Subsection 2.1.4. Conversely, the predictive variables constitute the set of meteorological indicators (see Table 1). This scenario constitutes a classical regression problem. We first introduce some notation and then survey the most popular regression methods used in crop protection on tabular data. We extend the discussion with the description of a pattern-aided regression method that deals with the complexity-accuracy trade-off introduced in previous sections.

2.2.1. Problem formulation and notation

Let us assume that we count on a set of n target observations represented as a column vector $\boldsymbol{y} \in \mathbb{R}^n$. Those target observations are associated to a set of observations on the predictive variables, organized in a matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$. Each row $\boldsymbol{x}_i^\top \in \mathbb{R}^d$ in the matrix stores the observed values of the d predictive variables associated to a target observation y_i. From now on, we denote vectors and matrices with names in bold to distinguish them from scalars and functions. Moreover, matrices are denoted with capital letters. If a predictive variable is categorical, e.g., plant variety, we assume its values have been encoded as real numbers, for instance, by resorting to strategies such as one-hot encoding or dimensionality reduction.

The goal of regression analysis is to learn a function f such that $\boldsymbol{y} = f(\boldsymbol{X}) + \boldsymbol{\epsilon}$ and $\boldsymbol{\epsilon}$ is minimal. The function f is a model of the data designed to predict the target variable for unseen instances $\boldsymbol{x}^\top \in \mathbb{R}^d$ of the predictive variables. The term $\boldsymbol{\epsilon}$ is the error of the regression model and accounts for potentially unobserved predictors of $\boldsymbol{y}$. The model f is learned on a set of training and validation observations.

2.2.2. Classical regression methods

Linear regression This method assumes that the relation between the target variable $\boldsymbol{y}$ and the predictive variables $\boldsymbol{X}$ is linear, that is,

$$\boldsymbol{y} = \boldsymbol{\beta}\boldsymbol{X'} + \boldsymbol{\epsilon} \text{ with } \boldsymbol{\beta} = argmin_{\hat{\boldsymbol{\beta}}} ||\boldsymbol{y} - \boldsymbol{X'}\hat{\boldsymbol{\beta}}||_2^2 \tag{1}$$

$$\boldsymbol{\beta} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}, \tag{2}$$

where $\boldsymbol{X'} = \mathbb{1} \oplus \boldsymbol{X}$, i.e., $\boldsymbol{X'} \in \mathbb{R}^{n \times (d+1)}$ and $\boldsymbol{\beta} \in \mathbb{R}^{d+1}$ are the parameters of the model (the operator $\oplus$ denotes column concatenation), namely the linear coefficients associated to each of the d predictive variables plus the intercept coefficient β_0. The parameters of the model can be computed by minimizing the loss function $\mathcal{L}_l(\hat{\boldsymbol{\beta}}) = ||\boldsymbol{y} - \boldsymbol{X'}\hat{\boldsymbol{\beta}}||_2^2$ with the method of ordinary least squares (OLS) as illustrated in Equation (2). Linear models are among the most popular regression methods due to their simplicity and interpretability. This is because the magnitude of the coefficients tells us explicitly how much a predictive variable contributes to the model's prediction. On the downside, the linearity assumption may come at the expense of low prediction accuracy, which is why linear models are often used as baseline methods.

Lasso To reduce the risk of over-fitting in linear regression, Lasso [18] proposes an L1-regularization of the loss function, which favors mod-

els with few non-zero coefficients. This is achieved by minimizing the following objective:

$$\boldsymbol{\beta} = argmin_{\hat{\boldsymbol{\beta}}}\, \mathcal{L}_l(\hat{\boldsymbol{\beta}}) + \theta||\hat{\boldsymbol{\beta}}||_1. \quad (3)$$

By minimizing the L1-norm of $\boldsymbol{\beta}$ we can obtain sparse models that can not only prevent or mitigate over-fitting, but that are less complex and therefore easier to inspect by humans. The penalization term θ is a hyper-parameter that controls the importance of the sparsity constraint in the optimization process. The Lasso method selects the set of parameters $\hat{\boldsymbol{\beta}}$ that achieves the highest performance in cross-validation.

Decision/regression trees A decision tree is a binary tree where each internal node evaluates a Boolean condition on a predictive variable. The children of a node are decision trees associated to an evaluation outcome, i.e., true or false. Leaves (also called final nodes) are linked to a prediction of the model for the target variable. When the target variable is numerical, we talk about *regression trees* [19]. Regression trees are white-box models because the model's prediction on a particular instance $\boldsymbol{x}^\top \in \mathbb{R}^d$ can be explained by following the path from the root to the leaf node that predicts the outcome for $\boldsymbol{x}^\top$. This makes regression trees interpretable models, provided that the tree is not too deep for human inspection. Despite their interpretability, decision trees are prone to over-fitting if not properly parameterized, and are usually outclassed in terms of predictive performance by ensemble methods such as random forests and gradient boosting trees.

Random forests Random Forests are ensembles of weak decision tree estimators [20]. Predictions are computed by averaging the predictions of each tree in the ensemble. The weak estimators are learned by applying bagging and random feature selection. In bagging, each tree is learned by sampling from $\boldsymbol{X}$ and $\boldsymbol{y}$ uniformly and with replacement. Moreover, the trees are trained on different subsets of the features, which gives each tree a "partial" but "unique" view of the data. These techniques make random forests very robust to over-fitting, and a very popular choice for crop protection [21]. On the downside, random forests are not interpretable because the aggregation step makes it very difficult to trace the outcome of the model back to the input features – without resorting to post-hoc inspection approaches as we will show later.

Gradient boosting Another popular ensemble method is gradient boosting [22]. Like random forests, the basic principle is to compute a robust prediction from the predictions of a set of weak learners. Different from random forests, learning is based on an additive model where each learner h_m is fit on the error of the previous learner h_{m-1} – technically on the negative gradient of the minimized loss function. Put differently, each new learner is trained to correct the errors of the previous one:

$$f_m(\boldsymbol{X}) = f_{m-1}(\boldsymbol{X}) + \gamma_m h_m(\boldsymbol{X}) \quad (4)$$

$$\gamma_i = \mathcal{L}(\boldsymbol{y}, f_m(\boldsymbol{X})) \quad (5)$$

The individual learners can be of any type, however decision trees are a common choice [22]. Gradient boosting models are very robust to over-fitting, and like random forests, behave pretty much like black boxes.

2.2.3. Hierarchical pattern-aided regression (HiPaR)

Pattern-aided regression Pattern-based regression models consist of a set of local models trained on regions of the data. Those regions are characterized by interpretable patterns, namely logical conditions on the predictive variables, e.g., *wind-speed* > 50. The local models are usually interpretable functions, e.g., linear functions, that capture local relationships between the target and the predictive variables that cannot be observed at the "global level". As shown in the literature [10,15], these methods exhibit higher predictive performance than linear regression at the price of a manageable increase in complexity. Examples of pattern-aided regression methods include piecewise regression [23], regression trees [20], model trees [24],[2] Contrast pattern-aided regression (CPXR) [15], and HiPaR [10]. We elaborate on the latter method in the following.

HiPaR Hierarchical Pattern-aided Regression [10] estimates the values of the target variable via a compact set of local hybrid rules on the predictive variables. These rules have the form:

$$p = C_1 \wedge \ldots C_m \Rightarrow \boldsymbol{y} = f_p(\boldsymbol{X}_p). \quad (6)$$

In this expression, the pattern p is a conjunction of conditions on the predictive variables such as *wind-speed* > 50 ∧ *humidity* > 30. Those conditions define subsets or regions of the data $\boldsymbol{X}_p \subset \boldsymbol{X}$. A hybrid rule is associated to a local linear model f_p that has been trained on $\boldsymbol{X}_p$, and that refines the predictions of a global linear model f trained on $\boldsymbol{X}$. The model f, called the *default* model, is used to make predictions whenever none of the local hybrid rules applies. After having learned the default model, HiPaR mines a compact set of hybrid rules by means of two phases:

1. During the enumeration phase, the learning algorithm explores the space of patterns p in a depth-first hierarchical fashion. When a pattern p is visited, HiPaR learns a hybrid rule of the form $p \Rightarrow \boldsymbol{y} = f_p(\boldsymbol{X}_p)$ on $\boldsymbol{X}_p$ – the set of observations that satisfy p –, and then explores the sub-regions of $\boldsymbol{X}_p$. Since the search space is exponential in the number of features, a set of pruning strategies reduces it by avoiding the exploration of unpromising sub-regions; for example a minimum support threshold is enforced to avoid sub-regions with very few points.
2. Despite the pruning strategies carried out during the enumeration stage, the set of resulting hybrid rules can still be very large. For this reason, HiPaR carries out a selection phase that retains a small set of hybrid rules with good performance and minimal overlap. This phase is governed by two hyper-parameters: the support and the overlap bias. They determine, respectively, to which extent very specific rules are preferred over general rules, and how much overlap between the selected rules is allowed.

Contrary to tree-based models, HiPaR's hybrid rules are extracted from a hierarchy with potentially overlapping regions as depicted in Fig. 1. When a new observation $\boldsymbol{x}^\top$ satisfies more than one hybrid rule, the final prediction is the weighted average of the predictions of the individual rules. The weight is inversely proportional to the rule's error on a validation subset. This makes HiPaR models more robust than linear functions and regression trees, but significantly more complex. That said, HiPaR hybrid rules remain white-box models that allow for simple inspection of the most important predictive variables in the prediction for an observation $\boldsymbol{x}^\top \in \mathbb{R}^d$.

Table 2 summarizes the strengths and weakness of the methods discussed in this section.

2.3. Training and testing procedures

2.3.1. Optimization and performance evaluation

One of the challenges of evaluating different machine learning models is to select the best configuration so that comparisons are fair and meaningful. In an agronomical scenario the important interannual differences make standard cross-validation unadapted. Therefore, we use cross-validation by year, that is, each year is used as a fold in the process. The data from a given year is separated from the rest of the dataset for testing, whereas the observations from remaining years are used to

[2] These are regression trees such that some nodes, usually the leaves, are linear models on the target variable.

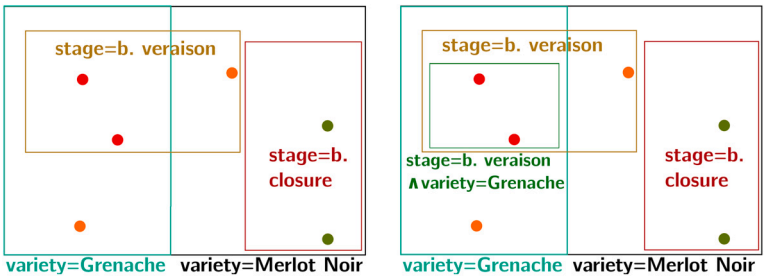


Fig. 1. A depiction of the regions explored by HiPaR for two steps of the enumeration phase. Each rectangle defines a region described by a pattern, on which HiPaR learns a local regression model. Regions can overlap; an example is the regions *stage* = "*b. veraison*" and *variety* = "*Grenache*". Once a region is explored, e.g., *stage* = "*b. veraison*", HiPaR will look at its sub-regions in a depth-first-search manner (figure on the right).

train the algorithm. That way we are able to estimate the actual capacity of the algorithms to predict for unseen scenarios, e.g., for a new year.

Inside each fold, we select the best model by optimizing the hyper-parameters of each method. HiPaR's enumeration phase can take long for very low support thresholds. Therefore we run the enumeration phase with a support threshold of 30% the size of the dataset once – i.e., regions covering fewer points are not explored –, and we then optimize the hyper-parameters of the selection phase to pick the most performing set of rules.

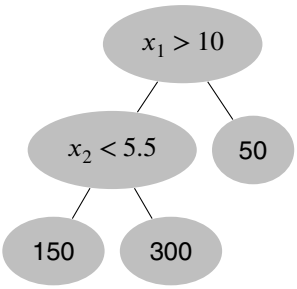
We use the coefficient of determination (R^2) as prediction performance metric. The R^2 score is defined as the proportion of the variance in the predicted target variable explained by the independent variables. Contrary to the root mean square error (RMSE), R^2 values can be compared among different prediction tasks (e.g., disease incidence and symptoms appearance). Indeed, the closer to 1 the R^2 is, the better the model fits the data. Values close to zero denote a performance comparable to predicting the mean of the target variable, whereas negative scores mean the model is worse than a mean-based simple predictor.

2.3.2. Complexity measure

To measure the complexity of the studied machine learning models, we resort to the complexity measure for pattern-based models proposed by [10] that counts the number of elements in the model. An element is either a non-zero coefficient or a condition on a predicting variable. We remark that this measure is also applicable to tree-based methods such as random forests or gradient boosting trees because each node of each tree of the ensemble defines either a condition on one attribute or a linear model – for simple regression trees this linear model is a single constant. The number of elements can be very large when the ensemble consists of many trees, which points out the complexity of such models.

Under this principle, a Lasso model is generally less complex than a HiPaR model with several rules. This is the case because for Lasso we only need to count the non-zero coefficients in the linear function, whereas for HiPaR we must consider both the number of conditions and the coefficients of each of the local models.

If we consider the following regression tree T:



Then its complexity $c(T)$ is 5. Likewise, if we consider the rule R:

$$C_1 \wedge C_2 \wedge C_3 \wedge C_4 \Rightarrow \boldsymbol{y} = 3x_1 + 4x_2 - 4x_3 + 8, \quad (7)$$

then its complexity $c(R)$ is 8 because the rule consists of 4 conditions and 4 linear coefficients.

2.4. Results

2.4.1. Performance-complexity trade-off

Fig. 2 depicts the trade-off between the complexity and accuracy of the studied machine learning methods. On the x-axis we show the complexity of the models (in log scale). The y-axis corresponds to the median R^2 coefficient of each model in cross-validation. Models located in the top-left part of the space strike a better accuracy-complexity trade-off as they predict the data more accurately with fewer elements. As suggested by [10], more complex models such as random forests or gradient boosting trees achieve the best performance at the price of high complexity. Lasso regression, our baseline, is often the least accurate model. HiPaR positions itself in between linear regression and ensemble methods striking a very interesting trade-off for 3 of the 4 prediction tasks.

We highlight that accuracy varies drastically across tasks: All models struggle when it comes to predicting the date of apparition of downy mildew in vine cultures, as the median R^2 for all methods is negative (bottom-right figure). We observe R^2 scores between 0.12 and 0.26 for the final downy-mildew incidence (on the bottom-left) with gradient boosting as the winner. HiPaR lies close to Lasso, which means that it did not find many regression rules improving performances marginally over the baseline. The performance different between the two target variables in the downy-mildew dataset could be explained by the relatively low number of observations for the date of symptoms apparition – 359 versus 700 observations for the end-of-season incidence.

The reach of the aggregated variables is relatively limited too. By this we mean most of these variables over a range of 4 weeks before data collection. While this confirms the trade-off, the low R^2 makes this dataset less interesting to study further.

The results for the sugar beet Cercospora are more encouraging. The R^2 median scores for the apparition date vary between 0.13 and 0.18 with gradient boosting leading the rank and followed by HiPaR (top-right figure). For the prediction of the end-of-season incidence performance ranges from 0.05 to 0.35. In this use case HiPaR outperforms all methods and finds a large number of rules that improve performance significantly when compared to a single linear model, and without incurring as much complexity as the ensemble methods.

When we look at the performance of the methods per year (Figs. 3-6), we notice that performance can vary drastically from one year to another, and that both end-of-season incidence and date of apparition are very hard to model for some years. This is true for all methods. As a general trend, we can observe that Cercospora end of season incidence predictions seem to follow a downward trend in performance. The performance variability across folds (Figs. 3-6) for the different methods is comparable and does not seem to follow a noticeable pattern.

Now that we have illustrated the accuracy-complexity trade-off present in our use cases, we delve into the knowledge captured by the

Table 2
Overview of the machine learning methods used in this study.

Models	Characteristics	Advantages	Disadvantages
Lasso	Sparse linear regression	Simple/interpretable	Baseline method
HiPaR	Pattern-based	Medium-complexity	High computation time
Random Forests	Ensemble-, tree-based	High accuracy, Built-in feature importance values	Black-box model
Gradient Boosting	Ensemble-based	High accuracy	Black-box model

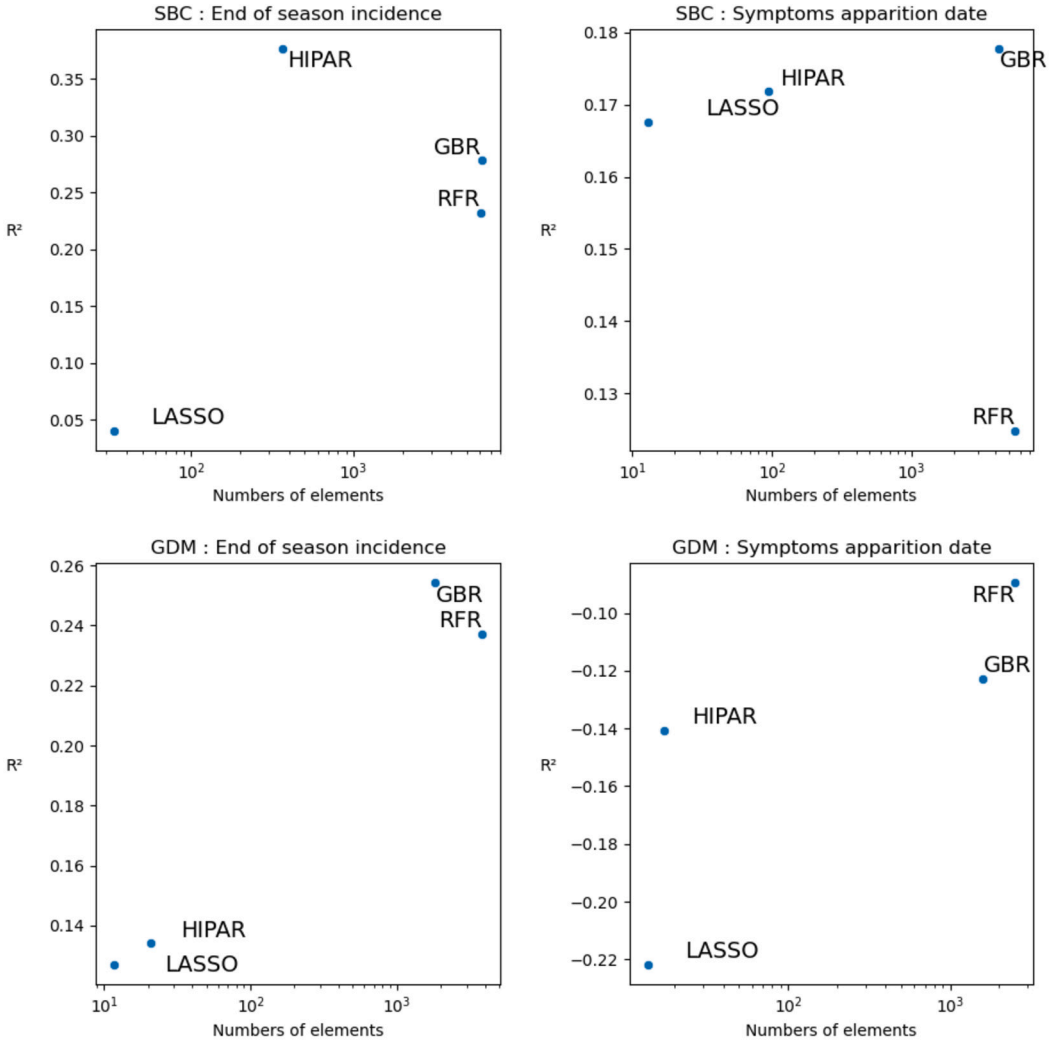


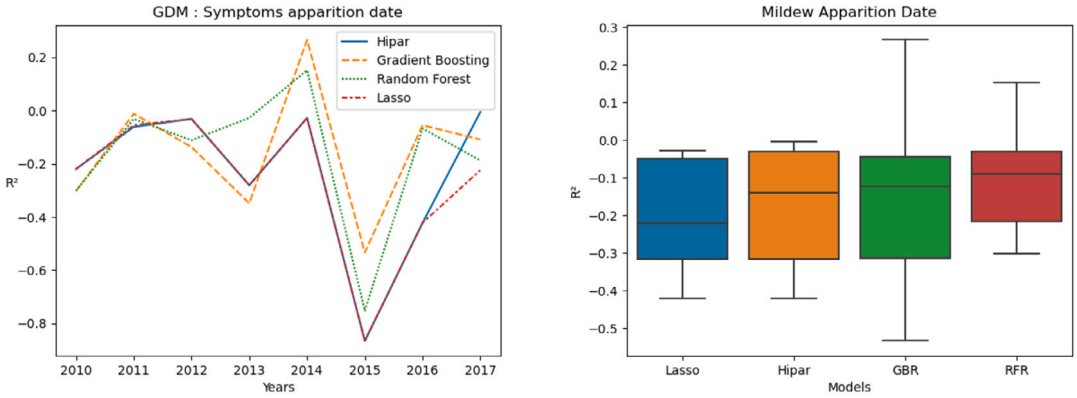
Fig. 2. R^2 of different machine learning models compared against their complexity. The x-axis correspond to the number of elements that compose each model (log scale). The y-axis is the median R^2 values in cross-validation. GBR stands for gradient boosting regression, and RFR for random forests regression.

different methods. To do so we analyze the models trained to predict year 2009 for the end-of-season incidence of the sugar beet Cercospora, as these models exhibit the highest explained variance across all years (R^2 scores of 0.67 and 0.66 for gradient boosting trees and random forests, 0.3 for Lasso, and 0.47 for HiPaR). For white-box models such as Lasso and HiPaR, we conduct direct inspection of the models' elements. For the complex black-box approaches, we resort to classical model inspection techniques and assess whether our models agree on the relationships between the predictive variables and the target variables.

2.4.2. Use case: incidence of the sugar beet Cercospora

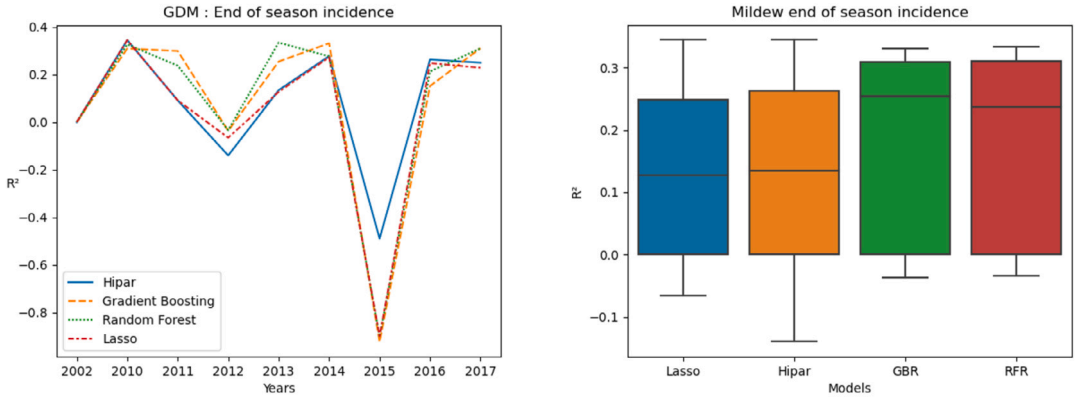
In this section we carry out an inspection phase aimed to distill agronomical insights from the experimental machine learning models trained to predict the incidence of sugar beet Cercospora. These models were trained on all years except 2009 and correspond to the most performing cross-validation round of our experiments. We resort to classical interpretation techniques including feature importance rankings, partial dependence plots, and simple rule inspection. The first technique tells us which are the most important variables that play a role in the prediction. PDPs and rule inspection allow us to identify *threshold effects* on the predicting variables, that is, cases when the behavior of the target variable varies in a piece-wise manner, i.e., according to thresholds on the predicting variables. Pattern-based regression methods such as HiPaR are good at detecting such kind of effects. Moreover, such methods allow us to study more fine-grained interactions among the predicting variables present in the rules. Our observations set the ground for the discussion in Section 3.

Feature importance A simple way to interpret the knowledge captured by a machine learning model is to construct a feature-importance ranking that tell us how much the model's input variables affect the model's



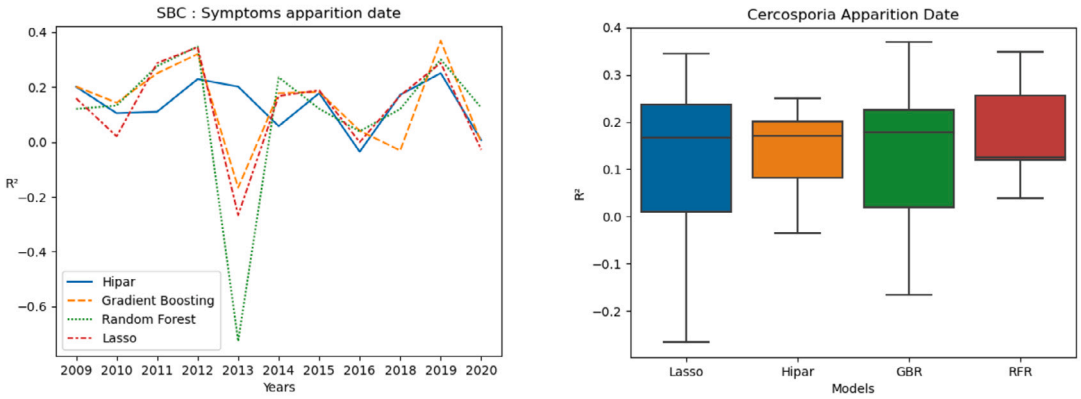
(a) R^2 year-to-year cross-validation (b) R^2 year-to-year distribution

Fig. 3. Mildew symptoms apparition date.



(a) R^2 year-to-year cross-validation (b) R^2 year-to-year distribution

Fig. 4. Mildew end of season incidence.

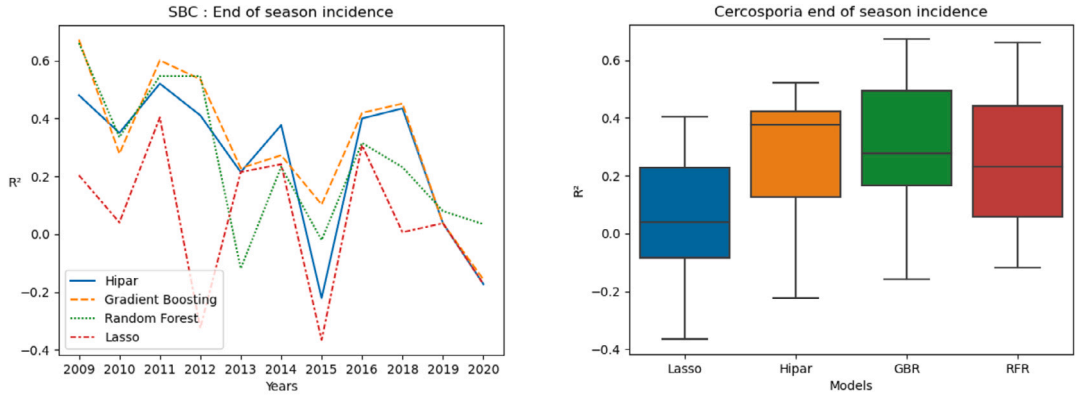


(a) R^2 year-to-year cross-validation (b) R^2 year-to-year cross-validation distribution

Fig. 5. Cercosporia symptoms date of apparition.

output. This ranking can be based on the actual contributions of a variable to the answers of a model, e.g., the coefficients of a linear regression, or on model-aware scores computed a posteriori for black-box models. In this spirit we contrast the feature-importance rankings of Lasso, RFR and GBR and depicted them in Fig. 7. Lasso's linear coefficients encode the actual contributions of the input features to the answers of the model. They are therefore signed. To turn the linear coefficients into importance scores, we take their absolute value. Conversely, RFR and GBR are based on tree ensembles for which different importance scores have been developed. We choose the permutation feature importance method as implemented in the scikit-learn library. This approach estimates the importance of a feature by shuffling its values across rows in $\boldsymbol{X}$. The resulting decrease in accuracy is then used to determine how much the model relies on a feature to make predictions – the higher the decrease, the informative the feature is for predicting the target variable.

As we can see, RFR and GBR yield very similar rankings – their top-4 variables are the same even though the order is not identical. The vari-



(a) R^2 year-to-year cross-validation

(b) R^2 year-to-year cross-validation distribution

Fig. 6. Cercosporia end of season incidence.

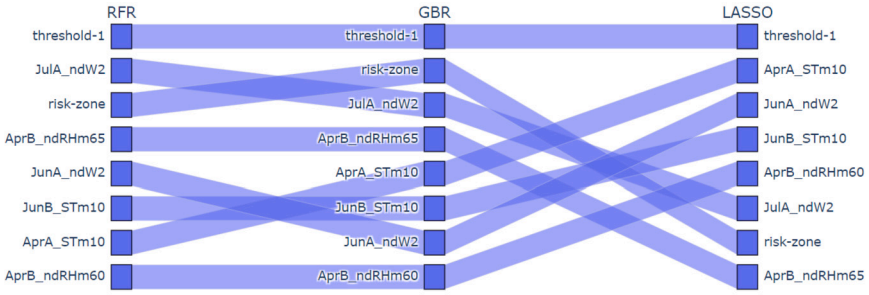


Fig. 7. Parallel coordinates chart comparing feature-importance rankings for Lasso, random forests, and gradient boosting trees when predicting end-of-season incidence for the sugar beet Cercospora. For each model we chose the top-4 most important features. For each model, features below the 4 can be further down the importance order than what is displayed.

Table 3
Top-5 important linear coefficients learned by Lasso.

Variable	Coefficient
Threshold-1	-41
AprA-STm10	27.64
JunA-ndW2	-23.39
JunB-STm10	22.24
AprB-ndRHm60	14.73

able *threshold-1* is the most important feature for all three models. This variable represents the day in which the first symptoms of Cercospora were detected in the culture. Conversely RFR and GBR's accuracy rely on the *risk-zone* expert-based indicator, which is less important for linear regression. While importance scores tell us which information the model is looking at, it does not tell whether those features tend to increase or decrease the model's incidence prediction. We can, however, obtain this information by looking at the linear coefficients learned by Lasso.

Table 3 shows the top-5 most important linear coefficients.

We remind the reader the meaning of these variables:

Threshold-1 : The symptoms apparition date

AprA-STm10 : The sum of the daily average temperatures of the days above 10 °C during the first half of April.

JunB-ndW2 : The number of days in the first half of June such that the average wind speed was higher than 2 m/s.

JulB-STm10 : The sum of the daily average temperatures above 10 °C during the second half of June.

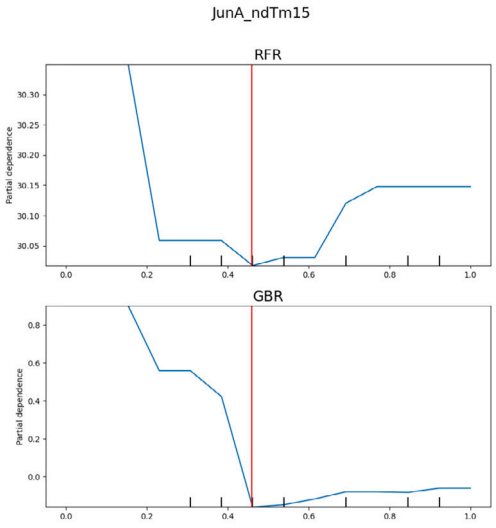
AprB-ndRHm60 : The number of days in the second half of April such that the relative humidity is higher than 60%.

Table 3 tells us that the later symptoms appear, the lower the final incidence tends to be. The predicted incidence tends to increase when temperature and humidity in June and April increase, whereas faster winds seem to hinder the development of Cercospora. These results must be taken with a grain of salt given the fact that our baseline Lasso model can explain only 30% of the target variable's variance. That said, these variables are used by more accurate models such as RFR and GBR, which means that we are not uncorrelated with the target variable.

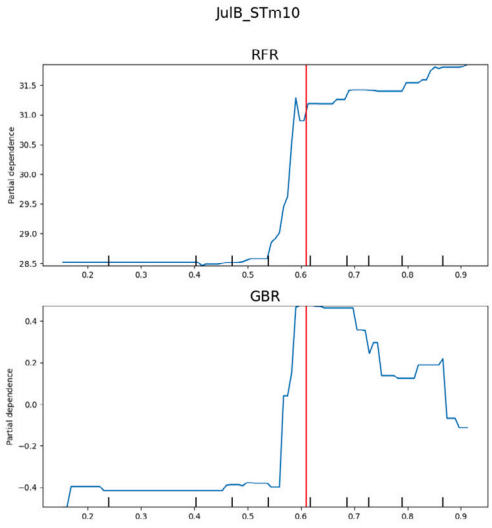
Threshold effects As stated before, pattern-based regression methods are constructed to detect predicting variable threshold effects on the target variable. In HiPaR such effects are explicitly stated in the rule conditions. To observe whether our models captured such effects we have a deep look at the hybrid rules learned by HiPaR on our studied use case, and contrast those thresholds to those learned by the more complex models, namely RFR and GBR. Since those models are actually based very large ensembles of threshold-based estimators, we observe those threshold effects by means of partial dependence plots (PDP). This widely-used inspection technique allows us to visualize the behavior of a model's prediction (y-axis) for the different values of a predicting variable (x-axis).

In our use case, HiPaR learned 3 hybrid rules whose conditions are listed in Table 4. As displayed before, thresholds (in red) used in HiPaR's rules roughly fits with changes in the PDPs behavior. While they not the most important features as seen before, it seems to indicate that these thresholds are not insignificant (according to RFR and GBR models). We suppose that these features might act as proxies for other features, or simply have an indirect influence on the final result that is not detectable by using PDPs.

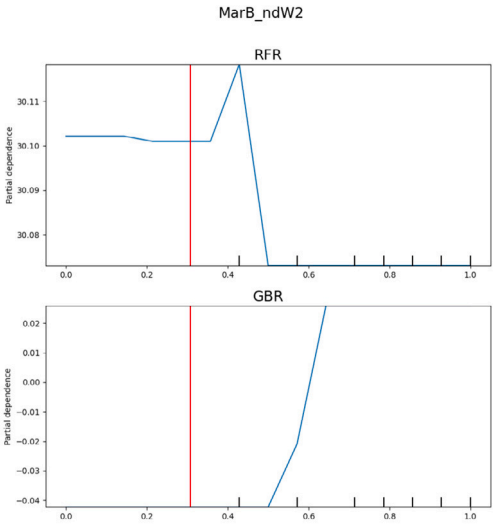
In other words, HiPaR detected different linear behaviors based on whether a plot lies or not within a region deemed risky by experts



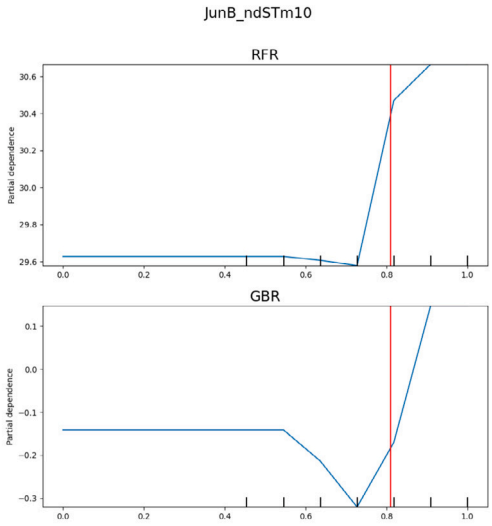
(a) Number of days where average temperatures were higher than 15° during the first half of july



(b) Number of hours where average temperatures were higher than 10° during the second half of june



(c) Number of days where average wind speed were higher than 2km/h during the 2nd half of march



(d) Number of days where average temperatures were higher than 10° during the 2nd half of june

Fig. 8. Partial Dependence Plots for the predicting variables *MarB-ndW4*, *JulB-STm10*, *JunA-ndTm15* and *JunB-STm10* on random forests and gradient boosting trees. The red line ▬ represents a threshold learned by HiPaR.

Table 4
Conditions of the hybrid rules learned by HiPaR when predicting the end-of-season incidence of the sugar beet Cercospora.

Rule 1	*JunA-ndTm15* $<$ 8, *risk-zone* $=$ *false*
Rule 2	*JulB-STm10* $<$ 324
Rule 3	*MarB-ndW2* $\geq$ 4, *JunB-ndTm15* $<$ 13

(*risk-zone*), or whether the number of hot days in June and July are below certain thresholds (*JunA-ndTm15*, *JulB-STm10*, *JunB-ndTm15*), or whether the second half of March was windy (*MarB-ndW2*). We now construct PDPs for the numerical variables on RFR and GBR, which we depict in Fig. 8.

Feature interactions Each of the conditions listed in Table 4 is associated to a local linear model (learned using Lasso). Those models reveal local interactions between the variables in the conditions and the linear coefficients, and are designed to refine the baseline linear (called also the default) model learned on the entire dataset. Out of 368 features used as input in the models, Lasso selects between 25 and 55. This represents between 6.7% and 15% of the available features. Moreover, local models are systematically less complex than the default model as Table 5 shows.

We can also observe that coefficients overlap between the different hybrid rules is low. This means that each local model is relying on different signals to make predictions on the end-of-season incidence. Fig. 9 depicts the intensity and polarity of 16 of those coefficients for both the default and local models.

Our first observation is that the apparition date (*threshold-1*) is consistently important across all models – and always correlated negatively with the predict incidence. The features *risk-zone* and *JunA-ndW2* (wind speed in the first half of June) are used in all models except the first rule

Fig. 9. A color encoding for the linear coefficients of the three hybrid rules learned by HiPaR. Cells in white ☐ denote features with a linear coefficient strictly equal to 0, which means those features aren't used by the model.

Table 5
Number of common non-zero coefficients of the linear models learned by HiPaR for the prediction of the end-of-season sugar beet Cercospora.

	Rule 1	Rule 2	Rule 3	Default Model
Rule 1	25	8	3	6
Rule 2		28	6	16
Rule 3			26	12
Default Model				55

because these variables appear in the conditional part of this rule (Table 4).

This rule can be interpreted as follows: Plots with lower disease exposure (*risk-zone* = *false*) and lower temperatures in the first half of June (*JunA-ndTm15* $<$ 8), experience an aggravated development of Cercosporia as humidity in May (*MayA-ndRHm60*), wind speed in March (*MarA-ndW4*), and rainfall in February (*FebA-SR*) increase. Wind during June (*JunB-ndW2*) is associated to a slow down of the disease.

The second rule suggests that lower temperatures in July (*JulB-STm10* $<$ 324) make Cercosporia sensitive to wind in January, February, June, and July (*JanB-ndW2*, *FebA-ndW2*, *JunA-ndW2*, *JulA-ndW2*). Conversely, a wet June (*JunA-ndRHm65*) or a windy March (*MarA-ndW4*) appear as aggravating factors. A windy July (*JulA-ndW4*), a rainy February (*FebA-SR*) and a hotter April (*AprA-STm10*) have a mitigated effect on the development of Cercosporia.

The third rule triggers when the month of March is windy (*MarB-ndW4* $\geq$ 4) June is not very hot (*JunB-ndTm15* $<$ 13). In that case, higher temperatures in May (*MayB-ndTm20*) and a wet April (AprB-ndRHm60) are correlated with the growth of Cercosporia growth. Conversely, wind in June (*JunA-ndW2*) and July (*JulA-ndW4*) exhibit a negative correlation with growth.

Finally, we observe that the default model combines signals from all the local rules, even though it does not always rely on the same variables. This happens because the learning objective of this model must fit the observations from all the sub-regions. This translates into selecting variables (such as *JunB-STm10*) that explain incidence for all the observations, i.e., at the global level, but that have little to no explanation power when limited to subsets of the data such as the observations on regions not deemed risky by the experts (*risk-zone* = *false*).

3. Discussions

We structure our discussion along three axes: (a) the complexity-accuracy trade-off discussed in Subsection 2.4.1, (b) the implications of complexity in interpretability, and (c) the agronomical insights offered by the models trained.

Complexity and accuracy Our results go in line with what has been observed in other works on model complexity [15,10], that is, the tendency of complex models to outperform simple models in terms of prediction accuracy. It is crucial to highlight though, that such a trend holds under the assumption that the models have been properly parameterized and trained. For instance, a complex model trained on very little data will surely over-fit that data specially if there are as many or more parameters than data points. Conversely if the data adheres to the learning hypothesis of a simple model, e.g., linearity, such model will surely shine in terms of performance regardless of its complexity. Finally, even if a model was trained under a reasonable learning hypothesis, testing it on data that diverges from the training distribution will result in unsatisfactory prediction performance. We can observe such a phenomenon for the models tested on years 2013 and 2015 for the prediction of both the incidence and the date of apparition in both cultures. The observations collected those years are atypical because some of the predicting variables exhibited measures outside the amplitudes observed other years. This translated into a clear under-fitting with the lowest R^2 scores registered in our experiments.

Interpretability It is widely-assumed that interpretability and model complexity are positively correlated. An illustration of such phenomenon can be observed from our use case. Both linear and pattern-based model allowed us to distill insights easily and directly from the structures of the models themselves. For more complex models such as random forests and gradient boosting trees we had to resort to external inspection tools such as the permutation-based accuracy decrease and the partial dependence plots (PDPs). Albeit effective, those techniques have limitations. Importance scores do not tell us if a variable is positively or negatively correlated with the prediction of the model. PDPs can be applied to up to two variables at the same time, and make independence assumptions that often do not hold on the data. This happens because each point in the curve is the result of averaging the model answers over all possible values of the remaining predicting variables. Since some combinations of values may be unlikely, PDPs must be taken with a grain of salt, specially when the predicting variables exhibit some correlation. That said, the PDPs for RFR and GBR in our experiments were in concordance with the threshold effects observed when using HiPaR. It should be noted that while RFR, GBR, and HiPaR resort to thresholds on the predicting variables, the fact they all outperform Lasso significantly suggest that threshold effects are a reasonable hypothesis for the prediction of plant diseases based on meteorological data.

Agronomical insights Based on our use case study on the sugar beet Cercospora, we observe that aggregating the meteorological indicators according to the seasons, i.e., winter, spring, and summer can effectively explain some of the variation in disease incidences.

Winter defines the initial conditions: This is the period in which the primary inoculum of Cercospora lies in the soil in the form of spores. Spring defines the development period for both crops and the Cercospora. Finally, summer encompasses the end of the season, and the moment in which the disease's symptoms, as well as its effects, are obvious.

As a general rule, dry summers seem to hinder the growth of Cercospora. This follows from the importance assigned by the models to the wind and temperature factors during June and July. Dry winters also seem to mitigate the disease's spread. Conversely, a hot and humid spring stands as the main aggravating factor in Cercospora's incidence. Thanks to the hybrid rules provided by HiPaR, we can obtain more nuanced relationships between the incidence and the predicting variables. Rule 2 in Table 4 tells us that a mild month of July should make us

focus the attention on the initial conditions (winter), in particular the wind and the sun exposure and the temperature – the two latter factors contributing positively to the presence of the primary inoculum. Moreover, a windy spring with mild temperatures in June should target our attention towards the development phase (spring) in particular towards temperature and humidity, which are positively correlated with incidence. In all cases, the date of apparition is the best predictor of the final incidence, which means that early detection is the best weapon against Cercospora.

We could not draw insights from the prediction of the downy-mildew on the vine because the transparent models explain no more than 14% of the observed variance for the incidence – the results for the date of apparition are worse. We think this performance gap is due to the fact that the dataset relies only on meteorological measures for the four weeks that precede the end of the season. In other words, this dataset lack the richness of the meteorological signals available for the sugar beet Cercospora dataset. This observation confirms the importance of accurate and complete meteorological measurements when modeling the dynamics of plant diseases. We also believe that the studying the impact of the granularity of the meteorological indicators in such tasks remains an interesting research avenue.

4. Conclusions

In this paper, we have shown the interest of exploring the complexity trade-off for machine learning models when applied to predicting the incidence of plant diseases. It is accepted that in some applications, complex models such as neural networks or gradient boosting generally perform better than simpler ones such as linear regression. This comes, however, at the cost of interpretability, which (a) is vital when we need to draw insights from the prediction model, and (b) fosters transparency, which can in turn favor acceptability by users. Post-hoc explanation methods can help us extract insights from accurate black-box models, but they are not the only solution as we have shown in this work: medium-complexity models based on pattern-aided regression can achieve competitive prediction performance while remaining simple and interpretable. Moreover, our experiments with post-hoc explainability techniques such as partial dependence plots suggest that pattern-aided regression can reveal threshold effects that are also exploited by the more accurate black-box ensemble methods. Using those models, we have also shown that medium-complexity methods are well suited to extract more pertinent information compared to simpler models. Likewise medium-complexity models are easier to interpret compared to more complex methods. This shows the utility of pattern-aided regression and makes it appealing for crop prediction. Since there is a direct correlation between interpretability and acceptability, evaluating the complexity of a model is not trivial and should be taken into account. This aspect has been already addressed from the angle of learning complexity [25] or from the perspective of data complexity [26], but rarely in terms of the complexity of the resulting model. Finally, our study suggests that the meteorological inter-annual variations make disease incidence prediction very challenging, and that predicting disease incidence for any year requires more research as well as more historical (quality) data.

In the future we envision to study whether increasing the temporal and spatial granularity of the meteorological attributes can help us improve the quality of our predictions. An interesting research avenue could be to apply representation learning techniques in order to learn novel and useful meteorological indicators that predict disease incidence more accurately. Given the inter-annual variations of weather patterns, future approaches should be able to categorize prediction models based on the meteorological profile of the data used to train them. We believe that unsupervised learning techniques could be adapted in that regard. Such approaches may be even necessary in the light of a climate that will keep changing in the upcoming years.

CRediT authorship contribution statement

Olivier Gauriau: Methodology, Resources, Software, Visualization, Writing – original draft. **Luis Galárraga:** Software, Supervision, Writing – original draft, Writing – review & editing. **François Brun:** Funding acquisition, Project administration, Supervision, Writing – original draft, Writing – review & editing. **Alexandre Termier:** Supervision, Writing – review & editing. **Loïc Davadan:** Data curation, Resources, Writing – review & editing. **François Joudelat:** Data curation, Resources, Writing – review & editing.

Declaration of competing interest

The 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.

Data availability

The data that has been used is confidential.

Acknowledgements

We thank the French applied agricultural research organization for sugar beet (ITB - Institut Technique de la Betterave) (ITB) and French wine and vine Institut (IFV – institut francais de la Vigne et du Vin) for providing the meteorological and agronomical data used in this study.

We also thank all the experts from each institute who helped us in interpreting our results with their insight: Fabienne Maupas, Ghislain Malatesta, Gouwie Céline (ITB) and Marc Raynal, Christian Debord, Xavier Burgun, Marc Vergnes (IFV). We thank Lucile Vallet (Acta) for her work and the preparation of the sugar beet dataset.

This work was funded by the DigitAg institute (ANR-16-CONV-0004) and the RegEpi project (ECOPHYTO R&D program, Ecophyto-2019-REGEPI grant, French Biodiversity Agency – OFB). This project data was also part of the network data science and modeling methods for agriculture and agri-food sector (www.modelia.org, funded by CASDAR grants of the French Ministry of Agriculture, RMT-SDMAA-19WRT034).

References

[1] Ian Heap, Global perspective of herbicide-resistant weeds, Pest Manag. Sci. (ISSN 1526-4998) 70 (9) (2014), https://doi.org/10.1002/ps.3696 1306–1315.

[2] Paul Parsons, Elaine Freeman, Ryan Weidling, Gary L. Williams, Philip Gill, Neil Byron, Using existing knowledge for the risk evaluation of crop protection products in order to guide exposure driven data generation strategies and minimise unnecessary animal testing, Crit. Rev. Toxicol. (ISSN 1547-6898) 51 (7) (2021) 600–621, https://doi.org/10.1080/10408444.2021.1987384.

[3] Mathilde Chen, Analyse du risque de mildiou de la vigne dans le Bordelais à partir de données régionales et d'informations locales collectées en cours de saison, PhD thesis, Université Paris-Saclay (ComUE), 2019.

[4] Gareth Edwards-Jones, Knowledge-based systems for crop protection: theory and practice, Crop Prot. (ISSN 0261-2194) 12 (8) (1993) 565–578, https://doi.org/10.1016/0261-2194(93)90119-4.

[5] Luisa Velasquez-Camacho, Marta Otero, Boris Basile, Josep Pijuan, Giandomenico Corrado, Current trends and perspectives on predictive models for mildew diseases in vineyards, Microorganisms (ISSN 2076-2607) 11 (1) (2023), https://doi.org/10.3390/microorganisms11010073 1–19.

[6] Thomas van Klompenburg, Ayalew Kassahun, Cagatay Catal, Crop yield prediction using machine learning: a systematic literature review, Comput. Electron. Agric. (ISSN 0168-1699) 177 (2020), https://doi.org/10.1016/j.compag.2020.105709 105709.

[7] Ryan H.L. Ip Li Minn Ang, Kah Phooi Seng, J.C. Broster, J.E. Pratley, Big data and machine learning for crop protection, Comput. Electron. Agric. (ISSN 0168-1699) 151 (2018) 376–383, https://doi.org/10.1016/j.compag.2018.06.008.

[8] Konstantinos G. Liakos, Patrizia Busato, Dimitrios Moshou, Simon Pearson, Dionysis Bochtis, Machine learning in agriculture: a review, Sensors (ISSN 1424-8220) 18 (8) (2018), https://doi.org/10.3390/s18082674 1–29.

[9] F.K. van Evert, S. Fountas, D. Jakovetic, V. Crnojevic, I. Travlos, C. Kempenaar, Big Data for weed control and crop protection, Weed Res. (ISSN 1365-3180) 57 (4) (2017), https://doi.org/10.1111/wre.12255 218–233.

[10] Luis Galárraga, Olivier Pelgrin, Alexandre Termier, HiPaR: hierarchical pattern-aided regression, in: Advances in Knowledge Discovery and Data Mining, Springer International Publishing, Cham, ISBN 978-3-030-75762-5, 2021, pp. 320–332.
[11] Toshiki Mori, Naoshi Uchihira, Balancing the trade-off between accuracy and interpretability in software defect prediction, Empir. Softw. Eng. (ISSN 1382-3256) 24 (2) (apr 2019) 779–825, https://doi.org/10.1007/s10664-018-9638-1.
[12] Ulf Johansson, Cecilia Sönströd, Ulf Norinder, Henrik Boström, Trade-off between accuracy and interpretability for predictive in silico modeling, Future Med. Chem. 3 (6) (2011) 647–663, https://doi.org/10.4155/fmc.11.23, PMID 21554073.
[13] Cynthia Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. 1 (2019) 206, https://doi.org/10.1038/s42256-019-0048-x.
[14] Andrew Bell, Ian Solano-Kamaiko, Oded Nov, Julia Stoyanovich, It's just not that simple: an empirical study of the accuracy-explainability trade-off in machine learning for public policy, in: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT '22, New York, NY, USA, Association for Computing Machinery, ISBN 9781450393522, 2022, pp. 248–266.
[15] G. Dong, V. Taslimitehrani, Pattern-aided regression modeling and prediction model analysis, IEEE Trans. Knowl. Data Eng. 27 (9) (2015) 2452–2465.
[16] Gianni Fenu, Francesca Maridina Malloci, Review forecasting plant and crop disease: an explorative study on current algorithms, Big Data Cogn. Comput. (ISSN 2504-2289) 5 (1) (2021), https://doi.org/10.3390/bdcc5010002 1–24.
[17] P. Quintana-Seguí, P. Le Moigne, Y. Durand, E. Martin, F. Habets, M. Baillon, C. Canellas, L. Franchisteguy, S. Morel, Analysis of near-surface atmospheric variables: validation of the SAFRAN analysis over France, J. Appl. Meteorol. Climatol. 47 (1) (2008) 92–107, https://doi.org/10.1175/2007JAMC1636.1, https://journals.ametsoc.org/view/journals/apme/47/1/2007jamc1636.1.xml.
[18] Robert Tibshirani, Regression shrinkage and selection via the Lasso, J. R. Stat. Soc. Ser. B 58 (1994) 267–288.
[19] Stefan Kramer, Structural Regression Trees, AAAI/IAAI, vol. 1, Citeseer, 1996, pp. 812–819.
[20] Leo Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[21] Dhivya Elavarasan, Durai Raj Vincent, Vishal Sharma, Albert Y. Zomaya, Kathiravan Srinivasan, Forecasting yield by integrating agrarian factors and machine learning models: a survey, Comput. Electron. Agric. (ISSN 0168-1699) 155 (2018) 257–282, https://doi.org/10.1016/j.compag.2018.10.024.
[22] Llew Mason, Jonathan Baxter, Peter Bartlett, Marcus Frean, Boosting algorithms as gradient descent, Adv. Neural Inf. Process. Syst. 12 (1999).
[23] Victor E. McGee, Willard T. Carleton, Piecewise regression, J. Am. Stat. Assoc. 65 (331) (1970) 1109–1124.
[24] Yong Wang, Ian H. Witten, Inducing model trees for continuous classes, in: ECML Poster Papers, 1997.
[25] Michael J. Kearns, The Computational Complexity of Machine Learning, MIT Press, 1990.
[26] Raaz Dwivedi, Chandan Singh, Bin Yu, Martin J. Wainwright, Revisiting complexity and the bias-variance tradeoff, arXiv preprint, arXiv:2006.10189, 2020.