A comparison of eight metamodeling techniques for the simulation of N_2O fluxes and N leaching from corn crops

Nathalie Villa-Vialaneix[a,b,*], Marco Follador[c], Marco Ratto[d], Adrian Leip[c]

[a] *IUT de Perpignan (Dpt STID, Carcassonne), Univ. Perpignan Via Domitia, France*
[b] *Institut de Mathématiques de Toulouse, Université de Toulouse, France*
[c] *European Commission, Institute for Environment and Sustainability, CCU, Ispra, Italy*
[d] *European Commission, Econometrics and Applied Statistics Unit, Ispra, Italy*

Abstract

The environmental costs of intensive farming activities are often underestimated or not traded by the market, even though they play an important role in addressing future society's needs. The estimation of nitrogen (N) dynamics is thus an important issue which demands detailed simulation based methods and their integrated use to correctly represent complex and non-linear interactions into cropping systems. To calculate the N_2O flux and N leaching from European arable lands, a modeling framework has been developed by linking the CAPRI agro-economic dataset with the DNDC-EUROPE bio-geo-chemical model. But, despite the great power of modern calculators, their use at continental scale is often too computationally costly. By comparing several statistical methods this paper aims to design a metamodel able to approximate the expensive code of the detailed modeling approach, devising the best compromise between estimation performance and simulation speed. We describe the use of two parametric (linear) models and six nonparametric approaches: two methods based on splines (ACOSSO and SDR), one method based on kriging (DACE), a neural networks method (multilayer perceptron, MLP), SVM and a bagging method (random forest, RF). This analysis shows that, as long as few data are available to train the model, splines approaches lead to best results, while when the size of training dataset increases, SVM and RF provide faster and more accurate solutions.

*Corresponding author.
Email address: nathalie.villa@math.univ-toulouse.fr (Nathalie Villa-Vialaneix)

Keywords: Metamodeling, splines, SVM, neural network, random forest, N_2O flux, N leaching, agriculture

1. Introduction

The impact of modern agriculture on the environment is well documented (Power, 2010; Tilman et al., 2002; Scherr and Sthapit, 2009; FAO, 2007, 2005; Singh, 2000; Matson et al., 1997). Intensive farming has a high consumption of nitrogen, which is often in-efficiently used, particularly in livestock production systems (Leip et al., 2011b; Webb et al., 2005; Oenema et al., 2007; Chadwick, 2005). This leads to a large surplus of nitrogen which is lost to the environment. Up to 95% of ammonia emission in Europe have their origin in agricultural activities (Kirchmann et al., 1998; Leip et al., 2011a) contributing to eutrophication, loss of biodiversity and health problems. Beside NH_3, nitrate leaching below the soil root zone and entering the groundwater poses a particular problem for the quality of drinking water (van Grinsven et al., 2006). Additionally, agricultural sector is the major source of anthropogenic emissions of N_2O from the soils, mainly as a consequence of the application of mineral fertilizer or manure nitrogen (Del Grosso et al., 2006; Leip et al., 2011c; European Environment Agency, 2010; Leip et al., 2005). N_2O is a potent greenhouse gas (GHG) contributing with each kilogram emitted about 300 times more to global warming than the same mass emitted as CO_2, on the basis of a 100-years time horizon (Intergovernmental Panel on Climate Change, 2007).

Various European legislations attempt to reduce the environmental impact of the agriculture sector, particularly the Nitrates Directive (European Council, 1991) and the Water Framework Directive (European Council, 2000). Initially, however, compliance to these directives was poor (Oenema et al., 2009; European Commission, 2002). Therefore, with the last reform of the Common Agricultural Policy (CAP) in the year 2003 (European Council, 2003), the European Union introduced a compulsory Cross-Compliance (CC) mechanism to improve compliance with 18 environmental, food safety, animal welfare, and animal and plant health standards (Statutory Management Requirements, SMRs) as well as with requirements to maintain farmlands in good agricultural and environmental condition (Good Agricultural and Environment Condition requirements, GAECs), as prerequisite for receiving direct payments (European Union Commission, 2004; European Council,

2009; European Union Commission, 2009; Dimopoulus et al., 2007; Jongeneel et al., 2007). The SMRs are based on pre-existing EU Directives and Regulations such as Nitrate Directives. The GAECs focus on soil erosion, soil organic matter, soil structure and a minimum level of maintenance; for each of these issues a number of standards are listed (Alliance Environnement, 2007).

It remains nevertheless a challenge to monitor compliance and to assess the impact of the cross-compliance legislations not only on the environment, but also on animal welfare, farmer's income, production levels etc. In order to help with this task, the EU-project Cross-Compliance Assessment Tool (CCAT) developed a simulation platform to provide scientifically sound and regionally differentiated responses to various farming scenarios (Elbersen et al., 2010; Jongeneel et al., 2007).

CCAT integrates complementary models to assess changes in organic carbon and nitrogen fluxes from soils (De Vries et al., 2008). Carbon and nitrogen turnover are very complex processes, characterized by a high spatial variability and a strong dependence on environmental factors such as meteorological conditions and soils (Shaffer and Ma, 2001; Zhang et al., 2002). Quantification of fluxes, and specifically a meaningful quantification of the response to mitigation measures at the regional level requires the simulation of farm management and the soil/plant/atmosphere continuum at the highest possible resolution (Anderson et al., 2003; Leip et al., 2011c). For the simulation of N_2O fluxes and N-leaching, the process-based biogeochemistry model DNDC-EUROPE (Leip et al., 2008; Li et al., 1992; Li, 2000) was used. As DNDC-EUROPE is a complex model imposing high computational costs, the time needed to obtain simulation results in large scale applications (such as the European scale) can be restrictive. In particular, the direct use of the deterministic model is prohibited to extract efficiently estimations of the evolution of N_2O fluxes and N-leaching under changing conditions. Hence, there is a need for a second level of abstraction, modeling the DNDC-EUROPE model itself, which is called a *meta-model* (see Section 2 for a more specific definition of the concept of metamodeling). Metamodels are defined from a limited number of deterministic simulations for specific applications and/or scenario and allow to obtain fast estimations.

This issue is a topic of high interest that has previously been tackled in several papers: among others, (Bouzaher et al., 1993) develop a parametric model, including spatial dependency, to model water pollution. (Krysanova and Haberlandt, 2002; Haberlandt et al., 2002) describe a two-steps approach

to address the issue of N leaching and water pollution: they use a process-based model followed by a location of the results with a fuzzy rule. More recently, (Pineros Garcet et al., 2006) compare RBF neural networks with kriging modeling to build a metamodel for a deterministic N leaching model called WAVE (Vanclooster et al., 1996). The present article compares in detail different modeling tools in order to select the most reliable one to meta-model the DNDC-EUROPE tasks in the CCAT project Follador and Leip (2009). This study differs from the work of Vanclooster et al. (1996) because of the adopted European scale and of the analysis of 8 meta-modeling approaches (also including a kriging and a neural network method). The comparison has been based on the evaluation of meta-model performances, in terms of accuracy and computational costs, with different sizes of the training dataset.

The rest of the paper is organized as follows: Section 2 introduces the general principles and advantages of using a meta-model; Section 3 reviews in details the different types of metamodels compared in this study; Section 4 explains the Design Of the Experiments (DOE) and show the results of the comparison, highlighting how the availability of the training data can play an important role in the selection of the best type and form of the approximation. The supplementary material of this paper can be found at: `http://afoludata.jrc.ec.europa.eu/index.php/dataset/detail/232`.

2. From model to metamodel

A model is a simplified representation (abstraction) of reality developed for a specific goal; it may be deterministic or probabilistic. An integrated use of simulation based models is necessary to approximate our perception of complex and nonlinear interactions existing in human-natural systems by means of mathematical input-output (I/O) relationships. Despite the continuous increase of computer performance, the development of large simulation platforms remains often prohibited because of computational needs and parametrization constraints. More precisely, every model in a simulation platform such as DNDC-EUROPE, is characterized by several parameters, whose near-optimum set is defined during the calibration. A constraint applies restrictions to the kind of data that the model can use or to specific boundary conditions. The flux of I/O in the simulation platform can thus be impeded by the type of data/boundaries that constraints allow - or not allow - for the models at hand.

The use of this kind of simulation platform is therefore not recommended for all the applications which require many runs, such as sensitivity analysis or what-if studies. To overcome this limit, the process of abstraction can be applied to the model itself, obtaining a model of the model (2nd level of abstraction from reality) called meta-model (Blanning, 1975; Kleijnen, 1975; Sacks et al., 1989; van Gighc, 1991; Santner et al., 2003). A metamodel is an approximation of detailed model I/O transformations, built through a moderate number of computer experiments.

Replacing a detailed model with a metamodel generally brings some payoffs (Britz and Leip, 2009; Simpson et al., 2001):

- easier integration into other processes and simulation platforms;
- faster execution and reduced storage needs to estimate one specific output;
- easier applicability across different spatial and/or temporal scales and site-specific calibrations, as long as data corresponding to the new system parametrization are available.

As a consequence, a higher number of simulation runs become possible: using its interpolatory action makes a thorough sensitivity analysis more convenient and leads to a better understanding of I/O relationships. Also it offers usually a higher flexibility and can quickly be adapted to achieve a wide range of goals (prediction, optimization, exploration, validation). However, despites these advantages, they suffer from a few drawbacks: internal variables or outputs not originally considered can not be inspected and the prediction for input regimes outside the training/test set is impossible. Hence, a good metamodeling methodology should be able to provide fast predictions. But, considering that limitations, it also must have a low computational cost to be able to build a new metamodel from a new data set including new variables and/or a different range for these input variables.

Let $(\mathbf{X}, \mathbf{y})$ be the dataset consisting of N row vectors of input/output pairs $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i = (x_i^1, \ldots, x_i^d)^T \in \mathbb{R}^d$ $(i = 1, \ldots, N)$ are the model input and $y_i \in \mathbb{R}$ $(i = 1, \ldots, N)$ are the model responses for N experimental runs of the simulation platform. The mathematical representation of I/O relationships described by the detailed model can be written as

$$y_i = f(\mathbf{x}_i) \qquad i = 1, \ldots, N \tag{1}$$

which corresponds to a first abstraction from the real system. From the values of $\mathbf{X}$ and $\mathbf{y}$, also called *training set*, f is approximated by a function $\hat{f} : \mathbb{R}^d \to \mathbb{R}$, called metamodel, whose responses can be written as

$$\hat{y}_i = \hat{f}(\mathbf{x}_i).$$

and that correspond to a second abstraction from the reality. In this second abstraction, some of the input variables of Eq. (1) might not be useful and one of the issue of metamodeling can be to find the smallest subset of input variables relevant to achieve a good approximation of model (1).

Finally, the differences between the real system and the metamodel response, will be the sum of two approximations (Simpson et al., 2001): the first one introduced by the detailed model (1st abstraction) and the second one due to metamodeling (2nd abstraction). Of course, the validity and accuracy of a metamodel are conditioned by the validity of the original model: in the following, it is then supposed that the 1^{st} level of abstraction induces a small error compared to reality. Then, in this paper, we only focus on the second error, $|\hat{y}_i - y_i|$, to assess the performance of different metamodels vs. the detailed DNDC-EUROPE model in order to select the best statistical approach to approximate the complex bio-geo-chemical model at a lower computational cost. Defining a correct metamodeling strategy is very important to provide an adequate fitting to the model, as suggested by (Kleijnen and Sargent, 2000; Meckesheimer et al., 2002).

Recent work, such as (Forrester and Keane, 2009; Wang and Shan, 2007), review the most widely used metamodeling methods: splines based methods (e.g., MARS, kriging...) (Wahba, 1990; Friedman, 1991; Cressie, 1990), neural networks (Bishop, 1995), kernel methods (SVM, SVR...) (Vapnik, 1998; Christmann and Steinwart, 2007), Gaussian Process such as GEM (Kennedy and O'Hagan, 2001), among others. Some of these metamodeling strategies were selected and others added to be compared in this paper. The comparison is made on a specific case study related to N leaching and N_2O fluxes prediction which is described in Section 4. The next section briefly describes each of the metamodels compared in this paper.

3. Review of the selected metamodels

Several methods were developed and compared to assess their performance according to increasing dataset sizes. We provide a brief description

of the approaches studied in this paper: two linear models (Section 3.1) and six nonparametric methods (two based on splines, in Sections 3.2.1 and 3.2.2, one based on a kriging approach, in Section 3.2.3, which is known to be efficient when analyzing computer experiments, a neural network method, in Section 3.2.4, SVM, in Section 3.2.5 and random forest, in Section 3.2.6).

3.1. Linear methods

The easiest way to handle the estimation of the model given in Eq. (1) is to suppose that f has a simple parametric form. For example, the *linear model* supposes that $f(\mathbf{x}) = \boldsymbol{\beta}^T\mathbf{x} + \beta_0$ where $\boldsymbol{\beta} \in \mathbb{R}^d$ is a vector and β_0 is a real number, both of them have to be estimated from the observations $((\mathbf{x}_i, y_i))_i$. An estimate is given by minimizing the sum of the square errors

$$\sum_{i=1}^{N} \left(y_i - (\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0)\right)^2$$

which leads to $\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$ and $\hat{\beta}_0 = \overline{y} - \hat{\boldsymbol{\beta}}^T\overline{\mathbf{X}}$ with $\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$ and $\overline{\mathbf{X}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$.

In this paper two linear models were used:

- in the first one, the explanatory variables were the 11 inputs described in Section 4.2. This model is referred as"LM1";
- the second one has been developed starting from the work of (Britz and Leip, 2009), that includes the 11 inputs of Section 4.2 but also their non linear transformations (square, square root, logarithm) and interaction components. A total of 120 coefficients were involved in this approach which is denoted by "LM2". Including transformations and combinations of the 11 inputs has been designed in an attempt to better model a possible nonlinear phenomenon of the original model.

In the second case, due to the large number of explanatory variables, the model can be over-specified, especially if the training set is small. Actually, if the dimensionality of the matrix of explanatory variables, $\mathbf{X}$, has a large dimension, $\mathbf{X}^T\mathbf{X}$ can be not invertible or ill-conditioned (leading to numerical instability). Hence, a stepwise selection based on the AIC criterion (Akaike, 1974) has been used to select an optimal subset of explanatory variables during the training step in order to obtain an accurate solution having a small number of parameters. This has been performed by using the `stepAIC` function of the **R** package `MASS`.

3.2. Nonparametric methods

In many modeling problems, linear methods are not enough to catch the complexity of the phenomenon which is, *per se*, nonlinear. In these situations, nonparametric are often more suited to obtain accurate approximations of the phenomenon under study. In this section, six nonparametric approaches are described: they are compared in Section 4 to model N_2O fluxes and N leaching.

3.2.1. ACOSSO

Among nonparametric estimation approach, the smoothing splines (Wahba, 1990; Gu, 2002) is one of the most famous and widely used. Recently, (Storlie et al., 2011) presented the ACOSSO, an adaptive approach based on the COSSO method (Lin and Zhang, 2006) which is in the same line as smoothing splines: it is described as "a new regularization method for simultaneous model fitting and variable selection in nonparametric regression models in the framework of smoothing spline ANOVA". This method penalizes the sum of component norms, instead of the squared norm employed in the traditional smoothing spline method. More precisely, in splines metamodeling, it is useful to consider the ANOVA decomposition of f into terms of increasing dimensionality:

$$f(\mathbf{x}) = f(x^1, x^2, \ldots, x^d) = f_0 + \sum_j f^{(j)} + \sum_{k>j} f^{(jk)} + \ldots + f^{(12\ldots d)} \qquad (2)$$

where x^j is the *j-th* explanatory variable and where each term is a function only of the factors in its index, i.e. $f^{(j)} = f(x^j)$, $f^{(jk)} = f(x^j, x^k)$ and so on. The terms $f^{(j)}$ represent the additive part of the model f, while all higher order terms $f^{(jk)} \ldots f^{(12\ldots d)}$ are denoted as "interactions". The simplest example of smoothing spline ANOVA model is the additive model where only $(f^{(j)})_{j=0,\ldots,d}$ are used.

To estimate f, we make the usual assumption that $f \in \mathcal{H}$, where $\mathcal{H}$ is a RKHS (Reproducing Kernel Hilbert Space) (Berlinet and Thomas-Agnan, 2004). The space $\mathcal{H}$ can be written as an orthogonal decomposition $\mathcal{H} = \{1\} \oplus \{\bigoplus_{j=1}^{q} \mathcal{H}_j\}$, where each $\mathcal{H}_j$ is itself a RKHS, $\oplus$ is the direct sum of Hilbert spaces and $j = 1, \ldots, q$ spans ANOVA terms of various orders. Typically q includes the main effects plus relevant interaction terms. f is then estimated by $\hat{f}$ that minimizes a criterion being a trade-off between accuracy to the data (empirical mean squared error) and a penalty which

aims at minimizing each ANOVA term:

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}(\mathbf{x}_i))^2 + \lambda_0 \sum_{j=1}^{q} \frac{1}{\theta_j}\|P^j \hat{f}\|^2_{\mathcal{H}} \tag{3}$$

where $P^j \hat{f}$ is the orthogonal projection of $\hat{f}$ onto $\mathcal{H}_j$ and the q-dimensional vector θ_j of smoothing parameters needs to be tuned somehow, in such a way that each ANOVA component has the most appropriate degree of smoothness.

This statistical estimation problem requires the tuning of the d hyper-parameters θ_j (λ_0/θ_j are also denoted as smoothing parameters). Various ways of doing that are available in the literature, by applying generalized cross-validation (GCV), generalized maximum likelihood procedures (GML) and so on (Wahba, 1990; Gu, 2002). But, in Eq. (3), q is often large and the tuning of all θ_j is a formidable problem, implying that in practice the problem is simplified by setting θ_j to 1 for any j and only λ_0 is tuned. This simplification, however, strongly limits the flexibility of the smoothing spline model, possibly leading to poor estimates of the ANOVA components.

Problem (3) also poses the issue of selection of $\mathcal{H}_j$ terms: this is tackled rather effectively within the COSSO/ACOSSO framework. The COSSO (Lin and Zhang, 2006) penalizes the sum of norms, using a LASSO type penalty (Tibshirani, 1996) for the ANOVA model: LASSO penalties are L_1 penalties that lead to sparse parameters (i.e., parameters whose coordinates are all equal to zero except for a few ones). Hence, using this kind of penalties allows us to automatically select the most informative predictor terms $\mathcal{H}_j$ with an estimate of $\hat{f}$ that minimizes

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}(\mathbf{x}_i))^2 + \lambda \sum_{j=1}^{Q} \|P^j \hat{f}\|_{\mathcal{H}} \tag{4}$$

using a single smoothing parameter λ, and where Q includes *all* ANOVA terms to be potentially included in $\hat{f}$, e.g. with a truncation at 2^{nd} or 3^{rd} order interactions.

It can be shown that the COSSO estimate is also the minimizer of

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}(\mathbf{x}_i))^2 + \sum_{j=1}^{Q} \frac{1}{\theta_j}\|P^j \hat{f}\|^2_{\mathcal{H}} \tag{5}$$

subject to $\sum_{j=1}^{Q} 1/\theta_j < M$ (where there is a 1-1 mapping between M and λ). So we can think of the COSSO penalty as the traditional smoothing spline penalty plus a penalty on the Q smoothing parameters used for each component. This can also be framed into a linear-quadratic problem, i.e. a quadratic objective (5) plus a linear constraint on $1/\theta_j$. The LASSO type penalty has the effect of setting some of the functional components ($\mathcal{H}_j$'s) equal to zero (e.g. some variables x^j and some interactions (x^j, x^k) are not included in the expression of $\hat{f}$). Thus it "automatically" selects the appropriate subset q of terms out of the Q "candidates". The key property of COSSO is that with one single smoothing parameter (λ or M) it provides estimates of all θ_j parameters in one shot: therefore it improves considerably the simplified problem (3) by setting $\theta_j = 1$ (still with one single smoothing parameter λ_0) and is much more computationally efficient than the full problem (3) with optimized θ_j's. An additional improvement from the COSSO is that the single smoothing parameter λ can be tuned to minimize the BIC (Bayesian Information Criterion) (Schwarz, 1978), thus allowing to target the most appropriate degree of parsimony of the metamodel. This is done by a simple grid-search algorithm as follows (see (Lin and Zhang, 2006) for details):

1. for each trial λ value, the COSSO estimate provides the corresponding values for θ_j and subsequently its BIC;
2. the grid-search algorithm will provide the $\hat{\lambda}$ with the smallest BIC.

The adaptive COSSO (ACOSSO) of (Storlie et al., 2011) is an improvement of the COSSO method: in ACOSSO, $\hat{f} \in \mathcal{H}$ minimizes

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}(\mathbf{x}_i))^2 + \lambda \sum_{j=1}^{q} w_j \|P^j \hat{f}\|_{\mathcal{H}} \tag{6}$$

where $0 < w_j \leq \infty$ are weights that depend on an initial estimate, $\hat{f}^{(0)}$, of f, either using (3) with $\theta_j = 1$ or the COSSO estimate (4). The adaptive weights are obtained as $w_j = \|P^j \hat{f}^{(0)}\|_{L_2}^{-\gamma}$, typically with $\gamma = 2$ and the L_2 norm $\|P^j \hat{f}^{(0)}\|_{L_2} = (\int (P^j \hat{f}^{(0)}(\mathbf{x}))^2 d\mathbf{x})^{1/2}$. The use of adaptive weights improves the predictive capability of ANOVA models with respect to the COSSO case: in fact it allows for more flexibility in estimating important functional components while giving a heavier penalty to unimportant functional components. The **R** scripts for ACOSSO can be found

at `http://www.stat.lanl.gov/staff/CurtStorlie/index.html`. In the present paper we used a MATLAB translation of such R script. The algorithm for tuning the hyper-parameters is then modified as follows:

1. an initial estimate of the ANOVA model $\hat{f}^{(0)}$ is obtained either using (3) with $\theta_j = 1$ or the COSSO estimate (4);
2. given this trial ANOVA model $\hat{f}^{(0)}$, the weights are computed as $w_j = \|P^j \hat{f}^{(0)}\|_{L_2}^{-\gamma}$;
3. given w_j and for each trial λ value, the ACOSSO estimate (6) provides the corresponding values for θ_j and subsequently its BIC;
4. the grid-search algorithm will provide the $\hat{\lambda}$ with the smallest BIC.

3.2.2. SDR-ACOSSO

In a "parallel" stream of research with respect to COSSO-ACOSSO, using the *state-dependent parameter regression* (SDR) approach of (Young, 2001), (Ratto et al., 2007) have developed a non-parametric approach, very similar to smoothing splines and kernel regression methods, based on recursive filtering and smoothing estimation (the Kalman filter combined with "fixed interval smoothing"). Such a recursive least-squares implementation has some key characteristics: (a) it is combined with optimal maximum likelihood estimation, thus allowing for an estimation of the smoothing hyper-parameters based on the estimation of a quality criterion rather than on cross-validation and (b) it provides greater flexibility in adapting to local discontinuities, heavy non-linearity and heteroscedastic error terms. Recently, (Ratto and Pagano, 2010) proposed a unified approach to smoothing spline ANOVA models that combines the best of SDR and ACOSSO: the use of the recursive algorithms in particular can be very effective in *identifying* the important functional components and in providing good estimates of the weights w_j to be used in (6), adding valuable information in the ACOSSO framework and allowing in many cases to improving ACOSSO performance. The Matlab script for this method can be found at `http://eemc.jrc.ec.europa.eu/Software-SS_ANOVA_R.htm`.

We summarize here the key features of Young's recursive algorithms of SDR, by considering the case of $d = 1$ and $f(x^1) = f^{(1)}(x^1) + e$, with $e \sim N(0, \sigma^2)$. To do so, we rewrite the smoothing problem as $y_i = s_i^1 + e_i$, where $i = 1, \ldots, N$ and s_i^1 is the estimate of $f^{(1)}(x_i^1)$. To make the recursive approach meaningful, the MC sample needs to be sorted in ascending order

with respect to x^1: i.e. i and $i-1$ subscripts are adjacent elements under such ordering, implying $x_1^1 < x_2^1 < \ldots < x_i^1 < \ldots < x_N^1$.

To recursively estimate the s_i^1 in SDR it is necessary to characterize it in some stochastic manner, borrowing from non-stationary time series processes (Young and Ng, 1989; Ng and Young, 1990). In the present context, the integrated random walk (IRW) process provides the same smoothing properties of a cubic spline, in the overall State-Space formulation:

$$\begin{array}{lrcl} \text{Observation Equation:} & y_i & = & s_i^1 + e_i \\ \text{State Equations:} & s_i^1 & = & s_{i-1}^1 + d_{i-1}^1 \\ & d_i^1 & = & d_{i-1}^1 + \eta_i^1 \end{array} \tag{7}$$

where d_i^1 is the "slope" of s_i^1, $\eta_i^1 \sim N(0, \sigma_{\eta^1}^2)$ and η_i^1 ("system disturbance" in systems terminology) is assumed to be independent of the "observation noise" $e_i \sim N(0, \sigma^2)$.

Given the ascending ordering of the MC sample, s_i^1 can be estimated by using the recursive Kalman Filter (KF) and the associated recursive Fixed Interval Smoothing (FIS) algorithm (see e.g. (Kalman, 1960; Young, 1999) for details). First, it is necessary to optimize the hyper-parameter associated with the state space model (7), namely the Noise Variance Ratio (NVR), where $\text{NVR}_1 = \sigma_{\eta^1}^2/\sigma^2$. This is accomplished by maximum likelihood optimization (ML) using prediction error decomposition (Schweppe, 1965). The NVR plays the inverse role of a smoothing parameter: the smaller the NVR, the smoother the estimate of s_i^1. Given the NVR, the FIS algorithm then yields an estimate $\hat{s}_{i|N}^1$ of s_i^1 at each data sample and it can be seen that the $\hat{s}_{i|N}^1$ from the IRW process is the equivalent of $\hat{f}^{(1)}(x_i^1)$ in the cubic smoothing spline model. At the same time, the recursive procedures provide, in a natural way, standard errors of the estimated $\hat{s}_{i|N}^1$, that allow for the testing of their relative significance. Finally, it can be easily verified (Ratto and Pagano, 2010) that by setting $\lambda/\theta_1 = 1/(\text{NVR}_1 \cdot N^4)$, and with evenly spaced x_i^1 values, the $\hat{f}^{(1)}(x_i^1)$ estimate in the cubic smoothing spline model equals the $\hat{s}_{i|N}^1$ estimate from the IRW process.

The most interesting aspect of the SDR approach is that it is not limited to the univariate case, but can be effectively extended to the most relevant multivariate one. In the general additive case, for example, the recursive procedure needs to be applied, in turn, for each term $f^{(j)}(x_i^j) = \hat{s}_{i|N}^j$, requiring a different sorting strategy for each $\hat{s}_{i|N}^j$. Hence the "back-fitting" procedure is applied, as described in (Young, 2000) and (Young, 2001). This procedure

provides both ML estimates of all NVR_j's and the smoothed estimates of the additive terms $\hat{s}^j_{i|N}$. So, the estimated NVR_j's can be converted into λ_0/θ_j values using $\lambda_0/\theta_j = 1/(\text{NVR}_j \cdot N^4)$, allowing us to put the additive model into the standard cubic spline form.

In the SDR context, (Ratto and Pagano, 2010) formalized an interaction function as the product of two states $s_1 \cdot s_2$, each of them characterized by an IRW stochastic process. Hence the estimation of a single interaction term $f(\mathbf{x}_i) = f^{(12)}(x_i^1, x_i^2) + e_i$ is expressed as:

$$\begin{array}{lrcl} \text{Observation Equation:} & y_i^* & = & s^I_{1,i} \cdot s^I_{2,i} + e_i \\ \text{State Equations: } (j=1,2) & s^I_{j,i} & = & s^I_{j,i-1} + d^I_{j,i-1} \\ & d^I_{j,i} & = & d^I_{j,i-1} + \eta^I_{j,i} \end{array} \tag{8}$$

where y^* is the model output after having taken out the main effects, $I = 1, 2$ is the multi-index denoting the interaction term under estimation and $\eta^I_{j,i} \sim N(0, \sigma^2_{\eta^I_j})$. The two terms $s^I_{j,i}$ are estimated iteratively by running the recursive procedure in turn.

The SDR recursive algorithms are usually very efficient in identifying in the most appropriate way each ANOVA component individually, hence (Ratto and Pagano, 2010) proposed to exploit this in the ACOSSO framework as follows.

We define $\mathcal{K}_{\langle j \rangle}$ to be the reproducing kernel (r.k.) of an additive term $\mathcal{F}_j$ of the ANOVA decomposition of the space $\mathcal{F}$. In the cubic spline case, this is constructed as the sum of two terms $\mathcal{K}_{\langle j \rangle} = \mathcal{K}_{01\langle j \rangle} \oplus \mathcal{K}_{1\langle j \rangle}$ where $\mathcal{K}_{01\langle j \rangle}$ is the r.k. of the parametric (linear) part and $\mathcal{K}_{1\langle j \rangle}$ is the r.k. of the purely non-parametric part. The second order interaction terms are constructed as the tensor product of the first order terms, for a total of four elements, i.e.

$$\begin{aligned} \mathcal{K}_{\langle i,j \rangle} &= (\mathcal{K}_{01\langle i \rangle} \oplus \mathcal{K}_{1\langle i \rangle}) \otimes (\mathcal{K}_{01\langle j \rangle} \oplus \mathcal{K}_{1\langle j \rangle}) \\ &= (\mathcal{K}_{01\langle i \rangle} \otimes \mathcal{K}_{01\langle j \rangle}) \oplus (\mathcal{K}_{01\langle i \rangle} \otimes \mathcal{K}_{1\langle j \rangle}) \oplus (\mathcal{K}_{1\langle i \rangle} \otimes \mathcal{K}_{01\langle j \rangle}) \oplus (\mathcal{K}_{1\langle i \rangle} \otimes \mathcal{K}_{1\langle j \rangle}) \end{aligned} \tag{9}$$

This suggested that a natural use of the SDR identification and estimation in the ACOSSO framework is to apply specific weights to each element of the r.k. $\mathcal{K}_{\langle \cdot,\cdot \rangle}$ in (9). In particular the weights are the L_2 norms of each of the four elements estimated in (8):

$$\hat{s}^I_i \cdot \hat{s}^I_j = \hat{s}^I_{01\langle i \rangle} \hat{s}^I_{01\langle j \rangle} + \hat{s}^I_{01\langle i \rangle} \hat{s}^I_{1\langle j \rangle} + \hat{s}^I_{1\langle i \rangle} \hat{s}^I_{01\langle j \rangle} + \hat{s}^I_{1\langle i \rangle} \hat{s}^I_{1\langle j \rangle}, \tag{10}$$

As shown in (Ratto and Pagano, 2010), this choice can lead to a significant improvement in the accuracy of ANOVA models with respect to the original

ACOSSO approach. Overall, the algorithm for tuning the hyper-parameters in the combined SDR-ACOSSO reads:

1. the recursive SDR algorithm is applied to get an initial estimate of each ANOVA term in turn (back-fitting algorithm);
2. the weights are computed as the L_2 norms of the parametric and non-parametric parts of the cubic splines estimates;
3. given w_j and for each trial λ value, the ACOSSO estimate (6) provides the corresponding values for θ_j and subsequently its BIC;
4. the grid-search algorithm will provide the $\hat{\lambda}$ with the smallest BIC.

3.2.3. Kriging metamodel: DACE

DACE (Lophaven et al., 2002) is a Matlab toolbox used to construct kriging approximation models on the basis of data coming from computer experiments. Once we have this approximate model, we can use it as a meta-model (emulator, surrogate model). We briefly highlight the main features of DACE. The kriging model can be expressed as a regression

$$\hat{f}(\mathbf{x}) = \beta_1 \phi^1(\mathbf{x}) + \cdots + \beta_q \phi^q(\mathbf{x}) + \zeta(\mathbf{x}) \tag{11}$$

where $\phi^j, j = 1, \ldots, q$ are deterministic regression terms (constant, linear, quadratic, etc.), β_j are the related regression coefficients and ζ is a zero mean random process whose variance depends on the process variance ω^2 and on the correlation $\mathcal{R}(v, w)$ between $\zeta(v)$ and $\zeta(w)$. In kriging, correlation functions are typically used, defined as:

$$\mathcal{R}(\theta, v - w) = \prod_{j=1:d} \mathcal{R}_j(\theta_j, w_j - v_j).$$

In particular, for the generalized exponential correlation function, used in the present paper, one has

$$\mathcal{R}_j(\theta_j, w_j - v_j) = \exp(-\theta_j |w_j - v_j|^{\theta_{d+1}})$$

Then, we can define $\mathbf{R}$ as the correlation matrix at the training points (i.e., the matrix with coordinates $r_{i,j} = \mathcal{R}(\theta, \mathbf{x}_i, \mathbf{x}_j)$) and the vector $\mathbf{r}_\mathbf{x} = [\mathcal{R}(\theta, \mathbf{x}_1, \mathbf{x}), \ldots, \mathcal{R}(\theta, \mathbf{x}_N, \mathbf{x})]$, $\mathbf{x}$ being an untried point. Similarly, we define the vector $\boldsymbol{\phi}_\mathbf{x} = [\phi^1(\mathbf{x}) \ldots \phi^q(\mathbf{x})]^T$ and the matrix $\boldsymbol{\Phi} = [\boldsymbol{\phi}_{\mathbf{x}_1} \cdots \boldsymbol{\phi}_{\mathbf{x}_N}]^T$ (i.e.,

$\boldsymbol{\Phi}$ stacks in matrix form all values of $\boldsymbol{\phi}_{\mathbf{x}}$ at the training points). Then, considering the linear regression problem $\boldsymbol{\Phi}\boldsymbol{\beta} \approx \mathbf{y}$ coming from Eq. (11), with parameter $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_q]^T \in \mathbb{R}^q$, the GLS solution is given by:

$$\boldsymbol{\beta}^* = (\boldsymbol{\Phi}^T \mathbf{R}^{-1} \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{R}^{-1} \mathbf{y}$$

which gives the predictor at untried $\mathbf{x}$

$$\hat{f}(\mathbf{x}) = \boldsymbol{\phi}_{\mathbf{x}}^T \boldsymbol{\beta}^* + \mathbf{r}_{\mathbf{x}}^T \boldsymbol{\gamma}^*,$$

where $\boldsymbol{\gamma}^*$ is the N-dimensional vector computed as $\boldsymbol{\gamma}^* = \mathbf{R}^{-1}(\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta}^*)$.

The proper estimation of the kriging metamodel requires, of course, to optimize the hyper-parameters θ in the correlation function: this is typically performed by maximum likelihood. It is easy to check that the kriging predictor *interpolates* $\mathbf{x}_j$, if the latter is a training point.

It seems useful to underline that one major difference between DACE and ANOVA smoothing is the absence of any "observation error" in (11). This is a natural choice when analyzing computer experiments and it aims to exploit the "zero-uncertainty" feature of this kind of data. This, in principle, makes the estimation of kriging metamodels very efficient, as confirmed by the many successful applications described in literature and justifies the great success of this kind of metamodels among practitioners. It also seems interesting to mention the so-called "nugget" effect, which is also used in the kriging literature (Montès, 1994; Kleijnen, 2009). This is nothing other than a "small" error term in (11) and it often reduces some numerical problems encountered in the estimation of the kriging metamodels to the form of (11). The addition of a nugget term leads to kriging metamodels that smooth, rather than interpolate, making them more similar to other metamodels presented here.

3.2.4. Multilayer perceptron

"Neural network" is a general name for statistical methods dedicated to data mining. They comprise of a combination of simple computational elements (neurons or nodes) densely interconnected through synapses. The number and organization of the neurons and synapses define the network topology. One of the most popular neural network class is the "multilayer perceptrons" (MLP) commonly used to solve a wide range of classification and regression problems. In particular, MLP are known to be able to approximate any (smooth enough) complex function (Hornik, 1991). Perceptrons

were introduced at the end of the 50s by Rosenblatt but they started becoming very appealing more recently thanks to the soaring computational capacities of computers. The works of (Ripley, 1994) and (Bishop, 1995) provide a general description of these methods and their properties.

For the experiments presented in Section 4.3, one-hidden-layer perceptrons were used. They can be expressed as a function of the form

$$f_{\mathbf{w}} : \mathbf{x} \in \mathbb{R}^p \to g_1 \left(\sum_{i=1}^{Q} w_i^{(2)} g_2 \left(\mathbf{x}^T \mathbf{w}_i^{(1)} + w_i^{(0)} \right) + w_0^{(2)} \right)$$

where:

- $\mathbf{w} := \left[(w_i^{(0)})_i, ((\mathbf{w}_i^{(1)})^T)_i, w_0^{(2)}, (w_i^{(2)})_i \right]^T$ are parameters of the model, called *weights*. They have to be learned in $(\mathbb{R})^Q \times (\mathbb{R}^p)^Q \times \mathbb{R} \times (\mathbb{R})^Q$ during the training;
- Q is a hyper-parameter indicating the number of neurons on the hidden layer;
- g_1 and g_2 are the activation functions of the neural networks. Generally, in regression cases (when the outputs to be predicted are real values rather than classes), g_1 is the identity function (hence the outputs are a linear combination of the neurons on the hidden layer) and g_2 is the logistic activation function $z \to \frac{e^z}{1+e^z}$.

The weights are learned in order to minimize the mean square error on the training set:

$$\hat{\mathbf{w}} := \arg\min \sum_{i=1}^{n} \|y_i - f_{\mathbf{w}}(\mathbf{x}_i)\|^2. \tag{12}$$

Unfortunately this error is not a quadratic function of w and thus no exact algorithm is available to find the global minimum of this optimization problem (and the existence of such a global minimum is not even guaranteed). Gradient descent based approximation algorithms are usually computed to find an approximate solution, where the gradient of $\mathbf{w} \to f_{\mathbf{w}}(\mathbf{x}_i)$ is calculated by the back-propagation principle (Werbos, 1974).

Moreover, to avoid overfitting, a penalization strategy, called *weight decay* (Krogh and Hertz, 1992), was introduced. It consists of replacing the

minimization problem (12) by its penalized version:

$$\hat{\mathbf{w}} := \arg\min \sum_{i=1}^{n} \|y_i - f_{\mathbf{w}}(\mathbf{x}_i)\|^2 + C\|\mathbf{w}\|^2$$

where C is the penalization parameter. The solution of this penalized mean square error is designed to be smoother than that given by Eq. (12). The `nnet` **R** function, provided in the **R** package `nnet` (Venables and Ripley, 2002), was used to train and test the one-hidden-layer MLP. As described in Section 4.3, a single validation approach was used to tune the hyper-parameters Q and C which were selected on a grid search ($Q \in \{10, 15, 20, 25, 30\}$ and $C \in \{0, 0.1, 1, 5, 10\}$).

3.2.5. SVM (Support Vector Machines)

SVM were introduced by (Boser et al., 1992) originally to address classification problems. Subsequently (Vapnik, 1995) presented an application to regression problems to predict dependent real valued variables from given inputs. In SVM, the estimate $\hat{f}$ is chosen among the family of functions

$$f : \mathbf{x} \in \mathbb{R}^d \to \langle \mathbf{w}, \phi(\mathbf{x}) \rangle_{\mathcal{H}} + b$$

where ϕ is a function from $\mathbb{R}^d$ into a given Hilbert space $(\mathcal{H}, \langle ., . \rangle_{\mathcal{H}})$, here a RKHS, $\mathbf{w} \in \mathcal{H}$ and $b \in \mathbb{R}$ are parameters to be learned from the training dataset. Despite several strategies were developed to learn the parameters $\mathbf{w}$ and b (Steinwart and Christmann, 2008), we opted for the original approach which consists of using the *ϵ-insensitive* loss function as a quality criterion for the regression:

$$L_\epsilon(\mathbf{X}, \mathbf{y}, \hat{f}) = \sum_{i=1}^{N} \max\left(|\hat{f}(\mathbf{x}_i) - y_i| - \epsilon, 0\right).$$

This loss function has the property to avoid considering the error when it is small enough (smaller than ϵ). His main interest, compared to the usual squared error, is its robustness (see (Steinwart and Christman, 2008) for a discussion). The SVM regression is based on the minimization of this loss function on the learning sample while penalizing the complexity of the obtained $\hat{f}$. More precisely, the idea of SVM regression is to find $\mathbf{w}$ and b solutions of:

$$\arg\min_{\mathbf{w},b} L_\epsilon(\mathbf{X}, \mathbf{y}, \hat{f}) + \frac{1}{C}\|\mathbf{w}\|^2_{\mathcal{H}} \tag{13}$$

where the term $\|w\|^2_{\mathcal{H}}$ is the regularization term that controls the complexity of $\hat{f}$ and C is the regularization parameter: when C is small, $\hat{f}$ is allowed to make bigger errors in favor of a smaller complexity; if the value of C is high, $\hat{f}$ makes (almost) no error on the training data but it could have a large complexity and thus not be able to give good estimations for new observations (e.g., those of the test set). A good choice must devise a compromise between the accuracy required by the project and an acceptable metamodel complexity.

(Vapnik, 1995) demonstrates that, using the Lagrangian and Karush-Kuhn-Tucker conditions, w takes the form

$$\mathbf{w} = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \phi(\mathbf{x}_i)$$

where α_i and α_i^* solve the so-called *dual optimization problem*:

$$\begin{aligned} \arg\max_{\alpha_i, \alpha_i^*} \Bigg(& -\frac{1}{2} \sum_{i,j=1}^{N} (\alpha_i - \alpha_i^*)(\alpha_i - \alpha_i^*) \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}} \\ & -\epsilon \sum_{i=1}^{N} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i (\alpha_i - \alpha_i^*) \Bigg) \\ \text{subject to: } & \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C]. \end{aligned} \tag{14}$$

This is a classical quadratic optimization problem that can be explicitly solved. (Keerthi et al., 2001) provide a detailed discussion on the way to compute b once w is found; for the sake of clarity, in this paper we skip the full description of this step.

In Eq. (14), ϕ is only used through the dot products $(\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}})_{i,j}$. Hence, ϕ is never explicitly given but only accessed through the dot product by defining a kernel, $\mathcal{K}$:

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}. \tag{15}$$

This is the so-called *kernel trick*. As long as $\mathcal{K} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is symmetric and positive, it is ensured that an underlying Hilbert space $\mathcal{H}$ and an underlying $\phi : \mathbb{R}^d \to \mathcal{H}$ exist satisfying the relation of Eq. (15). The very

common *Gaussian kernel*, $\mathcal{K}_\gamma(u,v) = e^{-\gamma\|u-v\|^2}$ for a $\gamma > 0$, was used in the simulations.

Finally, three hyper-parameters have to be tuned to use SVM regression:

- ϵ of the loss function;
- C, the regularization parameter of the SVM;
- γ, the parameter of the Gaussian kernel.

As described in Section 4.3, a single validation approach was used to tune the hyper-parameters C and γ which were selected on a grid search ($C \in \{10, 100, 1000, 2000\}$ and $\gamma \in \{0.001, 0.01, 0.1\}$). To reduce the computational costs and also to limit the number of hyperparameters to the same value as in MLP case (and thus to prevent the global method from being too flexible), we avoided tuning ϵ by setting it equal to 1, which corresponds approximately to the second decile of the target variable for each scenario and output. This choice fitted the standard proposed by (Mattera and Haykin, 1998) which suggests having a number of Support Vectors smaller than 50% of the training set. Simulations were done by using the function `svm` from the **R** package `e1071` based on the libsvm library (Chang and Lin, 2001).

3.2.6. Random Forest

Random forests (RF) were first introduced by (Breiman, 2001) on the basis of his studies on bagging and of the works of (Amit and Geman, 1997; Ho, 1998) on features selection. Basically, bagging consists of computing a large number of elementary regression functions and of averaging them. In random forest, elementary regression functions involved in the bagging procedure are regression trees (Breiman et al., 1984). Building a regression tree aims at finding a series of *splits* deduced from one of the d variables, x^k (for a $k \in \{1, \ldots, d\}$), and a threshold, τ, that divides the training set into two subsamples, called *nodes*: $\{i : \ x_i^k < \tau\}$ and $\{i : x_i^k \geq \tau\}$. The split of a given node, $\mathcal{N}$, is chosen, among all the possible splits, by minimizing the sum of the *homogeneity* of the two corresponding child nodes, $\mathcal{N}_c^1$ and $\mathcal{N}_c^2$, as follows:

$$\sum_{i \in \mathcal{N}_c^i} \left(y_i - \bar{y}^{\mathcal{N}_c^i} \right)^2$$

where $\bar{y}^{\mathcal{N}_c^i} = \frac{1}{|\mathcal{N}_c^i|} \sum_{i \in \mathcal{N}_c^i} y_i$ is the mean value of the output variable for the observations belonging to $\mathcal{N}_c^i$ (i.e., the intra-node variance).

The growth of the tree stops when the child nodes are homogeneous enough (for a previously fixed value of homogeneity) or when the number of observations in the child nodes is smaller than a fixed number (generally chosen between 1 and 5). The prediction obtained for new inputs, $\mathbf{x}$, is then simply the mean of the outputs, y_i, of the training set that belong to the same terminal node (a leaf). The pros of this method are its easy readability and interpretability; the main drawback is its limited flexibility, especially for regression problems. To overcome this limit, random forests combine a large number (several hundreds or several thousands) of regression trees, T. In the forest, each tree is built sticking to the following algorithm that is made of random perturbations of the original procedure to make the tree under-efficient (i.e., so that none of the tree in the forest is the optimal one for the training dataset):

1. A given number of observations, m, are randomly chosen from the training set: this subset is called *in-bag* sample whereas the other observations are called *out-of-bag* and are used to check the error of the tree;
2. For each node of the tree, a given number of variables, q, are randomly selected among all the possible explanatory variables. The best split is then calculated on the basis of these q variables for the m chosen observations.

All trees in the forest are fully learned: the final leafs all have homogeneity equal to 0. Once having defined the T regression trees, $\mathcal{T}_1, \ldots, \mathcal{T}_T$, the regression forest prediction for new input variables, $\mathbf{x}$, is equal to the mean of the individual predictions obtained by each tree of the forest for $\mathbf{x}$.

Several hyper-parameters can be tuned for random forests such as the number of trees in the final forest or the number of variables randomly selected to build a given split. But, as this method is less sensitive to parameter tuning than the other ones (i.e., SVM and MLP), we opted for leaving the default values implemented in the **R** package `randomForest` based on useful heuristics: 500 trees were trained, each defined from a bootstrap sample built with replacement and having the size of the original dataset. Each node was defined from three randomly chosen variables and the trees were grown until the number of observations in each node was smaller than five. Moreover, the full learning process always led to a stabilized out-of-bag error.

4. Simulations and results

4.1. Application to the Cross Compliance Assessment Tool

As described above in the Section 1, the impact assessment of Cross Compliance (CC) measures on the EU27 farmlands, required the development of a simulation platform called Cross Compliance Assessment Tool (CCAT). The CCAT framework integrates different models, such as Miterra (Velthof et al., 2009), DNDC-EUROPE (Follador et al., 2011), EPIC (van der Velde et al., 2009) and CAPRI (Britz and Witzke, 2008; Britz, 2008), in order to guarantee an exhaustive evaluation of the effects of agro-environmental standards for different input, scenario assumptions, compliance rates and space-time resolutions (Elbersen et al., 2010; De Vries et al., 2008). The simulated outputs are indicators for nitrogen (N) and carbon (C) fluxes, biodiversity and landscape, market response and animal welfare. The selection of the CC scenarios as well as the definition of the environmental indicators to be considered in this project, are described by (Jongeneel et al., 2008). The CCAT tool evaluates the effect of agricultural measures on N_2O fluxes and N leaching by means of the meta-model of the mechanistic model DNDC-EUROPE (Follador et al., 2011). N_2O is an important greenhouse gas (Intergovernmental Panel on Climate Change, 2007). Agriculture and in particular agricultural soils are contributing significantly to anthropogenic N_2O emissions (European Environment Agency, 2010). N_2O fluxes from soils are characterized by a high spatial variability and the accuracy of estimates can be increased if spatially explicit information is taken into consideration (Leip et al., 2011a). Similarly, leaching of nitrogen from agricultural soils is an important source of surface and groundwater pollution (European Environment Agency, 1995).

The main limits of using DNDC-EUROPE directly in the CCAT platform are the high computational costs and memory requirements, due to the large size of input datasets and the complexity and high number of equations to solve. To mitigate this problem, making the integration easier, we decided to develop a metamodel of DNDC-EUROPE (Follador and Leip, 2009). The choice of the best meta-modeling approach has been based on the analysis of performance of different algorithms, as described in details in Section 4.4. The best metamodel is expected to have low computational costs and an acceptable accuracy for all the dataset sizes.

4.2. Input and Output data description

The set of training observations (around 19 000 observations) used to define a metamodel $\hat{f}$ was created by linking the agro-economic CAPRI dataset with the bio-geochemical DNDC-EUROPE model at Homogeneous Spatial Mapping Unit (HSMU) resolution, as described in (Leip et al., 2008). We opted for corn cultivation as case study, since it covers almost 4.6% of UAA (utilized agricultural area) in EU27, playing an important role in human and animal food supply (European Union Commission, 2010)[1] and representing one of the main cropping system in Europe. To obtain a representative sample of situations for the cultivation of corn in EU27, we selected about 19,000 HSMUs on which at least 10% of the agricultural land was used for corn (Follador et al., 2011).

The input observations used to train the metamodels were drawn from the whole DNDC-EUROPE input database (Leip et al., 2008; Li et al., 1992; Li, 2000), in order to meet the need of simplifying the I/O flux of information between models in the CCAT platform. This screening was based on a preliminary sensitivity analysis of input data through the *importance function* of the **R** package `randomForest`, and subsequently it was refined by expert evaluations (Follador et al., 2011; Follador and Leip, 2009). At last, 11 input variables were used:

- Variable related to N input [kgN $ha^{-1}yr^{-1}$], such as mineral fertilizer (`N_FR`) and manure (`N_MR`) amendments, N from biological fixation (`Nfix`) and N in crop residue (`Nres`);
- variables related to soil: soil bulk density, `BD`, [g cm^{-3}], topsoil organic carbon, `SOC`, [mass fraction], clay content, `clay`, [fraction] and topsoil pH, `pH`;
- variables related to climate: annual precipitation `Rain`, [mm yr^{-1}], annual temperature `Tmean` [°C] and N in rain, `Nr`, [ppm].

They refer to the main driving forces taking part in the simulation of N_2O and N leaching with DNDC-EUROPE, such as farming practices, soil attributes and climate information. In this contribution we only show the results for the corn baseline scenario - that is the conventional corn cultivation in EU27,

[1] `http://epp.eurostat.ec.europa.eu`

as described by (Follador et al., 2011). Note that a single metamodel was developed for each CC scenario and for each simulated output in CCAT, as described in (Follador and Leip, 2009). Figure 1 summarizes the relations between the DNDC-EUROPE model and the metamodel.

As the number of input variables was not large, they were all used in all the metamodeling methods described in Section 3, without additional variable selection. The only exception is the second linear model (Section 3.1) which uses a more complete list of input variables obtained by various combinations of the original 11 variables and thus includes a variable selection process to avoid collinearity issues.

Two output variables were studied: the emissions of N_2O ([kg N yr^{-1} ha^{-1}] for each HSMU), a GHG whose reduction is a leading matter in climate change mitigation strategies, and the nitrogen leaching ([kg N yr^{-1} ha^{-1}] for each HSMU), which has to be monitored to meet the drinking water quality standards (Askegaard et al., 2005). A metamodel was developed for each single output variable. The flux of information through the DNDC-EUROPE model and its relationship with the metamodel's one are summarized in Figure 1. The data were extracted using a high performance computer cluster and the extraction process took more that one day for all the 19 000 observations.

[Figure 1 about here.]

4.3. Training, validation and test approach

The training observations were randomly partitioned (without replacement) into two groups: 80% of the observations (i.e., $N_L \simeq 15\ 000$ HSMU) were used for training (i.e., for defining a convenient $\hat{f}$) and the 20% remaining observations (i.e., $N_T \simeq 4\ 000$ HSMU) were used for validating the metamodels (i.e., for calculated an error score). Additionally, in order to understand the impact of the training dataset on the goodness of the estimations ($\hat{y}_i$) and to compare the different metamodel performance according to the data availability, we randomly selected from the entire training set a series of subsets, having respectively $N_L = 8\ 000$, 4 000, 2 000, 1 000, 500, 200 and 100 observations, each consecutive training subset being a subset of the previous one.

The methodology used to assess the behavior of different metamodels under various experimental conditions (size of the dataset and nature of the output) are summarized in Description 1.

Description 1 Methodology used to compare the metamodels under various experimental conditions

1: **for** Each metamodel, each output and each size N_L **do**
2: {**Train** the metamodel with the N_L training observations $\rightarrow$ definition of $\hat{f}$;
3: **Estimate the outputs** for the $N_T \simeq 4\,000$ inputs of the test set from $\hat{f} \rightarrow$ calculation of $\hat{y}_i$;
4: **Calculate the test error** by comparing the estimated outputs, $\hat{y}_i$, vs. the outputs of the DNDC-EUROPE model for the same test observations, y_i.}
5: **end for**

More precisely, for some metamodels, Step 2 requires the tuning of some hyper-parameters (e.g., SVM have three hyper-parameters, see Section 3). These hyper-parameters were tuned by:

- for ACOSSO and SDR: a grid-search to minimize BIC plus an algorithm to get the weights w_j: in these cases, an efficient formula, that does not require to compute each leave-one-out estimate of f, can be used to compute the BIC; moreover the COSSO penalty provides all θ_j given λ and w_j in a single shot. In the SDR identification steps, a maximum likelihood strategy is applied to optimize NVR's;

- for DACE, a maximum likelihood strategy;

- for MLP, SVM and RF, a *simple validation strategy* preferred to a cross validation strategy to reduce the computational time especially with the largest training datasets): half of the data were used to define several metamodels depending on the values of hyper-parameters on a grid search and the remaining data were used to select the best set of hyper-parameters by minimizing a mean square error criterion.

Hence, depending on which features are the most interesting (easy tuning of the hyperparameters, size of the training dataset, size of the dataset needing new prediction...), the use of one method is more or less recommended. Table 1 summarizes the main characteristics of the training and validation steps of each method as well as the characteristics to do new predictions. For instance, linear models are more

Table 1: Summary of the main features for training, validation (hyperparameters tuning) and test steps of each method.

Method	**Training characteristics**	**Validation characteristics**	**New predictions characteristics**
LM1	**Very fast** to train.	There is **no hyper parameter** to tune.	**Very fast**.
LM2	**Fast** to train but much slower than LM1 because of the number of parameters to learn.	There is **no hyper-parameter** to tune.	**Very fast**.
ACOSSO	**Fast to train only if the number of observations is very low**: the dimension of the kernel matrix is $N_L \times N_L$ and it is obtained as the sum of the kernels of each $[N_L \times N_L]$ ANOVA term, which can be long to calculate.	One hyper-parameter (λ) is tuned twice by minimizing BIC: the first time to get the weights w_j the second to get the final estimate (given λ and w_j the COSSO penalty provides automatically in a single shot all θ_j).	The time needed to obtain new predictions **can be high** depending on the sizes of both the training dataset and the test dataset. It requires to compute a kernel matrix having dimension $N_L \times N_T$.

Continued on next page

Table 1 – *Continued from previous page*

Method	Training characteristics	Validation characteristics	Test characteristics
SDR	**Fast to train only if the number of observations is very low**: the dimension of the kernel matrix is $N_L \times N_L$ and it is obtained as the sum of the kernels of each $[N_L \times N_L]$ ANOVA term, which can be long to calculate.	As for ACOSSO, the single hyper-parameter (λ) is tuned by minimizing BIC: the SDR identification step to provide w_j also optimizes hyper-parameters for each ANOVA component but this can be done **efficiently by the SDR recursive algorithms** (given λ and w_j the COSSO penalty provides automatically in a single shot all θ_j).	The time needed to obtain new predictions **can be high** depending on the sizes of both the training dataset and the test dataset. It requires to compute a kernel matrix having dimension $N_L \times N_T$.
DACE	**Fast to train only if the number of observations is very low**: the dimension of the kernel matrix is $N_L \times N_L$, and the inversion of a matrix $N_L \times N_L$ is required in the GLS procedure.	$d + 1$ hyper-parameters are tuned by ML, which becomes intractable already for moderate d: each step of the optimization a matrix $N_L \times N_L$ has to be inverted.	The time needed to obtain new predictions **can be high** depending on the sizes of both the training dataset and the test dataset. It requires to compute a kernel matrix having dimension $N_L \times N_T$.

Continued on next page

Table 1 – *Continued from previous page*

Method	Training characteristics	Validation characteristics	Test characteristics
MLP	**Hard to train**: because the error to minimize is not quadratic, the training step faces local minima problems and has thus to be performed several times with various initialization values. It is also very sensitive to the dimensionality of the data (that strongly increases the number of weights to train) and, to a lesser extent, to the number of observations.	2 hyperparameters have to be tuned but one is discrete (number of neurons on the hidden layer) which is easier. Nervelessness, cross validation is not suited: tuning is performed by simple validation and can thus be less accurate. It can be **time consuming**.	The time needed to obtain new predictions is **very low**: it depends on the number of predictions.
SVM	**Fast to train if the number of observations is low**: SVM are almost insensitive to the dimensionality of the data but the dimension of the kernel matrix is $N_L \times N_l$ and can be long to calculate.	Three hyperparameters have to be tuned and in the case where the size of the training dataset is large, cross validation is not suited. Tuning is performed by simple validation and can thus be less accurate. It is also **time consuming**.	The time needed to obtain new predictions **can be high** depending on the sizes of both the training dataset and the test dataset. It requires to compute a kernel matrix having dimension $N_L \times N_T$.

Continued on next page

Table 1 – *Continued from previous page*

Method	Training characteristics	Validation characteristics	Test characteristics
RF	**Fast to train**: **almost insensitive to the size or the dimensionality of the training dataset** thanks to the random selections of observations and variables. Most of the time needed to train is due to the number of trees required to stabilize the algorithm, that can sometimes be large.	Almost insensitive to hyperparameters so **no extensive tuning** is required.	The time needed to obtain new predictions is **low**: it depends on the number of predictions to do and also on the number of trees in the forest.

In Step 4, the test quality criterion was evaluated by calculating several quantities:

- the *Mean Squared Error* (MSE):

$$\text{MSE} = \frac{1}{N_T}\sum_{i=1}^{N_T}(\hat{y}_i - y_i)^2$$

  where y_i and $\hat{y}_i$ are, respectively, the model outputs in the test dataset and the corresponding approximated outputs given by the metamodel.

- the R^2 coefficient:

$$R^2 = 1 - \frac{\sum_{i=1}^{N_T}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N_T}(\hat{y}_i - \overline{y})^2} = 1 - \frac{MSE}{\text{Var}(y)}$$

  where $\overline{y}$ and $\text{Var}(y)$ are the mean and the variance of all y_i in the test dataset. R^2 is equal to 1 if the predictions are perfect and thus gives a way to quantify the accuracy of the predictions to the variability of the variable to predict.

- the *standard deviation of the SE* and the *maximum value of the SE* were also computed to give an insight on the variability of the performance and not only on its mean.

4.4. Results and discussion

This section includes several ways to compare the methods on the problem described in 4.2. First, Section 4.4.1 compares the accuracy of the predictions for various methods and various training dataset sizes. Then, Section 4.4.2 gives a comparison of the computational times needed either to train the model (with the maximum dataset size) and to make new predictions. Finally, Section 4.4.3 describes the model itself and gives an insight about its physical interpretation.

4.4.1. Accuracy

The performance on the test set is summarized in Tables 2 to 5: they include characteristics about the mean values of the squared errors (MSE and R^2) in Tables 2 and 3, respectively for N_2O and N leaching predictions, as well as characteristics related to the variability of the performance (standard deviations of the squared errors and maximum values of the squared errors) in Tables 4 and 5, respectively for N_2O and N leaching predictions. Note that, in almost all cases, the minimum values of the squared errors were equal or close to 0.

[Table 1 about here.]

[Table 2 about here.]

[Table 3 about here.]

[Table 4 about here.]

The evolution of R^2 on the test set in function of the size of the training set is displayed in Figures 2 (N_2O prediction) and 3 (N leaching) for each method.

[Figure 2 about here.]

[Figure 3 about here.]

From these results, several facts clearly appeared:

- Even for small datasets, the metamodeling approach behaves correctly with R^2 always greater than 80% for the best approaches. Note that the poorest results (those that are the closest to 80%) are obtained for small training dataset sizes (100 or 200). This means that, in the case where several metamodels are needed to model various assumptions of the input variables ranges, crude but acceptable estimates can be obtained at a very low computational cost. For more efficient predictions, larger datasets are more suited and achieve R^2 values greater than 90%.

- Predicting N leaching seems an easier task than predicting N_2O fluxes with greater performance for almost any training dataset size. This is not surprising because N_2O is generated as an intermediate product in the denitrification chain, being produced by the reduction of nitrate, but being consumed by N_2O denitrifiers. As a consequence, N_2O fluxes are the result of a fragile equilibrium between those processes which are both highly sensitive on environmental conditions such as pH, oxygen availability, substrate availability (Firestone et al., 1979). Thus, N_2O fluxes are characterized by a very high spatial variability and is much harder to predict than nitrogen leaching (Britz and Leip, 2009; Leip et al., 2011a).

- The best results are obtained for the largest training dataset. Mostly, for all methods, the performance increases with the size of the learning dataset despite some exceptions: sometimes, using a larger dataset makes the training process harder and can slightly deteriorate the performance (e.g., for MLP, large datasets leads to harder local minima problems in the optimization procedure: for this method, the best prediction of N leaching estimates is not obtained from the largest training set).

- In a similar way, the variability of the errors tends to decrease with the size of the training dataset but some methods behave differently (see, e.g., Acosso whose variability strictly increases with the size of the training dataset for N leaching prediction).

- In most cases, the most accurate predictions (according to MSE or R^2 values) are also the predictions that have the smallest variability either from the standard deviation point of view or from the smallest maximum point of view.

Looking deeper into the methods themselves, the following conclusions can also be derived:

- LM1 gives poor performance because the plain linear model is probably too simple to catch the complexity of the modeled phenomenon.

- LM2 performs very badly for small training datasets since it is over-specified (the number of parameters to be estimated is close to the size of the dataset; R^2 are negative which means that the model is less accurate than the trivial model predicting any observation by the mean value of the outputs). But for large training datasets, it behaved correctly. Additionally, the number of variables selected during the step AIC, in function of the training dataset size, is given in Table 6. The number of selected variables for N leaching prediction is higher than the number of selected variables for N_2O prediction but it also tends to be more stable regarding the dataset size. Also note that, in any case, the number of selected variables is high compared to the original number of variables (120): this means that the underlying model under study is certainly not plain linear and this explains why LM1 fails to approximate it accurately.

[Table 5 about here.]

- Splines and kriging based methods have the best performance for small and medium training datasets (especially for N leaching prediction) but they can not be run for large training datasets (up to 2 000 observations) due to the calculation costs. The Dace and SDR models have the best performance. Additionally, the number of selected variables for ACOSSO and SDR are given in Table 7. The number of components effectively included in the model tend decrease with the training set size, especially for N_2O prediction. Comparing this table with Table 6, the number of components is also quite small, even smaller than the number of original variables for some cases.

[Table 6 about here.]

- Machine learning methods (MLP, SVM and RF) behave correctly for medium training datasets and obtain the best performance for large training datasets. SVM and RF have the best results with a very good

overall accuracy, as, for these methods, R^2 are greater than 90% and 95%, respectively for N_2O and N leaching predictions.

Moreover, Wilcoxon paired tests on the residuals (absolute value) were computed to understand if the differences in accuracy between the best methods were significant: for N_2O prediction, the difference between the best performance (RF) and the second one (SVM) is significant (p-value equal to 0.16%) whereas, for N leaching prediction, the difference between the best performance (SVM) and the second one (RF) is not significant. This test confirms the differences between the best performance of metamodels obtained with different dataset sizes: for example, the difference between SVM trained with about 15 000 observations and Dace trained with 2 000 observations is significant (p-value smaller than $2.2 \cdot 10^{-16}$).

Finally, we took into account the time needed to train the metamodel and subsequently to use it for prediction. The time for training is not so important as it is spent only once during the calibration step. The time for prediction is a key point for CCAT project and so it played a leading role in choosing the best metamodel; it must be quite limited to allow fast multi-scenario simulations or sensitivity analysis. Table 8 provides the approximate time spent to train and use each method with large datasets (respectively, about 15 000 observations for the training step and about 19 000 observations for the prediction one) on a desktop computer.

[Table 7 about here.]

4.4.2. Computational time

The training time for LM1 was the best one but the corresponding performance is very poor. RF had a low training time since it does not require any parameter tuning and it is not very sensitive to the size of dataset thanks to the bootstrapping procedure. The prediction time is really low for all the methods compared to the DNDC-EUROPE runs which had demanded about 1 day to simulate the same outputs on a high performance computer cluster. Even though RF was not the fastest approach it provides the best compromise between speed and accuracy. SVM spent more time in prediction since it required the calculation of the kernel matrix whose size is proportional (and thus much more sensitive) to the number of new predictions to make. The same issues applies to splines approach, where the kernel matrix has to be re-computed for every ANOVA term in the decomposition, as well as for kriging, thus explaining the larger computational cost. The highest cost for

SDR predictions are linked to the more detailed decomposition, which implies a larger number of reproducing kernels. To compute the large amount of 19 000 model outputs, the time required for predictions does not exceed a few minutes in any cases.

4.4.3. Metamodeling interpretation

To give an indication of which variables are important in the prediction of both inputs, an "importance" measure was calculated for each variable of the best final model (i.e., random forest trained with the full training dataset for N_2O prediction and SVM trained with the full dataset for N leaching prediction). For random forests, the importance is quite common: for a given input variable, the values of out-of-sample observations are randomly permuted; the mean squared error is then calculated based on all out-of-sample sets for all trees in the forest. The increase in the mean squared error compared to the out-of-sample mean squared error calculated with the true values of the predictor is called the importance of the predictor (see (Genuer et al., 2010) for a complete study of this quantity in the framework of variable selection problems). Unfortunately, MLPs and SVMs are not based on bootstrapping so out-of-sample observations do not exist for these methods. Hence, importance cannot be defined or directly compared to the one given for random forests. Nervelessness, a close definition can be introduced by using the validation set selected for the tuning process and by comparing the mean squared error of permuted inputs to the true squared error on this validation set.

Figure 4 illustrates the values of the importance measure in both cases. It can be seen that the two metamodels are very different: that (RF) which aims at estimating N_2O fluxes (left) is mainly based on two important variables (`SOC` and `pH`) whereas SVM, used to estimate N leaching, has a less strict behavior: at least four variables are important in that last modeling, `N_MR`, `N_FR`, `pH` and `Nres`.

[Figure 4 about here.]

N_2O fluxes are mainly related to denitrification processes, which require anaerobic conditions and organic material as substrate (Firestone et al., 1979). Anaerobic conditions form if diffusion of oxygen is blocked in wet soils, or in denitrification "hotspots" around organic matter promoting very high oxygen consumption rates (Parkin, 1987). It is therefore not surprising

that the soil organic carbon content (SOC) was found to be the most important for the prediction of N_2O fluxes. Soil pH is also an important parameter, influencing both the reduction of nitrate (total denitrification) but also the reduction of N_2O to N_2 (Granli and Bøckman, 1994). For nitrogen leaching, on the other hand, we found that the most important factor was the most important factor was nitrogen input as manure amendment, mineral fertilizer spreading, and N from crop residue incorporation in the soil before sowing (these are indeed even more important than pH). To a large degree, nitrogen leaching is determined by soil texture which controls the percolation rate of water through the soil profile and precipitation. As a consequence, it is not surprising to find the top-factors determining nitrogen leaching in a relatively narrow range, as compared to N_2O fluxes.

4.5. Conclusion about the comparison of metamodeling strategies

The experiments described in the following subsections enlighten several facts: first, metamodeling strategies were able to approximate accurately N_2O and N leaching predictions at a low computational cost. Even with small dataset sizes (100 HSMUs to train the data), the overall accuracy rate, measured by R^2, is greater than 80% for at least one metamodel. In this case study, N_2O was harder to predict than N leaching. Then, increasing the size of the training dataset is time consuming but also leads to a better accuracy in the prediction for (almost) all the methods. Hence, the selection of a metamodeling approach has to be based on a careful compromise between computational costs and accuracy. This choice strictly depends on the size of available training data and on the project's target. We pointed out that splines and kriging based methods should be chosen when the number of training data is smaller than 2 000 since they provided the most accurate solution with a reasonable running time. With large datasets, random forests were able to handle the training step and to calculate accurate predictions with low computational costs (more than 15 000 observations were trained in about 15 minutes and only several seconds were needed in predicting 19 000 new values).

Finally, we pointed out, in Section 4.4.3, that combining metamodeling with an importance measure can also be used to provide a simplified insight on the important processes and on the main input variables involved in the prediction of N_2O fluxes and N leaching. This can help to find strategies to control nitrogen surplus or to perform a fast sensitivity analysis. This last issue is currently under investigation.

5. Conclusion

This article provides a full comparison of several metamodel approaches for the prediction of N_2O fluxes and N leaching from European farmlands. The conclusions of the meta-model comparison are general enough to be extended to other similar case studies. A more valuable and detailed impact assessment of CC standards at European or country level is possible only by simulating all the 207000 HSMUs that cover the EU27. This approach demands the collection of enormous amounts of data and their storage into large datasets. From our work, random forest proved to be a reliable and effective tool for elaborating large datasets with low computational costs and an acceptable accuracy. For these reasons it has been chosen to be integrated into the CCAT platform to estimate the N_2O fluxes and N leaching from the EU27 farmlands.

References

Akaike, H., 1974. A new look at the statistical model identification. IEEE Transactions on Automatic Control 19, 716–723.

Alliance Environnement, 2007. Evaluation of the application of cross compliance as foreseen under regulation 1782/2003. Part I: descriptive report. Technical Report. D.G. Agriculture. Institute for European Environmental Policy (IEEP), London, UK and Orade-Brche Sarl, Auzeville, France.

Amit, Y., Geman, D., 1997. Shape quantization and recognition with random trees. Neural Computation 9, 1545–1588.

Anderson, M., Kustas, W., Norman, J., 2003. Upscaling and downscaling - a regional view of the soil-plant-atmosphere continuum. Agronomy Journal 95, 1408–1423.

Askegaard, M., Olesen, J., Kristensen, K., 2005. Nitrate leaching from organic arable crop rotations: effects of location, manure and catch crop. Soil Use and Management 21, 181–188.

Berlinet, A., Thomas-Agnan, C., 2004. Reproducing Kernel Hilbert Spaces in Probability and Statistics. Kluwer Academic Publisher, Boston, Norwell, MA, USA / Dordrecht, The Netherlands.

Bishop, C., 1995. Neural Networks for Pattern Recognition. Oxford University Press, New York, USA.

Blanning, R., 1975. The construction and implementation of metamodels. Simulation 24, 177–184.

Boser, B., Guyon, I., Vapnik, V., 1992. A training algorithm for optimal margin classifiers, in: 5^{th} annual ACM Workshop on COLT, D. Haussler Editor, ACM Press. pp. 144–152.

Bouzaher, A., Lakshminarayan, P., Cabe, R., Carriquiry, A., Gassman, P., Shogren, J., 1993. Metamodels and nonpoint pollution policy in agriculture. Water Resources Research 29, 1579–1587.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

Breiman, L., Friedman, J., Olsen, R., Stone, C., 1984. Classification and Regression Trees. Chapman and Hall, New York, USA.

Britz, W., 2008. Automated model linkages: the example of CAPRI. Agrarwitschaft 57, 363–367.

Britz, W., Leip, A., 2009. Development of marginal emission factors for N losses from agricultural soils with the DNDC-CAPRI metamodel. Agriculture, Ecosystems and Environment 133, 267–279.

Britz, W., Witzke, P., 2008. CAPRI model documentation 2008. CAPRI project, Institute for Food and Resource Economics. Bonn, Germany. Online at: `http://www.capri-model.org`.

Chadwick, D., 2005. Emissions of ammonia, nitrous oxide and methane from cattle manure heaps: effect of compaction and covering. Atmospheric Environement 39, 787–799.

Chang, C., Lin, C., 2001. LIBSVM: a library for support vector machines. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Christmann, A., Steinwart, I., 2007. Consistency and robustness of kernel-based regression in convex risk minimization. Bernouilli 13, 799–819.

Cressie, N., 1990. The origins of kriging. Mathematical Geology 22, 239–352.

De Vries, W., Kros, H., Velthof, G., Oudendag, D., Leip, A., van der Velde, M., Kempen, M., 2008. Development and application methodology of environmental impact tool to evaluate cross compliance measures. Technical Report Deliverable 4.2.1 of EU STREP 44423. CCAT Project.

Del Grosso, S., Parton, W., Mosier, A., Walsh, M., D.S., O., Thornton, P., 2006. DAYCENT national-scale simulation of nitrous oxide emissions from cropped soils in the United States. Journal of Environment Quality 35, 1451–1460.

Dimopoulus, D., Fermantziz, I., Vlahos, G., 2007. The responsiveness of cross compliance standards to environmental pressure. Technical Report Deliverable 12 of the CC Network Project, SSPE-CT-2005-022727. Agricultural University of Athens. Athens, Greece.

Elbersen, B., Jongeneel, R., Kempen, M., Klein-Lankhorst, R., De Vries, W., Lesschen, J., Onate, J., Alonso, M., Kasperczyk, N., Schramek, J., Mik, M., Peepson, A., Bouma, F., Staritsky, I., Kros, H., 2010. Final report of CCAT project results. Technical Report Deliverable 2.8 of EU STREP 44423-CCAT. CCAT project.

European Commission, 2002. Implementation of Council Directive 91/676/EEC concerning the protection of waters against pollution caused by nitrates from agricultural sources. Technical Report. D.G. Environment, European Commission. Luxembourg.

European Council, 1991. Nitrate Directive 91/676/EEC. Technical Report. European Union. Brussels, Belgium.

European Council, 2000. Water Framework Directive 2000/60/EC. Technical Report. European Union. Brussels, Belgium.

European Council, 2003. Council regulation (EC). Technical Report 1782/2003. European Union. Brussels, Belgium.

European Council, 2009. Council regulation (EC) No 73/2009. Technical Report. European Council. Brussels, Belgium.

European Environment Agency, 1995. Europe's environment, the dobriss assessment. Technical Report. European Environment Agency.

914 European Environment Agency, 2010. Annual European community green-
915 house inventory 1980-2008 and inventory report 2010. Submission to the
916 UNFCCC secretariat. Technical Report. European Environment Agency.
917 Copenhagen, Denmark.

918 European Union Commission, 2004. Commission Regulation No 796/2004.
919 Technical Report. European Commission. Brussels, Belgium.

920 European Union Commission, 2009. Commission Regulation No 1122/2009.
921 Technical Report. European Commission. Brussels, Belgium.

922 European Union Commission, 2010. Agriculture in the EU - Statistical and
923 Economic Information - Report 2009. Technical Report. European Com-
924 mission. Brussels, Belgium.

925 FAO, 2005. Key to drought-resistant soil and sustained food production.
926 The importance of soil organic matter. Technical Report Soils bulletin 80.
927 Food Agricultural Organization. Rome, Italy.

928 FAO, 2007. Payer les agriculteurs pour les services environnmentaux. La
929 situation mondiale de l'alimentation et de l'agriculture. Technical Report.
930 Food Agricultural Organization. Rome, Italy.

931 Firestone, M., Smith, M., Firestone, R., Tiedje, J., 1979. The influence of
932 nitrate, nitrite, and oxygen on the composition of the gaseous products of
933 dentrification in soil. Soil Science Society of America Journal 43, 1140–
934 1144.

935 Follador, M., Leip, A., 2009. Derivation of DNDC meta-models to evaluate
936 the impact of cross compliance measures on nitrogen N surplus, N leaching,
937 N2O emissions at European scale. Technical Report Deliverable 4.2.3 of EU
938 STREP 44423-CCA. European Commission, D.G Joint Research Centre,
939 Institute for Environment and Sustainability, Climate Change Unit. Ispra,
940 Italy.

941 Follador, M., Leip, A., Orlandini, L., 2011. Assessing the impact of cross
942 compliance measures on nitrogen fluxes from european farmlands with
943 DNDC-EUROPE. Environmental Pollution In Press.

944 Forrester, A., Keane, A., 2009. Recent advances in surrogate-based optimiza-
945 tion. Progress in Aerospace Sciences 45, 50–79.

946 Friedman, J., 1991. Multivariate adaptive regression splines. Annals of Statis-
947 tics 19, 1–67.

948 Genuer, R., Poggi, J., Tuleau-Malot, C., 2010. Variable selection using ran-
949 dom forests. Pattern Recognition Letters 31, 2225–2236.

950 van Gighc, J., 1991. System design modeling and metamodeling. Springer,
951 New York, USA.

952 Granli, T., Bøckman, O., 1994. Nitrous oxide from agriculture. Norwegian
953 Journal of Agricultural Sciences Supplement No. 12. Norsk Hydro Research
954 Centre: Porsgrunn, Norway.

955 van Grinsven, H., Ward, M., Benjamin, N., De Kok, T., 2006. Does the
956 evidence about health risks associated with nitrate ingestion warrant an
957 increase of the nitrate standard for drinking water? Environmental Health:
958 A Global Access Science Source 5.

959 Gu, C., 2002. Smoothing Spline ANOVA Models. Springer-Verlag, New
960 York, USA.

961 Haberlandt, U., Krysanova, V., Bardossy, A., 2002. Assessment of nitrogen
962 leaching from arable land in large river basins. Part II: regionalisation using
963 fuzzy rule based modelling. Ecological Modelling 150, 277–294.

964 Ho, T., 1998. The random subspace method for constructing decision forests.
965 IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 832–
966 844.

967 Hornik, K., 1991. Approximation capabilities of multilayer feedfoward net-
968 works. Neural Networks 4, 251–257.

969 Intergovernmental Panel on Climate Change, 2007. Fourth Assessment Re-
970 port of the Intergovernmental Panel on Climate Change. Working Group
971 I - The Physical Science Basis. Cambridge University Press, Cambridge,
972 United Kingdom / New York, NY, USA,.

973 Jongeneel, R., Elbersen, B., Klein-Lankhorst, R., De Vries, W., Kros, H.,
974 Velthof, G., Kempen, M., Annen, D., Onate, J., van der Velde, M., Leip,
975 A., 2008. Report describing the operationalisation of the first selection of
976 indicators into impacts of Cross Compliance for the implementation in the

977 first prototype of the analytical tool. Technical Report Deliverable 2.3 of
978 EU STREP 44423-CCAT.

979 Jongeneel, R., Elbersen, B., de Vries, V., Klein-lankhorst, J., Schramek, J.,
980 Rudloff, B., Heckelei, T., Kempen, M., Annen, D., van der Velde, M., Leip,
981 A., Redman, M., Mikk, M., Onate, J., Slangen, L., 2007. General approach
982 to the assessment of the impacts of CC in the EU and list of indicators.
983 Technical Report Deliverables 2.1 and 2.2 of EU STREP 44423. CCAT
984 Project.

985 Kalman, R., 1960. A new approach to linear filtering and prediction prob-
986 lems. Transactions of the ASME, Journal of Basic Engineering 82D, 35–45.

987 Keerthi, S., Shevade, S., Bhattacharyya, C., Murty, K., 2001. Improvements
988 to platt's SMO algorithm for SVM classifier design. Neural Computation
989 13, 637–649.

990 Kennedy, M., O'Hagan, A., 2001. Bayesian calibration of computer models
991 (with discussion). Journal of the Royal Statistical Society, series B 63,
992 425–464.

993 Kirchmann, H.and Esala, M., Morken, J., Ferm, M., Bussink, W., Gus-
994 tavsson, J., Jakobsson, C., 1998. Ammonia emissions from agriculture.
995 Nutrient Cycling in Agroecosystems 51, 1–3.

996 Kleijnen, J., 1975. A comment on Blanning's "Metamodel for sensitivity
997 analysis: the regression metamodel in simulation". Interfaces 5, 21–23.

998 Kleijnen, J., 2009. Kriging metamodelin in simulation: a review. European
999 Journal of Operational Research 1992, 707–716.

1000 Kleijnen, J., Sargent, R., 2000. A methodology for fitting and validating
1001 metamodels in simulation. European Journal of Operational Research 120,
1002 14–29.

1003 Krogh, A., Hertz, J., 1992. A simple weight decay can improve generalization,
1004 in: Advances in Neural Information Processing Systems, Kaufmann, M..
1005 pp. 950–957.

1006 Krysanova, V., Haberlandt, U., 2002. Assessment of nitrogen leaching from
1007 arable land in large river basins. Part I: simulation experiments using a
1008 process-based model. Ecological Modelling 150, 255–275.

Leip, A., Achermann, B.and Billen, G., Bleeker, A., Bouwman, L., de Vries, W., Dragosits, U., Döring, U., Fernall, D., Geupel, M., Johnes, P., Le Gall, A.C., Monni, S., Nevečeřal, R., Orlandini, L., Prud'homme, M., Reuter, H., Simpson, D., Seufert, G., Spranger, T., Sutton, M., van Aardenne, J., Voss, M., Winiwarter, W., 2011a. Integrating nitrogen fluxes at the European scale, in: Sutton, M., Howard, C., Erisman, J., Billen, G., Bleeker, A., van Grinsven, H., Grennfelt, P., Grizzetti, B. (Eds.), European Nitrogen Assessment. Cambridge University Press, New York, USA. chapter 16, pp. 345–376.

Leip, A., Britz, W., De Vries, W., Weiss, F., 2011b. Farm, land, and soil nitrogen budgets for agriculture in Europe. Environmental Pollution In Press.

Leip, A., Busto, M., Winiwarter, W., 2011c. Developing stratified N2O emission factors for Europe. Environmental Pollution In Press.

Leip, A., Dämmgen, U., Kuikman, P., van Amstel, A., 2005. The quality of European (EU-15) greenhouse gas inventories from agriculture. Environmental Sciences 2, 177–192.

Leip, A., Marchi, G., Koeble, R., Kempen, M., Britz, W., Li, C., 2008. Linking an economic model for european agriculture with a mechanistic model to estimate nitrogen and carbon losses from arable soils in europe. Biogeosciences 5, 73–94.

Li, C., 2000. Modeling trace gas emissions from agricultural ecosystems. Nutrient Cycling in Agroecosystems 58, 259–273.

Li, C., Frolking, S., Frolking, T., 1992. Model of nitrous oxide evolution from soil driven by rainfall events: 1, model structure and sensitivity. Journal of Geophysical Research 97, 9759–9776.

Lin, Y., Zhang, H., 2006. Component selection and smoothing in smoothing spline analysis of variance models. Annals of Statistics 34, 2272–2297.

Lophaven, S., Nielsen, H., Sondergaard, J., 2002. DACE - A Matlab kriging toolbox, Version 2.0. Technical Report IMM-TR-2002-12. Informatics and Mathematical Modelling, Technical University of Denmark. DK-2800 Kgs. Lyngby, Denmark.

Matson, P., Parton, W., Power, A., Swift, M., 1997. Agricultural intensification and ecosystem properties. Science 277, 504–509.

Mattera, D., Haykin, S., 1998. Support vector machines for dynamic reconstruction of a chaotic system, in: Schölkopf, B., Burges, C., Smola, A. (Eds.), Advances in Kernel Methods: Support Vector Learning. MIT Press, Cambridge, MA, USA, pp. 209–241.

Meckesheimer, M., Booker, A., Barton, R., Simpson, T., 2002. Computationally inexpensive metamodel assessment strategies. AIAA Journal 40, 2053–2060.

Montès, P., 1994. Smoothing noisy data by kriging with nugget effects, in: Laurent, P., Le Méhauté, A., Shumaker, L., Peters, A. (Eds.), Wavelets, Images and Surface Fitting. Wellesley, MA, USA. chapter AK Peters, pp. 371–378.

Ng, C., Young, P., 1990. Recursive estimation and forecasting of non-stationary time series. Journal of Forecasting 9, 173–204.

Oenema, O., Oudendag, D., Velthof, G., 2007. Nutrient losses from manure management in the European Union. Livestock Science 112, 261–272.

Oenema, O., Witzke, H., Klimont, Z., Lesschen, J., Velthof, G., 2009. Integrated assessment of promising measures to decrease nitrogen losses from agriculture in EU-27. Agriculture, Ecosystems and Environment 133, 280–288.

Parkin, T., 1987. Soil microsites as a source of denitrification variability. Soil Science Society of America Journal 51, 1194–1199.

Pineros Garcet, J., Ordonez, A., Roosen, J., Vanclooster, M., 2006. Metamodeling: theory, concepts and application to nitrate leaching modelling. Ecological Modelling 193, 629–644.

Power, A., 2010. Ecosystem services and agriculture: tradeoffs and synergies. Philosofical Transactions of the Royal Society B 365, 2959–2971.

Ratto, M., Pagano, A., 2010. Recursive algorithms for efficient identification of smoothing spline anova models. Advances in Statistical Analysis 94, 367–388.

Ratto, M., Pagano, A., Young, P.C., 2007. State dependent parameter metamodelling and sensitivity analysis. Computer Physics Communications 177, 863–876.

Ripley, B., 1994. Neural networks and related methods for classification. Journal of the Royal Statistical Society, Series B 56, 409–456.

Sacks, J., Welch, W., Mitchell, T., Wynn, H., 1989. Design and analysis of computer experiments. Statistical Science 4, 409–435.

Santner, T., Williams, B., Notz, W., 2003. The Design and Analysis of Computer Experiments. Springer, New York, NY, USA.

Scherr, S., Sthapit, S., 2009. Farming and land use to cool the planet, in: Institute, T.W. (Ed.), State of World 2009. W.W. Norton & Co., Washington DC, USA. chapter 3, pp. 30–49.

Schwarz, G., 1978. Estimating the dimension of a model. Annals of Statistics 6, 461–464.

Schweppe, F., 1965. Evaluation of likelihood functions for Gaussian signals. IEEE Transactions on Information Theory 11, 61–70.

Shaffer, M., Ma, L., 2001. Carbon and nitrogen dynamics in upland soils, in: Shaffer, M., Ma, L., Hansen, S. (Eds.), Modeling Carbon and Nitrogen Dynamics for Soil Management. CRC Press LLC, Bacon Raton, Florida, USA. chapter 2, pp. 11–27.

Simpson, T., Peplinski, J., Koch, P., Allen, J., 2001. Metamodels for computer-based engineering design: survey and recommendations. Engineering with Computers 17, 129–150.

Singh, R., 2000. Environmental consequences of agricultural development: a case study from the green revolution state of Haryana, India. Agriculture, Ecosystems and Environment 82, 97–103.

Steinwart, I., Christman, A., 2008. Support Vector Machines. Information Science and Statistics.

Steinwart, I., Christmann, A., 2008. Support Vector Machines. Information Science and Statistics, Springer.

Storlie, C., Bondell, H., Reich, B., Zhang, H., 2011. Surface estimation, variable selection, and the nonparametric oracle property. Statistica Sinica 21, 679–705.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, series B 58, 267–288.

Tilman, D., Cassman, K., Matson, P., Naylor, R., Polasky, S., 2002. Agricultural sustainability and intensive production practices. Nature 418, 671–677.

Vanclooster, M., Viaene, P., Christiaens, K., Ducheyne, S., 1996. Wave: a mathematical model for simulating water and agrochemicals in the soil and vadose environment, reference and user's manual (release 2.1). Technical Report. Institute for Land and Water Management, Katholieke Universiteit Leuven. Leuven, Belgium.

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer Verlag, New York, USA.

Vapnik, V., 1998. Statistical Learning Theory. Wiley, New York, USA.

van der Velde, M., Bouraoui, F., de Vries, W., 2009. Derivation of EPIC meta-models to evaluate the impact of cross compliance measures on leaching and runoff of nitrogen and soil erosion at European scale. CCAT Deliverable (report) 4.2.3.2.. CCAT project.

Velthof, G., Oudendag, D., Witzke, H., Asman, W., Klimont, Z., Oenema, O., 2009. Integrated assessment of nitrogen emissions from agriculture in EU-27 using MITERRA-EUROPE. Journal of Environmental Quality 38, 1–16.

Venables, W., Ripley, B., 2002. Modern Applied Statistics with S-PLUS. Springer, New York, USA.

Wahba, G., 1990. Spline Models for Observational Data. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, USA.

Wang, G., Shan, S., 2007. Review of metamodeling techniques in support of engineering design optimization. Journal of Mechanical Design 129, 370–380.

1133 Webb, J., Menzi, H., Pain, B., Misselbrook, T., Dämmgen, U., Hendriks, H.,
1134 Döhler, H., 2005. Managing ammonia emissions from livestock production
1135 in Europe. Environmental Pollution 135, 399–406.

1136 Werbos, P., 1974. Beyond regression: New tools for prediction and analysis
1137 in behavior sciences. Ph.D. thesis. Commitee on Applied Mathematics,
1138 Harvard University. Cambridge, MA, USA.

1139 Young, P., 1999. Nonstationary time series analysis and forecasting. Progress
1140 in Environmental Science 1, 3–48.

1141 Young, P., 2000. Stochastic, dynamic modelling and signal processing: time
1142 variable and state dependent parameter estimation, in: Fitzgerald, W.,
1143 Walden, A., Smith, R., Young, P. (Eds.), Nonlinear and Nonstationary
1144 Signal Processing. Cambridge University Press, Cambridge, USA, pp. 74–
1145 114.

1146 Young, P., 2001. The identification and estimation of nonlinear stochastic sys-
1147 tems, in: Mees, A. (Ed.), Nonlinear Dynamics and Statistics. Birkhauser,
1148 USA, Boston, USA, pp. 127–166.

1149 Young, P., Ng, C., 1989. Variance intervention. Journal of Forecasting 8,
1150 399–416.

1151 Zhang, Y., Li, C., Zhou, X., Moore, B., 2002. A simulation model linking crop
1152 growth and soil biogeochemistry for sustainable agriculture. Ecological
1153 Modelling 151, 75–108.

List of Figures

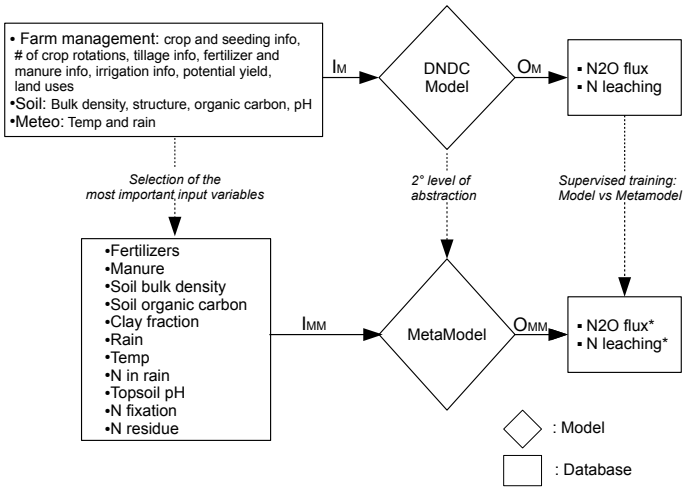

Figure 1: Flow of data through the DNDC-EUROPE model (M) and relationship with the metamodel's one (MM). The input variables of the metamodels were selected from the original DNDC-EUROPE dataset (screening). The estimated (*) output were compared with the detailed model's output during the training and test phases to improve the metamodel and to evaluate the goodness of the approximation.

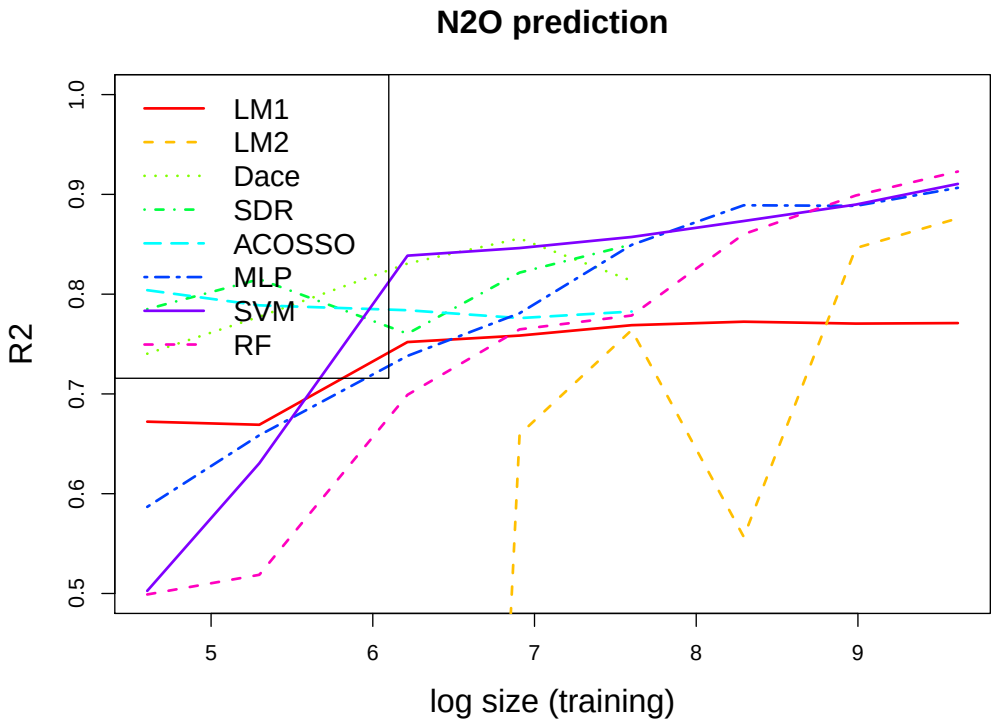

Figure 2: R^2 evolution in function of the size of the train set (log scale) for N_2O prediction

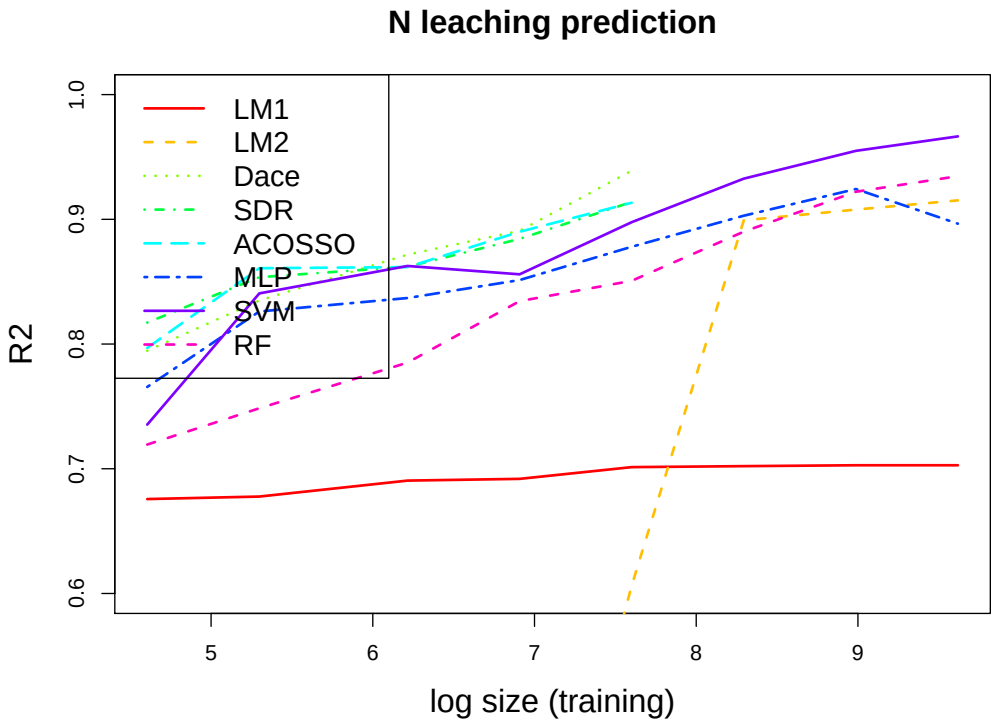

Figure 3: R^2 evolution in function of the size of the train set (log scale) for N leaching prediction

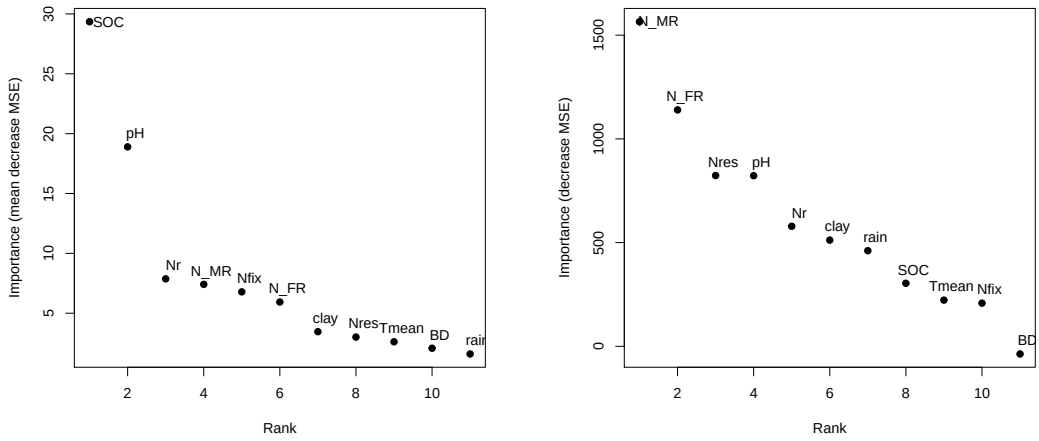

Figure 4: Importance measure for each variable in the case of (left) N_2O prediction with the full training dataset and random forest and of (right) N leaching prediction with the full training dataset and SVM (For the meaning of the acronyms, please refer to Section 4.2)

List of Tables

Size of the dataset	LM1	LM2	Dace	SDR	Acosso	MLP	SVM	RF
100	67.22% 11.50	✓ ✓	74.03% 9.11	78.50% 7.54	**80.40**% 6.88	58.68% 14.50	50.26% 17.45	49.90% 17.57
200	66.91% 11.61	-13 093% 4 626	77.74% 7.81	**81.50**% 6.49	78.88% 7.41	65.86% 11.98	63.05% 12.96	51.87% 16.89
500	75.20% 8.70	-163% 92.35	83.07% 5.94	76.04% 8.41	78.39% 7.58	73.81% 9.19	**83.86**% 5.66	69.91% 10.56
1 000	76.85% 8.47	65.94% 11.95	**85.58**% 5.06	82.16% 6.26	77.60% 7.86	78.81% 7.69	84.62% 5.40	76.47% 8.25
2 000	76.89% 8.11	76.40% 8.28	81.34% 6.55	84.16% 5.27	78.26% 7.63	84.94% 5.28	**85.73**% 5.01	77.86% 7.77
4 000	77.24% 7.99	55.67% 15.55	✓ ✓	✓ ✓	✓ ✓	**88.91**% 3.89	87.33% 4.45	86.01% 4.90
8 000	77.05% 8.05	84.62% 5.40	✓ ✓	✓ ✓	✓ ✓	88.85% 3.91	88.98% 3.86	**89.89**% 3.55
$\simeq$ 15 000	77.10% 8.03	87.60% 3.28	✓ ✓	✓ ✓	✓ ✓	90.66% 3.28	91.05% 3.14	**92.29**% 2.71

Table 2: R^2 (first line) and MSE (second line) on the test set for each method and various sizes of the training dataset for N_2O prediction. For each size, the best R^2 is in bold. ✓ corresponds to cases impossible to train, either because the model is over-specified (more parameters to estimate than the number of observations: LM2) or because the training size is too large for the method to be used (Dace/SDR/Acosso)

Size of the dataset	LM1	LM2	Dace	SDR	Acosso	MLP	SVM	RF
100	67.57%	✓	79.46%	**81.72**%	79.69%	76.56%	73.54%	71.94%
	1 742	✓	1 103	982	1 091	1 259	1 421	1 507
200	67.77%	-2 086%	83.49%	85.36%	**86.08**%	82.61%	84.06%	74.85%
	1 731	$> 10^6$	887	786	747	934	856	1 351
500	69.05%	36.92%	87.17%	86.20%	86.17%	83.69%	**86.26**%	78.51%
	1 662	3 388	689	741	743	876	738	1 154
1 000	69.19%	27.24%	**89.08**%	88.43%	89.00%	85.13%	85.59%	83.44%
	1 655	3 908	587	621	591	799	774	889
2 000	70.13%	60.62%	**93.90**%	91.39%	91.33%	84.94%	89.77%	85.07%
	1 604	2 115	328	462	466	655	549	802
4 000	70.21%	89.92%	✓	✓	✓	**93.26**%	87.33%	89.01%
	1 600	541	✓	✓	✓	521	362	590
8 000	70.28%	90.78%	✓	✓	✓	92.43%	**95.49**%	92.21%
	1 596	495	✓	✓	✓	406	242	418
$\simeq$ 15 000	70.28%	91.52%	✓	✓	✓	89.65%	**96.65**%	93.46%
	1 596	455	✓	✓	✓	556	180	351

Table 3: R^2 (first line) and MSE (second line) on the test set for each method and various sizes of the training dataset for N leaching prediction. For each size, the best R^2 is in bold. ✓ corresponds to cases impossible to train, either because the model is over-specified (more parameters to estimate than the number of observations: LM2) or because the training size is too large for the method to be used (Dace/SDR/Acosso)

Size of the dataset	LM1	LM2	Dace	SDR	Acosso	MLP	SVM	RF
100	80.4	✓	72.7	52.4	**50.2**	125.5	159.6	150.0
	2 400	✓	2 319	1 845	**1 597**	2 911	3 816	3 538
200	84.5	$> 10^5$	68.1	**52.3**	64.6	100.3	113.7	145.4
	2 461	$> 10^6$	2 207	**1 915**	2 098	2 534	2 636	3 352
500	59.3	1 472.9	49.6	74.0	60.2	84.9	**42.5**	99.1
	2 027	48 769	1 928	2 589	2 303	2 172	**1 753**	2718
1 000	56.9	203.5	48.6	51.0	63.4	53.9	**48.5**	77.7
	1 980	8 384	1 643	**1 633**	2 065	1 888	1 874	2 348
2 000	50.3	81.5	66.7	37.8	62.9	**38.4**	41.6	70.4
	1 826	2 890	2 456	1 212	3 000	**1 039**	1 663	2 421
4 000	46.1	539.2	✓	✓	✓	**33.0**	37.6	52.8
	1 711	32 290	✓	✓	✓	**1 110**	1 519	2 040
8 000	42.2	60.9	✓	✓	✓	**31.0**	43.2	38.3
	1 564	2 846	✓	✓	✓	**1 072**	1 773	1 645
$\simeq$ 15 000	42.2	29.0	✓	✓	✓	29.0	35.7	**25.6**
	1 568	1 339	✓	✓	✓	1 339	1 833	**807**

Table 4: Standard deviation (first line) and maximum (second line) of the squared errors on the test set for each method and various sizes of the training dataset for N_2O prediction. For each size, the minimal standard deviation and the minimal value of the maxima are in bold. ✓ corresponds to cases impossible to train, either because the model is over-specified (more parameters to estimate than the number of observations: LM2) or because the training size is too large for the method to be used (Dace/SDR/Acosso)

Size of the dataset	LM1	LM2	Dace	SDR	Acosso	MLP	SVM	RF
100	6.11	✓	**5.83**	6.45	8.14	6.72	9.79	7.99
	173.1	✓	**180.5**	177.0	241.1	238.5	367.7	275.7
200	6.26	$> 10^4$	**5.24**	7.15	8.61	6.28	5.36	7.95
	184.7	$> 10^5$	152.4	290.6	279.5	213.8	**146.8**	284.6
500	6.99	45.7	7.34	7.38	8.62	6.89	**6.77**	7.83
	204.1	1 427.7	238.2	**280.0**	280.9	213.8	302.8	290.7
1 000	7.37	82.3	7.64	**7.10**	8.90	7.72	10.24	7.47
	220.9	4 090.4	270.6	**239.8**	255.3	289.0	358.1	291.1
2 000	5.91	71.9	**2.66**	3.15	9.13	5.74	6.63	5.53
	177.4	4 309.1	**96.6**	113.3	128.7	225.9	320.6	212.7
4 000	5.71	4.94	✓	✓	✓	**3.50**	3.61	4.51
	167.0	213.5	✓	✓	✓	**134.5**	123.1	218.2
8 000	5.59	4.31	✓	✓	✓	2.80	**2.38**	2.60
	162.0	161.8	✓	✓	✓	77.8	77.4	**70.4**
$\simeq$ 15 000	5.53	2.54	✓	✓	✓	4.74	**1.35**	3.00
	157.2	72.1	✓	✓	✓	147.0	**36.1**	128.7

Table 5: Standard deviation (first line $\times 10^3$) and maximum (second line $\times 10^3$) of the squared errors on the test set for each method and various sizes of the training dataset for N leaching prediction. For each size, the minimal standard deviation and the minimal value of the maxima are in bold. ✓ corresponds to cases impossible to train, either because the model is over-specified (more parameters to estimate than the number of observations: LM2) or because the training size is too large for the method to be used (Dace/SDR/Acosso)

Training dataset size	Number of selected variables (N_2O prediction)	Number of selected variables (N leaching prediction)
200	79	95
500	74	84
1 000	75	89
2 000	79	94
4 000	94	95
8 000	98	97
$\simeq$ 15 000	96	100

Table 6: Number of variables selected by AIC stepwise procedure in LM2 for N_2O prediction and N leaching prediction in function of the training dataset size

Training dataset size	Number of selected variables (N_2O prediction)		Number of selected variables (N leaching prediction)	
	ACOSSO	SDR	ACOSSO	SDR
100	30	23	24	39
200	13	17	31	26
500	17	32	18	19
1 000	7	9	28	31
2 000	9	10	29	30

Table 7: Number of ANOVA components selected by the COSSO penalty in ACOSSO and SDR for N_2O prediction and N leaching prediction as a function of the training dataset size.

Use	LM1	LM2	Dace	SDR	Acosso	MLP	SVM	RF
Train	<1 s.	50 min	80 min	4 hours	65 min	2.5 hours	5 hours	15 min
Prediction	<1 s.	<1 s.	90 s.	14 min	4 min.	1 s.	20 s.	5 s.

Table 8: Approximative time for training from about 15 000 observations (first line) and for predicting about 19 000 observations (second line) on a desktop computer (Processor 2GHz, 1.5GO RAM). In the case of SDR, ACOSSO and DACE we report the time for training using samples with 2 000 model runs because the method can not be used for largest training datasets.