Data-driven spatio-temporal estimation of soil moisture and temperature based on Lipschitz interpolation

Abstract

This study estimates agricultural soil variables using a non-parametric machine learning technique based on Lipschitz interpolation. This method is adapted for the first time to learn spatio-temporal dynamics, accounting for two-dimensional spatial and one temporal coordinate inputs separately. The estimator is validated on real agricultural data, addressing challenges like measurement noise and quantization. The experimental setup, including an edge layer with measurement devices and a cloud layer for data storage and processing, is detailed. Despite its simplicity, the method presents a compelling alternative to Gaussian processes and neural networks.

Keywords: Spatio-temporal estimation; Lipschitz interpolation; Non-parametric learning; Agriculture soil monitoring; Experimental validation

1. Introduction

1.1. Motivation

The spatio-temporal variability of agricultural soil variables constitutes, possibly, one of the main challenges for any support decision system or, even more, automatic irrigation or fertigation system [1]. The problem becomes more apparent when trying to capture the variability of different properties in the soil whose dynamics, if known, are complex and difficult to identify for a particular soil, climate and crop. This is the case, for instance, of soil moisture or nutrients, for which diffusion is almost negligible [2, 3].

Sensor networks might offer a technological solution to the problem, by deploying a sufficiently large number of devices. However, the problem still persists, as commercial instrumentation is expensive and the farmers cannot usually afford the number of devices required to have sufficient spatio-temporal representativeness. Deploying a large number of low-cost sensors, as reported in [4, 5, 6], does not solve the problem, as they are not completely reliable.

This is where artificial intelligence comes handy, since the sensor network can be endowed with smart algorithms that, on one hand, reduce the amount of devices to be installed and, on the other hand, ensure an adequate performance. Those algorithms, which might run in both edge and cloud layers, pursue to extract the maximum information from a given amount of data.

In the context of dynamical systems, this is indeed the objective of state estimators, to recover internal or hidden variables from the measurements collected from the process.

1.2. Review of the state of the art

Spatial state estimators, which do not consider the temporal variability, have been widely studied. This is, for example, the case of [7], which considers processes described as partial differential equations; regression kriging in [8]; or [9], who uses stochastic machine learning techniques to create spatial fields models. The application of this kind of methods to agriculture can only be done if the variable of interest has very slow dynamics, as in [10].

Spatio-temporal estimation methods extend traditional spatial estimation by considering the temporal dimension. This problem has been approached from two different perspectives: those which rely on a model describing the dynamics; and those which only exploits the collected data.

For the former, different modifications of the Kalman filter have been proposed [11, 12], including *kriged* adaptations of the filter [13]. The application of techniques from geostatistics have also attracted the attention of the agriculture community, as in [14]. The reader might find a complete review on statistics for spatio-temporal data in [15].

The main benefit of the previous methodologies constitutes, in fact, their main disadvantage. Having an accurate model describing the spatio-temporal dynamics of the different variables, although useful, is very difficult, as the models are highly complex [16]. It is even more complicated in the agriculture soil, as the number of parameters that needs to be identified to use the models is overwhelming (see [17]).

A possible solution to this problem might come from the use of data-driven or data-based estimation methods. Techniques such as tension spline and global polynomial interpolation have proven to be effective to analyze groundwater levels, as detailed in [18]. Other approaches, like inverse distance weighting and local polynomial interpolation, have been applied successfully in environmental modeling, discussed in [19].

Recent works have introduced other innovative methods. For instance, [20] examines the use of convolutional long short-term memory neural networks, demonstrating their applicability beyond agriculture and showcasing their versatility in various domains. In [21], a novel framework for spatio-temporal predictions, particularly in environmental contexts, is presented. It uses Empirical Orthogonal Functions (EOFs) decomposition along with deep feedforward neural networks, offering promising results in environmental forecasting and data interpolation. Additionally, [22] explores the application of big data and Kernel Density Estimation techniques, specifically for tracking the COVID-19 pandemic's progression.

Other authors have proposed the use of non-parametric machine learning techniques. Non-parametric methods, such as spatio-temporal kriging [23] or Gaussian processes (GPs) [24], depend explicitly on the database in the prediction stage. In contrast, parametric methods as the ones mentioned before do only consider the trained parameters at this stage.

Non-parametric learning methods offer certain advantages and disadvantages over parametric methods. On the one hand, they do not rely on a fixed structure of the process, so prior knowledge of the underlying model of the function to be learnt is not required. On the other hand, the predictions to be made depend explicitly on the available data set, so they are typically computationally more expensive than parametric methods in terms of predicting, and a larger amount of data is usually needed. One of their primary benefits is that they can not only predict the desired variable but also estimate how certain that

prediction is. Applications of these methodologies to the estimation of agricultural variables can be found in [25, 26, 27].

In the context of predicting agricultural variables (moisture or temperature) based on spatiotemporal measurements (x, y, t), non-parametric methods may present significant advantages over parametric methods. Firstly, the complex and dynamic nature of soil moisture patterns is challenging to model accurately with fixed structures and parameters inherent in parametric methods. Non-parametric methods, which rely directly on the data, offer greater flexibility and adaptability to capture these intricate dynamics without necessarily making strong assumptions about the underlying distribution. For example, support vector machine approaches, such as those discussed by [28], demonstrate how data-driven models can effectively learn and predict soil moisture patterns by capturing nonlinear relationships in the data. Secondly, for applications that require real-time implementation on edge devices, non-parametric methods are preferable because they do not necessitate extensive re-training processes. Methods like kriging, as explored by the authors in [29], provide an efficient non-parametric approach to spatial interpolation without the need for complex re-training. This characteristic is particularly beneficial in scenarios where computational resources are limited, and the ability to perform continuous updates based on new data without complex re-training is crucial. Thus, non-parametric methods provide a practical and efficient solution for accurate and responsive soil moisture interpolation in agricultural settings.

1.3. Contributions of the manuscript

In this work, the method known as *kinky inference* (KI) [30] is proposed, tailored to the problem of spatio-temporal estimation in agriculture soil. It is based on Lipschitz interpolation, a technique that has received various names in the literature [31, 32], and classified as a non-parametric machine learning method. Unlike Gaussian processes or kriging, which are grounded in stochastic frameworks, Lipschitz interpolation offers a deterministic bound on the uncertainty of predictions, providing a clear and quantifiable measure of prediction reliability.

A notable advantage of Lipschitz interpolation lies in its simplicity and computational efficiency. The method relies on straightforward mathematical operations such as comparisons, additions, and multiplications, avoiding the matrix inversions that are required in GPs. This is especially significant in the context of implementation on dedicated platforms, such as Field Programmable Gate Arrays (FPGAs) [33] and other specialized hardware designed for distributed computing at the edge layer. This capability enables real-time, on-site processing of data, crucial for timely decision-making in agricultural settings.

The main contributions and novelties of this paper are:

- A three-dimensional KI estimator is proposed, considering two spatial and one temporal dimensions. An adaption of the standard KI method is proposed, allowing to account separately for the influence of each variable.

- The method is implemented and validated in a agricultural farm. The crop was irrigated by sprinklers which, in contrast to drippers, highlight the fully spatio-temporal

dynamics of the agricultural variables. Details on the deployed network of intelligent devices are given. The collected data is made publicly available.

- Data physically collected in the field requires a quantization prior to its transmission. The analysis of the impact of the quantization in the estimation is another contribution of this work.

- A deep comparison of the proposed methods to different existing predictors based on GPs, neural network, and linear regression is presented. This comes along with a discussion that highlights the advantages and disadvantages of the proposed methodology.

The manuscript is organised as follows: Section 2 formally states the three-dimensional estimation problem. Section 3 describes the proposed method. Section 4 introduces the case study, with a description of the devices, the wireless sensor network, and the communication system employed to obtain the data. The experimental results appear in Section 5, and discussed in Section 6. Finally, conclusions are drawn in Section 7.

Notation. Let $x_i \in \mathbb{R}^n$ be a set of vectors for $i = 1, \ldots, N$. Operator $\min\{x_i\}$ ($\max\{x_i\}$) is defined componentwise, returning a vector $y \in \mathbb{R}^n$ such that

$$y(j) = \min_{i=1,\ldots,N} \left(\max_{i=1,\ldots,N} \right) x_i(j), \, \forall j = 1, \ldots, n. \tag{1}$$

The absolute value over a vector $v \in \mathbb{R}^n$ is computed componentwise, that is, $|v| = \{y : y(j) = |v(j)|, \forall j = 1, \ldots, n\}$. Given two vectors v, w, the notation $v \leq w$ implies that the inequality holds for every component.

2. Problem statement

Consider a plain field where soil temperature z_T and moisture z_M might take different values for each temporal and spatial location, i.e.:

$$\mathcal{Z}_T := \{z_T(x, y, t) \in \mathcal{T}, (x, y) \in \mathcal{F}, t \in \mathbb{R}^+ : \; z_T(x, y, t) = f_T(x, y, t)\}, \tag{2a}$$

$$\mathcal{Z}_M := \{z_M(x, y, t) \in \mathcal{M}, (x, y) \in \mathcal{F}, t \in \mathbb{R}^+ : \; z_M(x, y, t) = f_M(x, y, t)\}, \tag{2b}$$

where $(x, y, t) \in \{\mathcal{F} \times \mathbb{R}^+\}$ defines the spatial and temporal coordinates, $\mathcal{Z}_M$ and $\mathcal{Z}_T$ represent the soil moisture and temperature maps, and $f_T : \mathbb{R}^3 \to \mathbb{R}$ and $f_M : \mathbb{R}^3 \to \mathbb{R}$ are unknown functions to be learnt. Set $\mathcal{F}$ defines the field surface, and time is assumed to be positive, without any loss of generality. Sets $\mathcal{T}, \mathcal{M}$ refer to the measuring range, defined as $\{z_\sigma : \underline{z_\sigma} \leq z_\sigma \leq \overline{z_\sigma}\}$, where $\underline{z_\sigma}$ and $\overline{z_\sigma}$ are the minimum and maximum values measurable by the deployed sensors, with $\sigma \in \{T, M\}$.

Remark. Richards equation [34], which models the dynamics of variably saturated water flow, together with the heat equation can be used to model the spatio-temporal behaviour

of the agricultural variables considered in this manuscript. In other words, the solutions of those partial differential equations will be the maps $\mathcal{Z}_T, \mathcal{Z}_M$. However, these partial differential equations depend on a set of parameters that are difficult, if not impossible, to know, such as the unsaturated hydraulic conductivity, the dispersion tensor, or the heat diffusion tensor. Moreover, some of those parameters are not constant, but time and/or space-varying. This fact motivates the use of the proposed learning algorithm.

The measured data $\tilde{f}_\sigma$ at (x, y, t), for $\sigma \in \{T, M\}$, is described by:

$$\tilde{f}_\sigma(x, y, t) = f_\sigma(x, y, t) + v_\sigma(x, y, t), \tag{3}$$

where $v_T(x, y, t), v_M(x, y, t)$ represent noises affecting the temperature and moisture measurements, respectively.

Assumption 1. *Measurement noise is assumed to be bounded, this is,* $|v_T(x, y, t)| \leq \bar{v}_T$, $|v_M(x, y, t)| \leq \bar{v}_M$, $\forall (x, y, t) \in \{\mathcal{F} \times \mathbb{R}^+\}$, *and these bounds* $\bar{v}_T, \bar{v}_M$ *are assumed to be known.*

Through Assumption 1, the proposed method will be able to find estimations with guaranteed certificates.

This measured data must be quantized before its transmission, in such a way that the transmitted data $\tilde{z}_\sigma$ can be defined as:

$$\tilde{z}_\sigma(x, y, t) = Q\left(\tilde{f}_\sigma(x, y, t)\right), \;\; \sigma \in \{T, M\}, \tag{4}$$

where the quantizer $Q(\tilde{f}_\sigma) : \mathbb{R} \to \mathbb{R}$ is defined by its sensitivity Δ_σ and the number of quantization steps N_q, such that $\Delta_\sigma \times N_q = \overline{z_\sigma} - \underline{z_\sigma}$. Therefore[1]:

$$Q(\tilde{f}_\sigma) = \begin{cases} \underline{z_\sigma}, & \underline{z_\sigma} \leq \tilde{f}_\sigma \leq \frac{1}{2}\Delta_\sigma \\ \Delta_\sigma \left\lfloor \frac{\tilde{f}_\sigma}{\Delta_\sigma} + \frac{1}{2} \right\rfloor, & \frac{1}{2}\Delta_\sigma < \tilde{f}_\sigma \leq N_q\Delta_\sigma - \frac{1}{2}\Delta_\sigma \\ \overline{z_\sigma}, & N_q\Delta_\sigma - \frac{1}{2}\Delta_\sigma < \tilde{f}_\sigma \leq \overline{z_\sigma} \end{cases}, \tag{5}$$

being $\lfloor \cdot \rfloor$ the floor operation.

Then, the objective problem is stated as follows.

Problem 1. *Assume a plain field where maps* $\mathcal{Z}_T, \mathcal{Z}_M$ *in* (2) *represent temperature and soil moisture at each spatial and temporal coordinate. Corrupted data is collected from the field as in* (3)*, and later quantized for transmission as in* (4)*. Then, provided Assumption 1 is satisfied, the goal is to obtain an estimation of the variables at unseen coordinates* (x, y, t)*, denoted* $\hat{z}_\sigma(x, y, t)$*, and, also, extreme bounds* $\underline{\hat{z}}_\sigma(x, y, t), \overline{\hat{z}}_\sigma(x, y, t)$ *such that*

$$\underline{\hat{z}}_\sigma(x, y, t) \leq z_\sigma(x, y, t) \leq \overline{\hat{z}}_\sigma(x, y, t), \; \sigma \in \{T, M\}, \forall (x, y, t) \in \{\mathcal{F} \times \mathbb{R}^+\}. \tag{6}$$

[1]The number of quantization steps could be chosen differently for each variable. It is presented here with a common value to ease the presentation, and to match the real configuration of the experimental set-up described in Section 4.

Remark. The choice of this particular agricultural variables, soil temperature and moisture, is motivated by the particular sensors deployed in the field. The method, of course, can be of application to any other variable. In any case, there is a second important motivation, as this choice will allow to compare the performance of the learning methods when the dynamics of the variable is mainly modeled as a diffusion process (temperature) and when the dynamics is more complex (moisture).

Remark. The third spatial dimension (depth coordinate) is not considered in (2). Even though it is well known that these variables change in this direction (see, for instance, [35]), the reason behind this assumption is twofold. First, the estimation method that will be proposed in Section 3 is general, and it can be easily extended to consider additional coordinates. However, this will hinder the exposition of the experimental results (too many dimensions to be drawn). Second, in the experiment described in Section 4, the sensors have all been deployed at the same depth. As the proposed methodology learns from the collected data, making predictions of the variables at a different depth would blur the illustration of the process.

3. Methods

3.1. Analysis of the quantization noise

The effect of the quantizer can be seen as an additional noise, with deterministic bounds, added to the measurements. This is proven in the next proposition, where subscripts $\sigma \in \{T, M\}$ have been removed to ease the presentation.

Proposition 1. *Consider the quantizer $Q(\tilde{f})$ described in (5), where $\tilde{f} = f + v$ is a measurement of the real variable f corrupted by noise v, such that $|v| < \bar{v}$. Then, $Q(\tilde{f}) = f + v_q$, with $|v_q| < \frac{1}{2}\Delta + \bar{v}$.*

Proof. Let's consider the three cases of the quantizer. Consider, without loss of generality, that $\underline{z} = 0$, and thus $\Delta \times N_q = \overline{z}$.

a) $0 \leq \tilde{f} \leq \frac{1}{2}\Delta$: Thus, $f + v_q = 0$, for $-v \leq f \leq \frac{1}{2}\Delta - v$. Then $|v_q| \leq \frac{1}{2}\Delta + \bar{v}$.

b) $\frac{1}{2}\Delta < \tilde{f} \leq N_q\Delta - \frac{1}{2}\Delta$: Thus $f + v_q = \Delta \left\lfloor \frac{f+v}{\Delta} + \frac{1}{2} \right\rfloor$, for $\frac{1}{2}\Delta - v < f \leq N_q\Delta - \frac{1}{2}\Delta - v$. For $f = \frac{1}{2}\Delta - v$:

$$v_q = \Delta \left\lfloor \frac{f+v}{\Delta} + \frac{1}{2} \right\rfloor - f = \Delta \lfloor 1 \rfloor - \frac{1}{2}\Delta - v = \frac{1}{2}\Delta - v. \tag{7}$$

For $f = N_q\Delta - \frac{1}{2}\Delta - v$:

$$v_q = \Delta \left\lfloor \frac{f+v}{\Delta} + \frac{1}{2} \right\rfloor - f = \Delta \lfloor N_q \rfloor - N_q\Delta + \frac{1}{2}\Delta + v = \frac{1}{2}\Delta + v. \tag{8}$$

Then, $|v_q| \leq \frac{1}{2}\Delta + \bar{v}$.

c) $N_q\Delta - \frac{1}{2}\Delta < \tilde{f} \leq N_q\Delta$: Thus, $f + v_q = N_q\Delta$, for $N_q\Delta - \frac{1}{2}\Delta - v \leq f \leq N_q\Delta - v$. Then $|v_q| \leq \frac{1}{2}\Delta + \bar{v}$. □

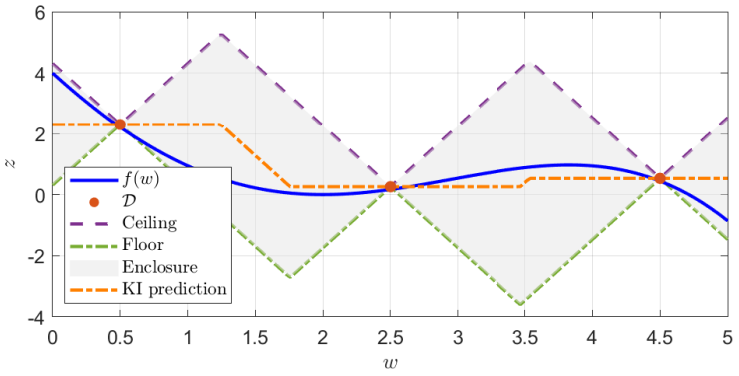


Figure 1: Illustration of the kinky inference prediction over $f(w) = (w-2)^2 \cos\left(\frac{w}{3}\right)$, with $N = 3$ and $L = 4$

3.2. Description of the learning method

In this paper a class of learning rules known as *kinky inference*, which encompasses Lipschitz interpolation techniques, is applied. To do so, it is assumed that the ground truth functions (i.e. f_T and f_M) are Lipschitz continuous.

Definition 1 (Lipschitz continuity). *Given an input space $\mathcal{W} \subset \mathbb{R}^{n_w}$ and an output space $\mathcal{Z} \subset \mathbb{R}^{n_z}$, a function $z = f(w) : \mathcal{W} \to \mathcal{Z}$ is Lipschitz continuous if for every $w_1, w_2 \in \mathcal{W}$,*

$$\|f(w_1) - f(w_2)\| \leq L^* \|w_1 - w_2\|, \tag{9}$$

where $L^* > 0 \in \mathbb{R}$ is called the Lipschitz constant.

The measurements from the process are stored in a data set, denoted $\mathcal{D}$ and defined as:

$$\mathcal{D} := \{(w_i, \tilde{z}_i), i = 1, \ldots, N\}, \tag{10}$$

where the output $\tilde{z}_i \in \mathbb{R}^{n_z}$ denotes a noise-corrupted measurement of $z_i = f(w_i)$, and N stands for the number of points observed. The data set containing inputs $w_i \in \mathbb{R}^{n_w}$ is denoted $\mathcal{W}_\mathcal{D}$.

Given $\mathcal{D}$, and a Lipschitz constant L, Lipschitz continuity of f yields a region, or enclosure, that must contain the unknown function. Then, the standard KI [30] calculates the estimation as the average value in that enclosure, which is given by the minimum ceiling and the maximum floor (see Figure 1), as follows:

Definition 2 (Kinky inference predictor). *Given $\mathcal{D}$ and a Lipschitz constant L, the prediction $\hat{f}$ of the unknown function f at a query point $q \in \mathcal{W}$ is computed as*

$$\hat{f}(q; L, \mathcal{D}) = \frac{1}{2}\Big(\min_{i \in \mathcal{D}} \{\tilde{z}_i + L\|w_i - q\|\} + \max_{i \in \mathcal{D}} \{\tilde{z}_i - L\|w_i - q\|\} \Big). \tag{11}$$

The smallest L that satisfies (9) is called the true Lipschitz constant, and it is not known in general. A *lazily adapted constant* rule was proposed in [36], based on the available observations. Such Lipschitz constant, denoted $L_{\mathcal{D}}$, is obtained as the smallest that is consistent with the observations, up to the bound on the noise level, denoted $\bar{e}$:

$$L_{\mathcal{D}} = \max_{\mathcal{D}} \left\{ \frac{\|\tilde{z}_i - \tilde{z}_{\tilde{i}}\| - 2\bar{e}}{\|w_i - w_{\tilde{i}}\|} : i \neq \tilde{i} \in \{1, \ldots, N\} \wedge \|w_i - w_{\tilde{i}}\| > 0 \right\}. \tag{12}$$

The standard KI is based on a scalar Lipschitz constant $L \in \mathbb{R}$, which cannot consider the effect that each input has on each output separately. An extension to this method was proposed in [37], called the *componentwise Hölder* KI (CHoKI), in which a matricial Lipschitz constant $\mathbf{L} \in \mathbb{R}^{n_z \times n_w}$ was introduced. In this paper, we propose to tailor the componentwise Hölder predictor of [37] to the Lipschitz case.

Definition 3 (Componentwise Lipschitz continuity). *Given an input space $\mathcal{W} \subset \mathbb{R}^{n_w}$ and an output space $\mathcal{Z} \subset \mathbb{R}^{n_z}$, a function $f : \mathcal{W} \to \mathcal{Z}$ is componentwise $\mathbf{L}$-Lipschitz continuous if for every $w_1, w_2 \in \mathcal{W}$ and $\forall k = 1, \ldots, n_z$*

$$|f_k(w_1) - f_k(w_2)| \leq \sum_{j=1}^{n_w} \mathbf{L}_{k,j} |w_{1,j} - w_{2,j}|, \tag{13}$$

where $\mathbf{L} \in \mathbb{R}^{n_z \times n_w}$ is the matricial Lipschitz constant, and f_k is the k-th component of function f.

Based on the previous property, the CHoKY learning method is introduced next.

Definition 4 (CHoKI predictor). *Given $\mathcal{D}$ and a Lipschitz matricial constant $\mathbf{L}$, the prediction $\hat{f}$ of the unknown function f at a query point $q \in \mathcal{W}$ is computed as*

$$\hat{f}(q; \mathbf{L}, \mathcal{D}) = \frac{1}{2} \Big(\min_{i \in \mathcal{D}} \{ (\tilde{z}_i + \mathfrak{d}_{\mathbf{L}}(|w_{i,j} - q_j|) \} + \max_{i \in \mathcal{D}} \{ \tilde{z}_i - \mathfrak{d}_{\mathbf{L}}(|w_{i,j} - q_j|) \} \Big), \tag{14}$$

where

$$\mathfrak{d}_{\mathbf{L}}(w) := \left(v : v_k = \sum_{j=1}^{n_w} \mathbf{L}_{k,j} w_j, \forall k = 1, \ldots, n_z \right). \tag{15}$$

As for the standard KI, a procedure to compute an upper bound of $\mathbf{L}$ based on the available observations, $\mathbf{L}_{\mathcal{D}}$, is proposed. The available learning data set is divided into two sets, which must be disjoint, named $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$. Then, $\mathbf{L}_{\mathcal{D}}$ is computed minimising the prediction error over $\mathcal{D}_{\text{test}}$, predicting with $\mathcal{D}_{\text{train}}$, such that

$$\mathbf{L}_{\mathcal{D}} = \arg\min_{\mathbf{L}} \sum_{w_i \in \mathcal{D}_{\text{test}}} \|\hat{f}(w_i; \mathbf{L}_{\mathcal{D}}, \mathcal{D}_{\text{train}}) - \tilde{z}_i\|^2 \tag{16a}$$

$$\text{s.t.} \quad |\tilde{z}_i - \tilde{z}_j| - 2\bar{e} \leq \mathfrak{d}_{\mathbf{L}}(|w_i - w_j|), \; \forall w_i \neq w_j \in \mathcal{W}_{\mathcal{D}}. \tag{16b}$$

Several nice properties of this predictor were provided in [37]. If the estimated constant (16) is employed in the estimation step (14), the result is guaranteed to be Lipschitz continuous, sample consistent, and with a bounded prediction error, which decreases as more observations are collected.

In addition, there exists various advantages that this method provides in comparison to other non-parametric machine learning techniques. The computational effort to make predictions is smaller than in Gaussian processes, enabling its embedding in dedicated platforms. In particular, the computational burden of a KI prediction grows linearly with the size of the data set [30], this is, $\mathcal{O}(N)$; however, the computational burden of a prediction with Gaussian processes is $\mathcal{O}(N^3)$ [24]. The training stage of Gaussian processes is also a time-demanding process, as it will be shown later in Section 5.

Additionally, the choice of the prior kernel parameters impacts significantly on the outcome. This selection demands expert adjustment which may not always be accessible. Besides, remember that it is assumed that the ground truth function is opaque, that is, nothing is assumed to be known regarding its shape or distributions. All these challenges detract from the simplicity of the methodology proposed in this work, which does not require parameter tuning or design of variables.

3.3. Application of the learning method to the problem

For the problem analysed in this paper, we aim to learn soil temperature f_T and moisture f_M in a plain field. Therefore, there are three dimensions in the input space, i.e. $w = [x, y, t]^T$, and two dimensions in the output space, which are z_T and z_M, given by (2).

Problem 1 can be solved using either the standard KI predictor or the CHoKI predictor. Regarding the CHoKI predictor, firtsly, an estimation of the Lipschitz matrix constants $\mathbf{L}_{\mathcal{D},T}$ and $\mathbf{L}_{\mathcal{D},M}$ must be computed using (16). Given a new query $q = (x, y, t) \in \mathcal{W}$, Problem 1 is addressed computing $\hat{z}_T$ and $\hat{z}_M$ using (14).

By specifying the time instant and coordinates of interest in q, one can get predictions of temperature or soil moisture. These queries may be based on future predictions at the sensor coordinates, spatial interpolations at certain time, or a combination of both: estimations for data gaps in space and time.

Using the CHoKI learning method, the prediction error is bounded in its worst-case. Such bound is denoted $\mu \in \mathbb{R}$, and it is such that:

$$\|f(q) - \hat{f}(q; \mathbf{L}_{\mathcal{D}}, \mathcal{D})\| \leq \mu := \mathfrak{d}_{(\mathbf{L}^* + \mathbf{L}_{\mathcal{D}})}(\mathcal{R}_{\mathcal{D}}) + 2\bar{e}, \tag{17}$$

where $\mathcal{R}_{\mathcal{D}}$ is the maximum possible distance between the input data points in $\mathcal{D}$ and any query in the input space $\mathcal{W}$, this is,

$$\mathcal{R}_{\mathcal{D}} = \max_{q \in \mathcal{W}} \min_{w \in \mathcal{W}_{\mathcal{D}}} |q - w|, \tag{18}$$

and $\bar{e} = \bar{v} + \Delta/2$ stands for the maximum noise, including measurement and quantization noise, according to Proposition 1.

Thus, notice that for queries far from observed data (both in space and time), the prediction error bound increases rapidly. Besides, such bound depends on the unknown

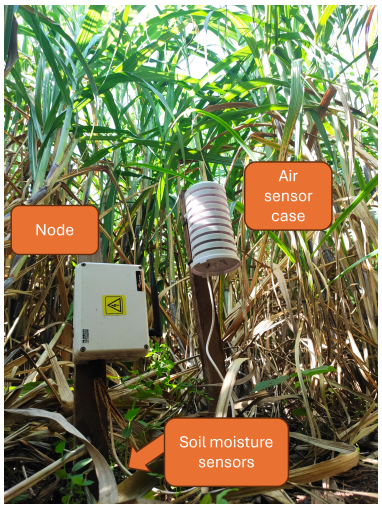


Figure 2: Picture of a node that has multiple sensors for soil moisture and air temperature, in the sugarcane field

constant $\mathbf{L}^*$. Some works have assumed knowledge of such constant (see [38]), and others have proposed practical validation tests to estimate μ, see [39].

4. Materials

4.1. Data measurement network: description and capabilities

A data measurement network comprises a set of interconnected devices, or *nodes*, aiming to collect, process and communicate information from the field.

The nodes used in this paper are shown in Figure 2. They have been designed and built by the authors[2], and they behave as intelligent devices that manage six soil temperature and moisture sensors. These sensors are connected to the core of the node using the I2C protocol. The sensors are powered with voltages of up to 5 V and amperages below milliamperes (mA), allowing easy integration with embedded micro-controllers.

Temperature sensors are quite reliable with the manufacturer's calibration, despite their low price. However, capacitive sensors measure the volumetric water content in a very reduced volume of soil, thus a very local measurement of a highly nonlinear non-diffusive moisture field is obtained. Even if perfectly calibrated, the measurement only represents a very short portion of the soil. This fact will be important later in Sections 5 and 6.

The processor of every node is a Lopy4 board, by Pycom. It includes an ESP32 processor, which is widely used in countless boards distributed by different companies. It is a low cost processor, not exceeding 10€ per unit. Another strong point is its dual-core processor that reaches 298 MHz. The consumption is below milliamperes in a deep sleep mode, and does not usually exceed 100 mA in active mode.

[2]The design and testing of the nodes is a complex task that demands the joint effort of more people, which have been accordingly acknowledged.

The Lopy4 and the expansion board are integrated into a customized board that allows the interconnection of the sensors and the power cut of the sensors in deep sleep mode.

Sensors' data is measured with a sampling time of 20 minutes. The data processing in the nodes consists of averaging the measurements of different sensors, discard of outliers, and a moving-average filter. Temperature is stored in Celsius, while a calibration is performed for soil moisture, such that the measurement stands for volumetric water content. That is, it ranges between 0% and 100%, being 100% a saturated soil and 0% dry soil. The calibration consists in transforming the 1-byte-long measurement from hexadecimal format to decimal, and applying a linear transformation of the form $s_c = ms+n$, being s the original signal and s_c the calibrated one. The coefficients are obtained from experimental tests, being $m = 17/49$ and $n = -85/49$ for temperature and $m = 20/49$ and $n = -100/49$ for moisture. Further details on the characterization of moisture sensors, including repeatability and influence of the temperature, can be found in [40].

Following the manufacturer specifications, the signals may include noises, which are bounded with $\bar{v}_T = 0.80$°C and $\bar{v}_M = 1.65\%$, for the operation ranges.

Both temperature and moisture data are quantized before transmission. The operation ranges are $[0, 85]$°C for the temperature, and $[0, 100]\%$ for soil moisture. One byte is transmitted for each measurement, but 11 positions are reserved, leaving $N_q = 245$ available positions. Then, $\Delta_T = 0.347$°C and $\Delta_M = 0.408\%$. Table 1 summarizes all the features related to the data.

Variable	Units	Range	Meas. Noise	Levels	Quan. Noise	Sample time
Temperature	°C	$[0, 85]$	$\bar{v}_T = 0.80$	245	$\Delta_T = 0.347$	20 min
Moisture	%	$[0, 100]$	$\bar{v}_M = 1.65$	245	$\Delta_M = 0.408$	20 min

Table 1: Main features of the data

4.2. Edge and cloud layer description

Every node sends the measured data from the crop-field (*edge* layer) wirelessly to a central unit, who is in charge of gathering data and forwarding it to an external server (*cloud* layer). This unit is a gateway with LoRaWAN communication protocol [41] capabilities. This device acts as a bridge between the devices that use LoRaWAN and the cloud. The LoRaWAN protocol cuts down communication usage significantly, and it allows wireless communication distances of up to 20 km. Besides, it is a radio-frequency-based protocol, thus avoiding the need for 4G / 5G cards.

The final destination of the outgoing data packages is an SQL database. In particular, a database in Postgre SQL has been deployed. Its main features are its open-source structure with a large community, that it is free and cross-platform, and that it has a shallow learning curve and the ability to handle large volumes of data.

The last part of this data reception scheme comprises an API (web service) linking the users who require information with the database. Figure 3 shows a schematic picture with all the components described.

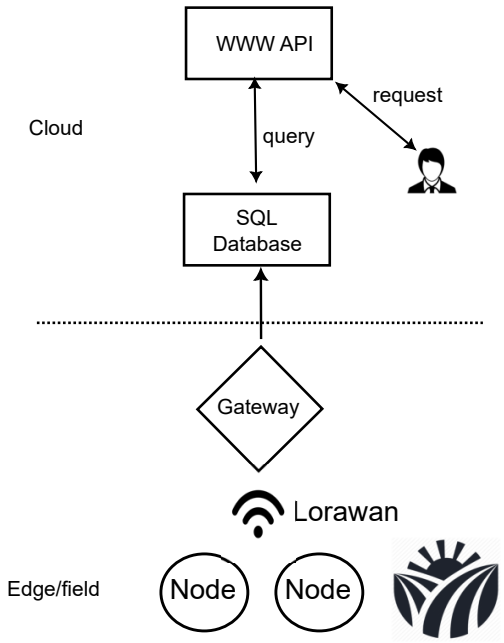


Figure 3: Complete scheme of the entire data reception structure

4.3. Description of the implementation

The experimental case study was performed in Arroyos y Esteros, Paraguay, 73 km from the country's capital, Asunción. It is located at geographical coordinates 57°04'50" West and latitude 25°05'40" South.

The field has dimensions of 100×20 meters, with a positive slope of 0.6% to the south. The soil is clayey loam, with a content of 54.13% sand, 21.20% silt and 24.67% clay. The crop is divided in 11 rows separated by 1.8 m. Each row has 200 sugarcane plants. The irrigation system consists of 10 sprinklers, with 180° coverage, organized in 2 groups of 5 equally separated sprinklers along the two longer borders of the field.

Twelve devices have been deployed. The first node is located at latitude 25°05'39.65"S and longitude 57°04'49.90"W. This point is taken as the origin of the spatial coordinates, $x = y = 0$. Next, the other devices are placed as indicated in Table 2, and as represented in Figure 4. The x-axis has been aligned towards the principal direction after performing a principal component analysis with the nodes coordinates, which results in a $-132.39°$ rotation with respect to the north-east frame (see Figure 4). The location of the nodes within the field was chosen considering the position of the sprinklers (see Figure 4) and its coverage region, which was circular. Nodes 1,4,6,7,12 were situated under the coverage of two sprinklers, while the others were placed under one unique sprinkler. Furthermore, a second objective was to ensure coverage of the entire area while also considering a spatial randomness component.

4.4. Main features of the data set

The installation of the nodes and their connection to the cloud layer was successfully done by March 4$^{\text{th}}$ 2022. Then, the data are accessible through the web API mentioned in

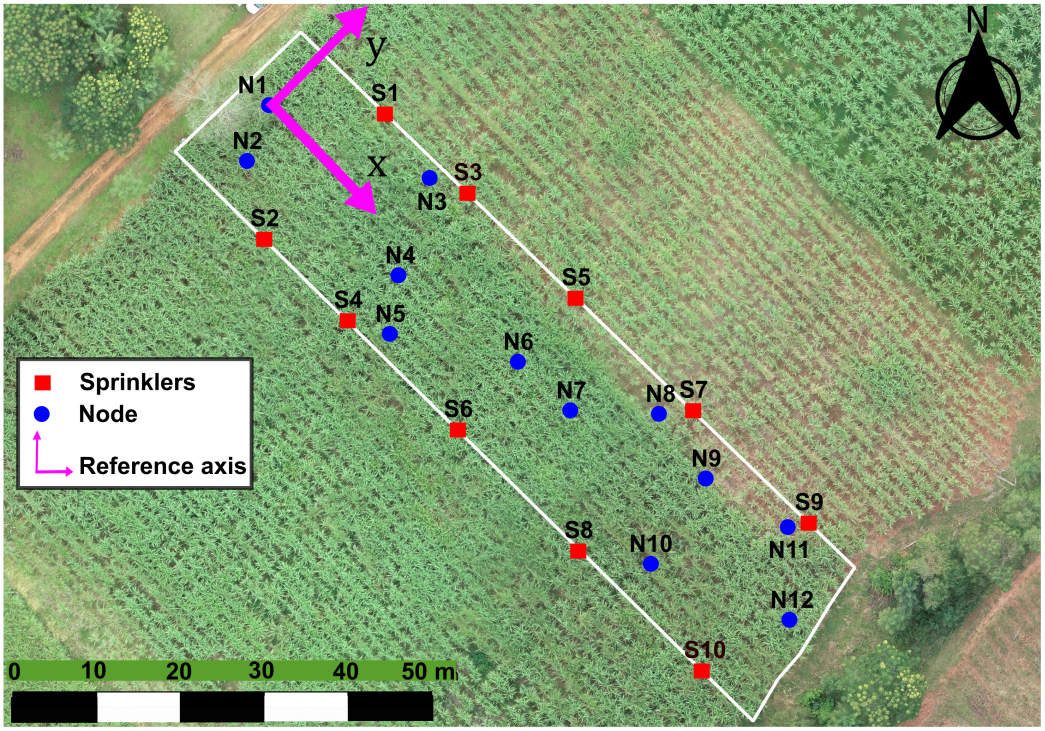


Figure 4: Satellite image of the field, showing the position of the nodes and sprinklers

Node	1	2	3	4	5	6	7	8	9	10	11	12
x (m)	0	2.29	23.00	25.70	31.30	47.45	58.84	63.81	73.96	77.08	85.77	90.34
y (m)	0	-9.16	3.90	-5.29	-11.43	-5.02	-5.86	0.33	0.02	-10.88	1.21	-7.53

Table 2: Spatial coordinates of the nodes

Section 4.2. The requests made to the API return data in `json` files.

In this study, data collected from 13:00:00 (UTC) on March 09$^{\text{th}}$ 2022 until 03:00:00 March 31$^{\text{st}}$ 2022 is used. In what is to follow, the time coordinate is expressed in hours, relatively to the first measurement. Hence, t ranges from 0 to 517.88 h.

The whole data set $\mathcal{D}$ contains $N = 15776$ data points. A time-varying representation is shown in Figure 5 for the moisture, and Figure 6 for the temperature. Irrigation (or rain) is noticeable on March 11$^{\text{th}}$ or 23$^{\text{rd}}$ in Figure 5, as well as the daily increase -and nightly decrease- in temperature in Figure 6.

The total time range of the dataset might seem quite short for a crop whose cycle lasts more than 12 months. While this is true in pure agronomic terms, the methodology described here pursues the estimation of variables whose dynamics have characteristic times notably shorter. Temperature presents a clear 24-hour period, as illustrated in Figure 6, whereas moisture depends on the irrigation pattern, as shown in Figure 5. Thus, the dataset contains enough points to illustrate the behaviour of the method, although it does not cover long-

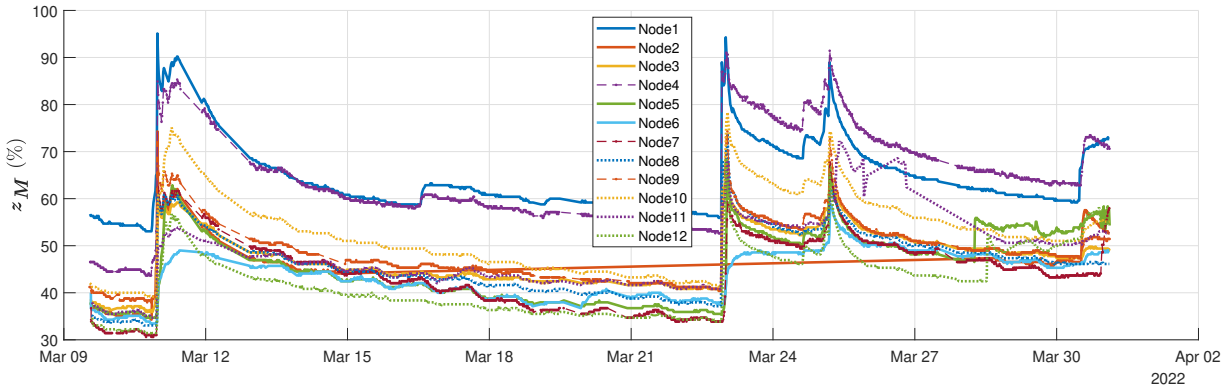


Figure 5: Temporal representation of the soil moisture data set

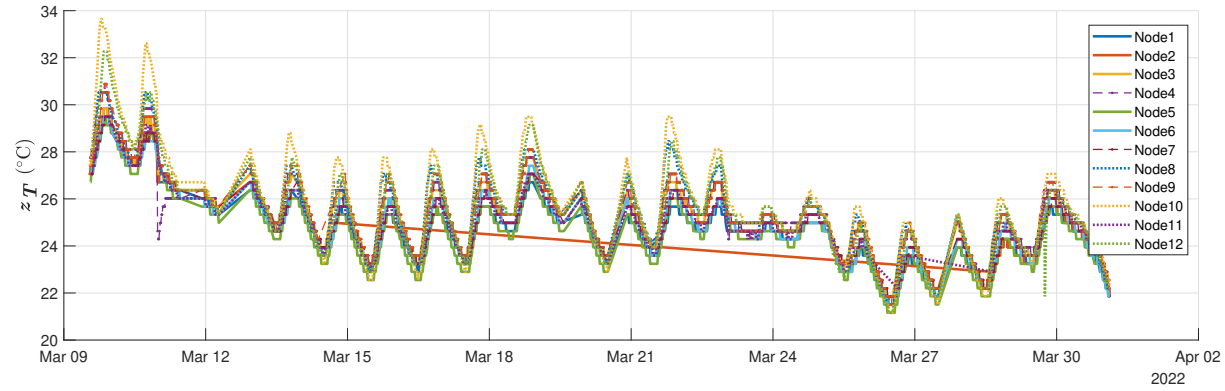


Figure 6: Temporal representation of the soil temperature data set

terms agronomic factors, such as climate or crop stage.

The data used in this article are made available in a Mendeley Data repository (see [omitted for blind-review]). The data are structured in several `csv` files, that can be easily imported to reproduce the results presented in this article.

The farm's data structure consists of two tables. The first table contains measurement data, including a timestamp in the format `DD-MMM-YYYY HH:mm:ss`, a node identifier ranging from 1 to 12, temperature $\tilde{z}_T$ in Celsius, and soil moisture $\tilde{z}_M$ as a percentage. The second table provides context data, including the node identifier and coordinates x and y measured in meters.

5. Results

The data set obtained as explained in Section 4.4 is used to analyze the estimation methods proposed in this manuscript, and to compare such methods against others available in expert literature.

Three different analysis are carried out. The first two of them are validation analysis:

- In the first one, the data set $\mathcal{D}$ is randomly split into two disjoint sets: $\mathcal{D}_{\text{learn}}$, which is used to train the learning algorithm; and $\mathcal{D}_{\text{val}}$, which is used to validate the predictions, assessing the performance of the method provided that the real (measured) output is known. See Section 5.1.

- In the second analysis, the data from nodes 4, 7 and 9 (see Figure 4) is used as $\mathcal{D}_{\text{val}}$, predicted using the data from the rest of nodes. See Section 5.2.

- The third analysis consists of the prediction over unseen points, that is, over locations where there are no nodes, and at times when no measurements where collected. See Section 5.3.

In all cases, the performance of the methods proposed in this work are compared to others: linear regression, a neural network, and two based on Gaussian processes.

But, first of all, a filtering of the data set was performed, in order to prune outliers. Noisy measurements may yield very large Lipschitz constants, even in consideration of the regularization term presented in Equation (12). Hence, the top 1% of points that generated the largest constants are removed from $\mathcal{D}$.

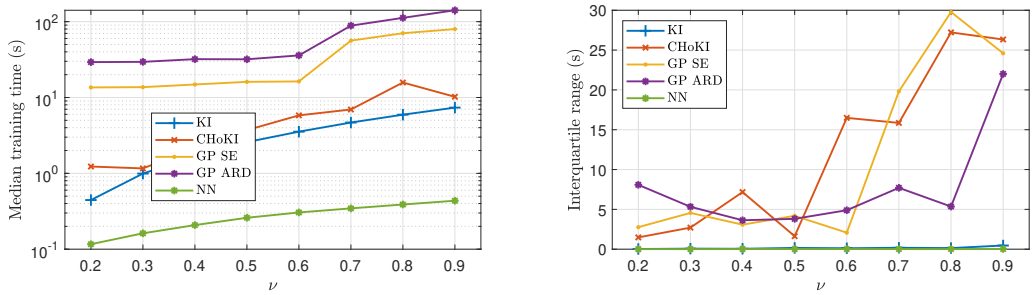
Remark. Note that, within the validation analysis, the whole available data set will be split into two subsets, learning and validation sets, i.e., $\mathcal{D} = \mathcal{D}_{\text{learn}} \cup \mathcal{D}_{\text{val}}$. At the same time, remember that the learning process of the CHoKI predictor consists of finding the Lipschitz matrix $\mathbf{L}_{\mathcal{D}}$ as per (16). As required by the method, the available data set $\mathcal{D}_{\text{learn}}$ will be further split into two disjoint subsets, training and test sets, i.e. $\mathcal{D}_{\text{learn}} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{test}}$. For the third analysis, the whole data set $\mathcal{D}$ will be used for the learning process, i.e. $\mathcal{D}_{\text{val}} = \emptyset, \mathcal{D}_{\text{learn}} \equiv \mathcal{D}$, again taking into account the separation between $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ in the training step.

5.1. Validation analysis with random split of the dataset

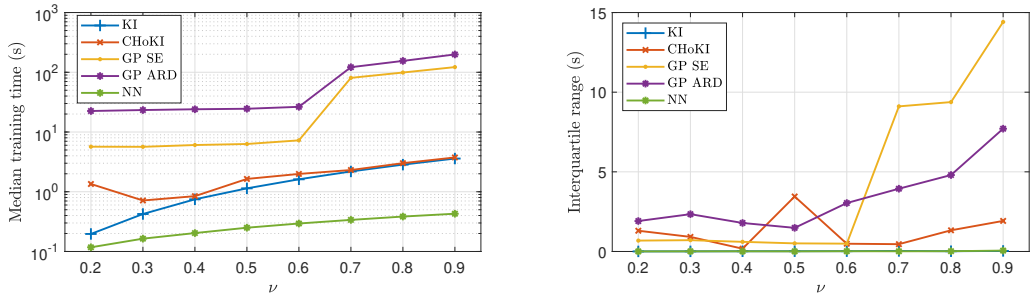
Six different learning algorithms are trained and validated: (i) the standard KI (11), (ii) the CHoKI (14), (iii) a Gaussian process with an squared-exponential (SE) kernel, (iv) a Gaussian process with an automatic relevance determination (ARD)-exponential kernel, (v) a linear regression algorithm and (vi) a neural network (NN).

The chosen neural network is composed of 4 hidden layers with 10 neurons per layer, and it is trained with the Levenberg-Marquardt back-propagation algorithm. The number of neurons and layers has been chosen as a trade-off between the predictive capabilities of the network and its computational requirements.

The results are shown next, and discussed in detail in Section 6. Six metrics are considered to assess the performance of the methods: two of them are related to the computational requirements, in particular, the training time and memory usage; three of them are related to the predictive capability of the methods; and finally, a last metric, only of application for the methods presented in this manuscript, is related to the computed Lipschitz constant, which is somehow related to the conservatism of the method. Large constants will generate large deterministic bounds for the estimated variables.



(a) Moisture



(b) Temperature

Figure 7: Training times, as a function of the number of data points in the training set

Since the value of those metrics might depend on the particular datasets used for training and validation, the sensibility of each metric to the number of points in the learning data set is analyzed. To do so, $\mathcal{D}_{\text{learn}}$ is constructed randomly with $\nu \times 100\%$ samples of $\mathcal{D}$, ν ranging from 0.2 to 0.9, increasing in steps of 0.1. With such $\mathcal{D}_{\text{learn}}$, the mentioned algorithms are trained and later validated over $N_{\text{val}} = 1500$ randomly chosen data points $\mathcal{D}_{\text{val}}$, not contained in $\mathcal{D}_{\text{learn}}$. In the CHoKI case, $\mathcal{D}_{\text{test}}$ is obtained choosing randomly 10% of the data contained in $\mathcal{D}_{\text{learn}}$, while the rest is used in $\mathcal{D}_{\text{train}}$.

Finally, to provide for extra robustness 20 repetitions are done for each configuration (random partition of the dataset) and the median and interquartile range of each metric are provided. This way, the sensitivity of the method with respect to the random choice of data is also analyzed. Of course, for each experiment, the same data is used to learn and validate each of the algorithms.

Let us begin with the metrics related to the computational requirements. Figure 7 depicts the training times of the five first methods (computed on an Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz with 8Gb of RAM). The training time of the linear regression is not shown, since it is way faster than the rest: between 0.063 and 0.12 ms.

It is also interesting to analyze the memory requirements to make predictions with these methods. Typically, non-parametric methods as KI or GPs suffer from the explicit dependence of the data set. The memory required to make a prediction using the algorithms is shown in Figure 8. Note that the memory is not a random variable, and thus, is not depicted its median or interquartile range. Note that the memory required for the linear

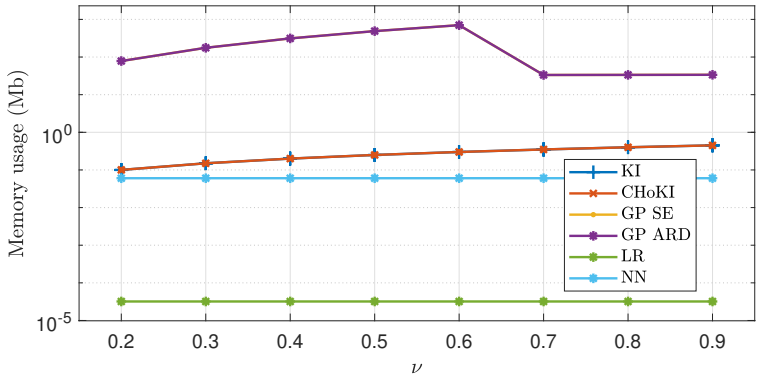


Figure 8: Memory usage by the algorithms to perform a prediction, as a function of the number of data points in the training set

regression algorithm is only 32 bits: three coefficients (for x, y, t) and an extra one for the bias term. The memory used for the NN does not vary with the data set size either, since it is a parametric method.

The predictive capability of the algorithms can be measured with different metrics that compare the predictions made with the methods, i.e. $\hat{z}_\sigma$, with the measured data in the validation set, i.e. $\tilde{z}_\sigma$. It is important to observe that the real value of the variables, i.e. z_σ, is unknown, since the measured data is corrupted by noise.

In this manuscript, three metrics are considered: the root-mean-square error (RMSE), the mean absolute error (MAE), and the R^2 determination coefficient, respectively, which are computed as:

$$\mathrm{RMSE} = \sqrt{\frac{\sum\limits_{\tilde{z}_\sigma \in \mathcal{D}_{\mathrm{val}}} \left(\tilde{z}_\sigma(x,y,t) - \hat{z}_\sigma(x,y,t)\right)^2}{N_{\mathrm{val}}}}, \tag{19a}$$

$$\mathrm{MAE} = \frac{\sum\limits_{\tilde{z}_\sigma \in \mathcal{D}_{\mathrm{val}}} |\tilde{z}_\sigma(x,y,t) - \hat{z}_\sigma(x,y,t)|}{N_{\mathrm{val}}}, \tag{19b}$$

$$R^2 = 1 - \frac{\sum\limits_{\tilde{z}_\sigma \in \mathcal{D}_{\mathrm{val}}} (\tilde{z}_\sigma(x,y,t) - \hat{z}_\sigma(x,y,t))^2}{\sum\limits_{\tilde{z}_\sigma \in \mathcal{D}_{\mathrm{val}}} (\tilde{z}_\sigma(x,y,t) - m(\tilde{z}_\sigma(x,y,t)))^2}, \tag{19c}$$

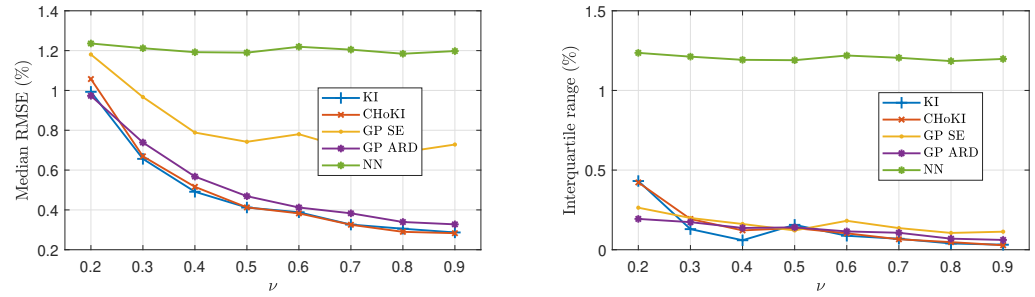
being $m(s)$ the mean of s. Figures 9, 10 and 11 include the analysis for each metric.

RMSE and MAE focus on error magnitudes, with RMSE being more sensitive to large discrepancies, whereas R^2 provides an overall view of the model's explanatory power. Complementing these metrics is essential for a comprehensive evaluation of the machine learning methods presented [42].

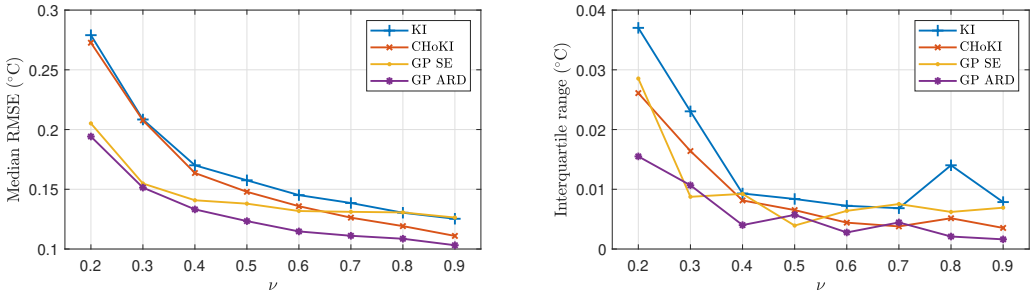
These predictive metrics are not shown for the linear regression algorithm, since the obtained predictive performance is very poor, and the inclusion in the figures would impede a clear analysis of the performance of the rest of algorithms. This can be easily understood

with the medians of each metric obtained with linear regression, which is: for temperature RMSE of 1.34°C, MAE of 1°C and R^2 of 0.36; for moisture RMSE of 9.15%, MAE of 7%, and R^2 of 0.19. Similarly, in the temperature case, the neural network does not perform well either, with a median RMSE of 0.8°C, and it is therefore also omitted in the figures.

It is worth remarking that building a larger and more complex neural network results in better predictive capabilities. But, even in those cases, the performance was always worse than that of the non-parametric methods.

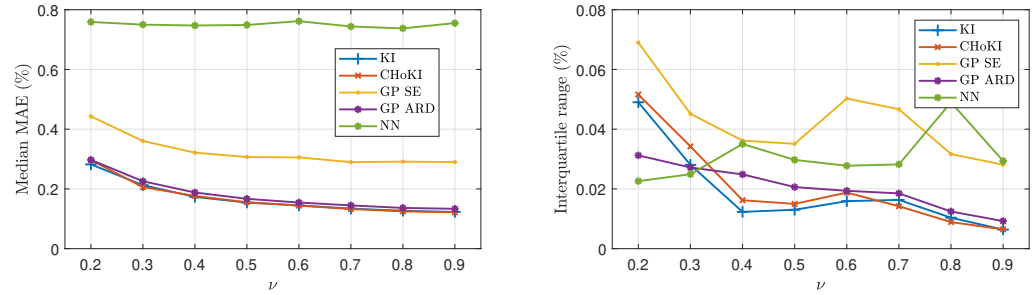


(a) Moisture

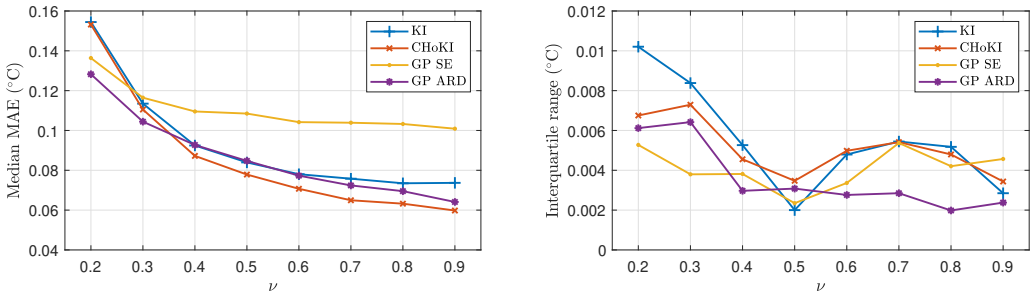


(b) Temperature

Figure 9: Root-mean-square error, as a function of the number of data points in the training set

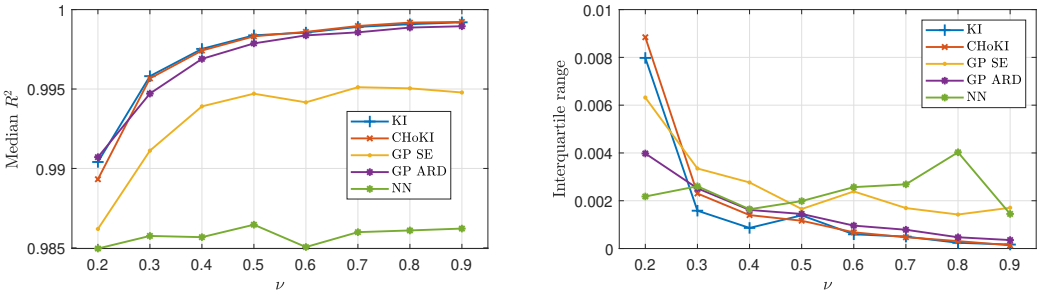


(a) Moisture

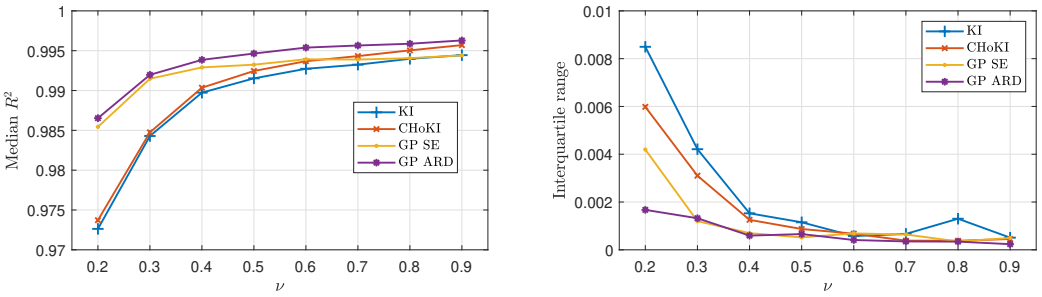


(b) Temperature

Figure 10: Mean absolute error, as a function of the number of data points in the training set

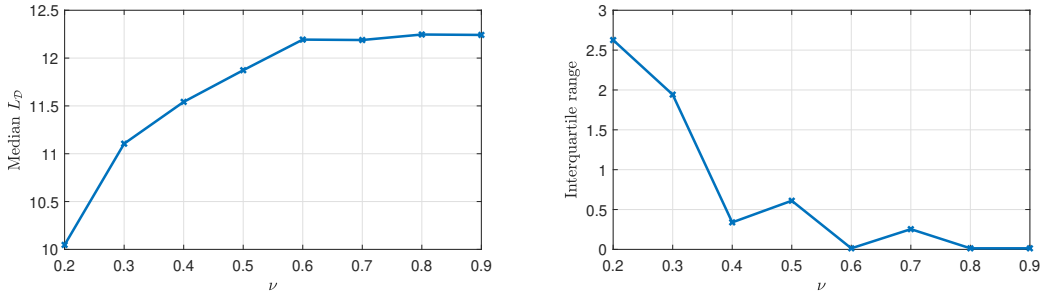


(a) Moisture

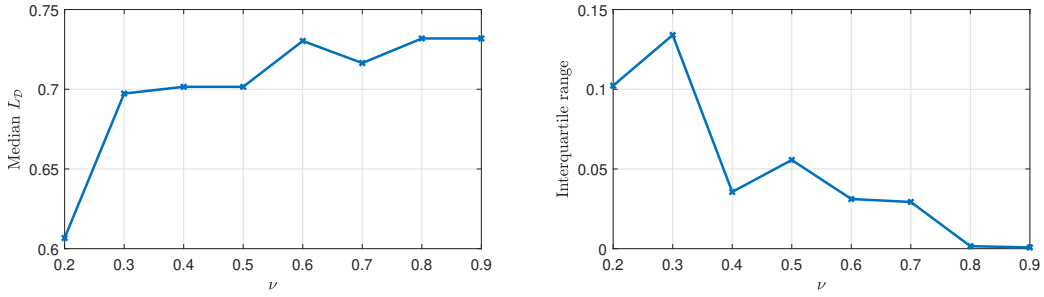


(b) Temperature

Figure 11: R^2 coefficient, as a function of the number of data points in the training set



(a) Moisture



(b) Temperature

Figure 12: Estimated Lipschitz constant via the LACKI algorithm, as a function of the number of data points in the training set

Finally, and only for the methods proposed in this manuscript, the estimated Lipschitz constants through equations (12)-(16) are represented in Figures 12 and 13, respectively.

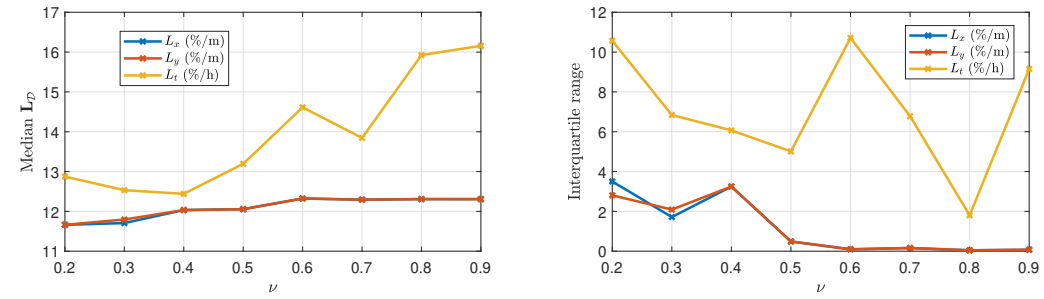
5.2. Validation analysis over unseen nodes

The analysis is done now removing the data from nodes 4, 7 and 9 (see their location in Figure 4 or Table 2) from $\mathcal{D}_{\text{learn}}$. The six machine learning algorithms mentioned before are trained using the data generated by the rest of the nodes, and validated over the mentioned nodes. The results are shown in Figure 14. The resulting prediction obtained with the linear regression and neural network algorithms are omitted, since they do not capture the dynamics as well as the non-parametric methods. In this case, the prediction achieved with all the methods is shown for the whole length of the experiment, and compared with the measured data.

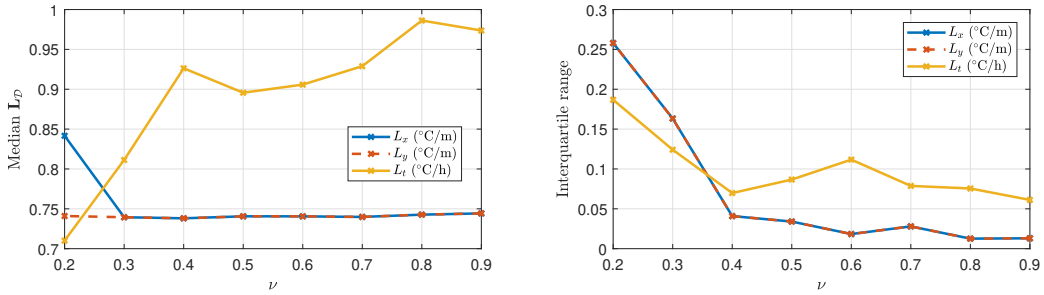
5.3. Estimation analysis

As a final illustration of the capabilities of the proposed method, predictions are made over the field, at locations where no sensors were deployed, and at times when no measurements were collected. The estimation methods allow to predict the value of the variables, based on certain measurements. Here, the whole filtered data set is used for training and predicting purposes. No validation can be done in this case, as there is no available data.

In order to predict over the whole field (see Section 4.3), a mesh grid is considered along the spatial coordinates, dividing the ranges $-5 \leq x \leq 95$ into 200 equally spaced samples,



(a) Moisture

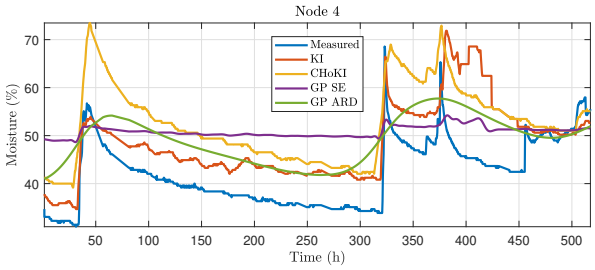


(b) Temperature

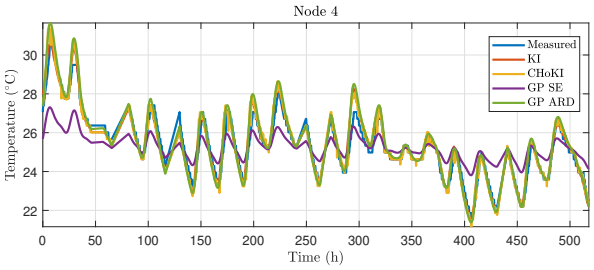
Figure 13: Estimation of the componentwise Lipschitz constants via the CHoKI algorithm, as a function of the number of data points in the training set

and $-15 \leq y \leq 6$ in 42 equally spaced samples, which results in 8400 spatial coordinates in which to estimate the unknown moisture and temperature functions.

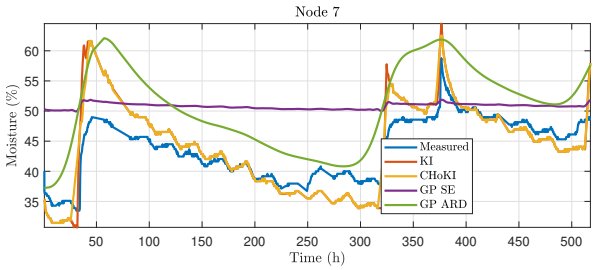
Next, the queries are constructed assigning a time (i.e., temporal coordinate) in which the prediction is to be made. The estimation results are shown with a colour map in Figure 15 when $t = 100\,\text{h}$. The circles in the figure represent, by means of their color, the moisture measured at that location (where the nodes are), at the time as close as possible to the query time (100 h in this case). Remember that the measurements are collected each 20 min. The results are only shown for the non-parametric methods, since the parametric ones render very poor estimates.



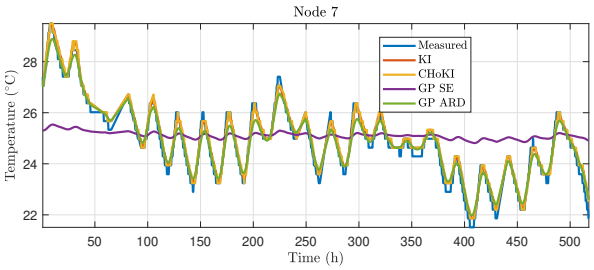
(a) Node 4 moisture



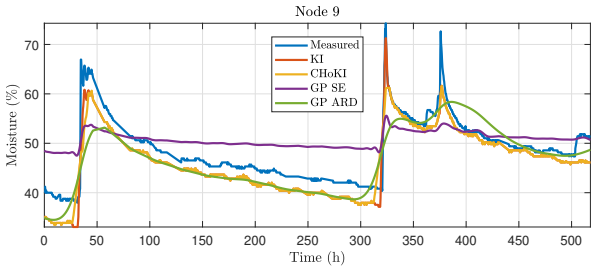
(b) Node 4 temperature



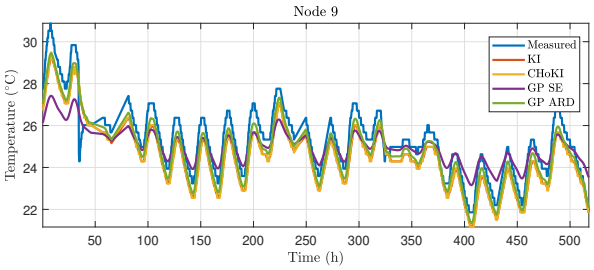
(c) Node 7 moisture



(d) Node 7 temperature

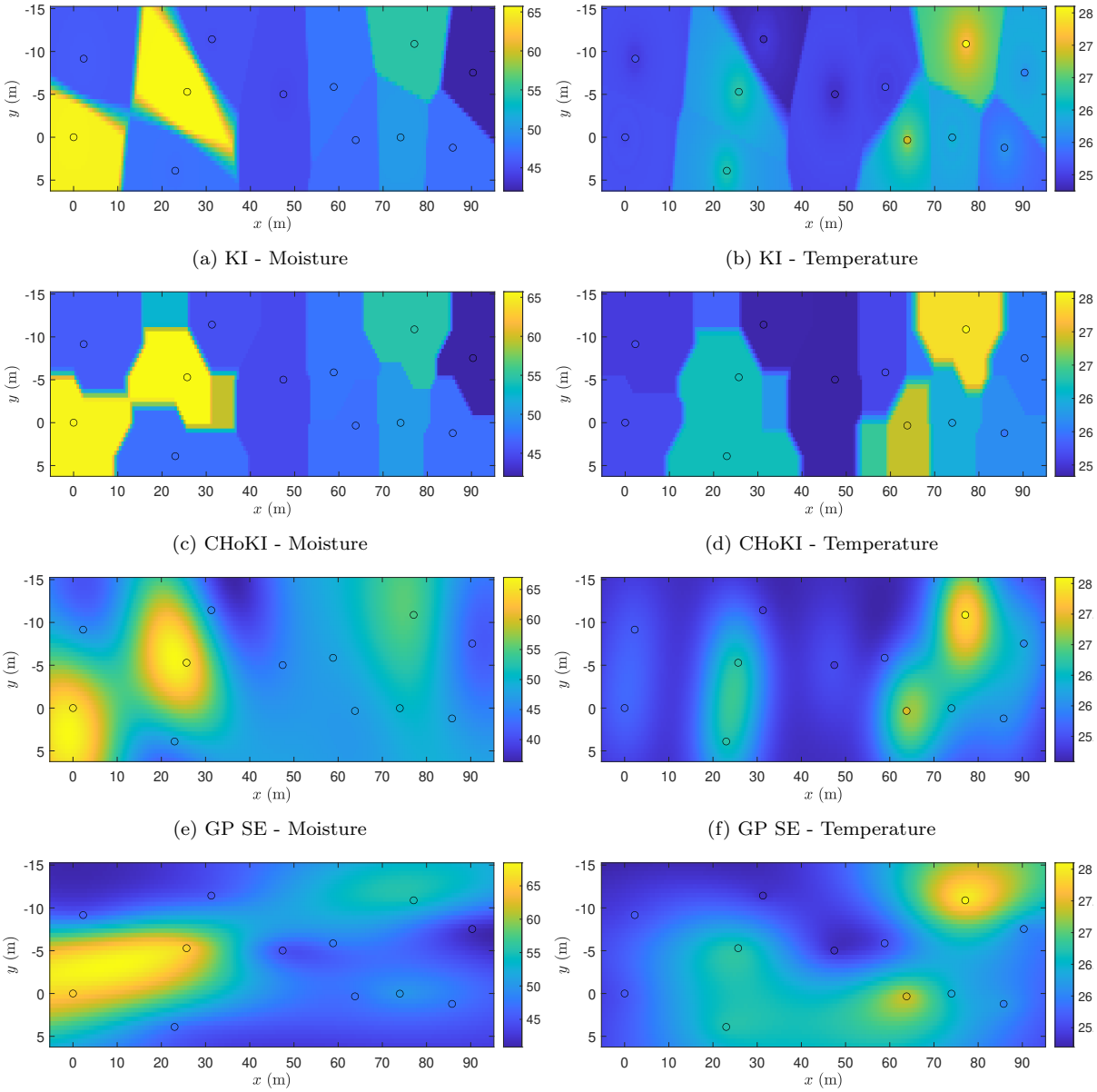


(e) Node 9 moisture



(f) Node 9 temperature

Figure 14: Validation over unseen nodes



(a) KI - Moisture

(b) KI - Temperature

(c) CHoKI - Moisture

(d) CHoKI - Temperature

(e) GP SE - Moisture

(f) GP SE - Temperature

(g) GP ARD - Moisture

(h) GP ARD - Temperature

Figure 15: Estimation of the soil moisture and temperature over the whole field, at time $t = 100\,\mathrm{h}$

6. Discussion

Main results and findings. The results shown in the previous section reveal that the estimation methods based on Lipschitz interpolation are able to produce reliable spatio-temporal estimations of agricultural variables in the soil.

As expected, increasing the size of the data set used for training impacts the predictive metrics, at the cost of higher training times.

Taking the influence of the spatial and temporal coordinates on the moisture separately (see $\mathbf{L}$ in (16)) allows the CHoKI method to slightly decrease prediction errors, in comparison to the standard KI procedure.

Comparison with other methods. The comparison with the methods based on Gaussian processes, linear regression and neural network is very rich.

The first aspect that should be highlighted is that parametric methods (linear regression and neural networks) behave worse in terms of their predictive capabilities. This can be explained, somehow, by the very nature of the function to be learn. Since the independent variables are the spatial and temporal coordinates, it is unlikely that a linear function of x, y, t capture the complex dynamics observed with the data. Neural networks might obtain better results, since they can implement nonlinear functions. But the real improvement is obtained when the prediction is based not only on those independent variables, but also on the available data. This is why non-parametric methods outperforms the other ones in this case.

Concerning the non-parametric methods, and or the first validation analysis, it is observed that the median and interquartile range of the errors for the methods based on Lipschitz interpolation tend to be similar to those obtained with GP-ARD, as the size of the training set grows (See Fig. 9). KI-based methods offer better results in terms of MAE and R^2. RMSE in the estimation of the temperature is small for GPs, noting that it might benefit from the diffusion properties of this variable. From Figures 9-11, one can conclude that using a standard exponential kernel (GP-SE) produces bigger errors, and much larger variability, specially for the moisture. However, for these four methods analyzed, the estimation errors are smaller than the measurement noise, so all of them can be considered as good estimators, attending to this validation analysis.

Regarding the computational metrics, the observed training times and memory usage is much smaller for the proposed methods than for GPs. Please notice that Figures 7 and 8 use a logarithmic scale, so the reduction of computational time and memory is substantial.

It is worth mentioning that Gaussian processes require prior knowledge to correctly choose an appropriate kernel. In absence of such knowledge, the engineer's expertise when choosing the kernel and its prior parameters has a crucial effect on the result.

The second analysis reveals new details hidden in the former. Remember that, for this case, the estimations are made over positions where no training data is available, since the data from the nodes located at those positions is used only for validation purposes. Now, the estimation performances obtained for temperature and moisture are very different, as depicted in Fig. 14.

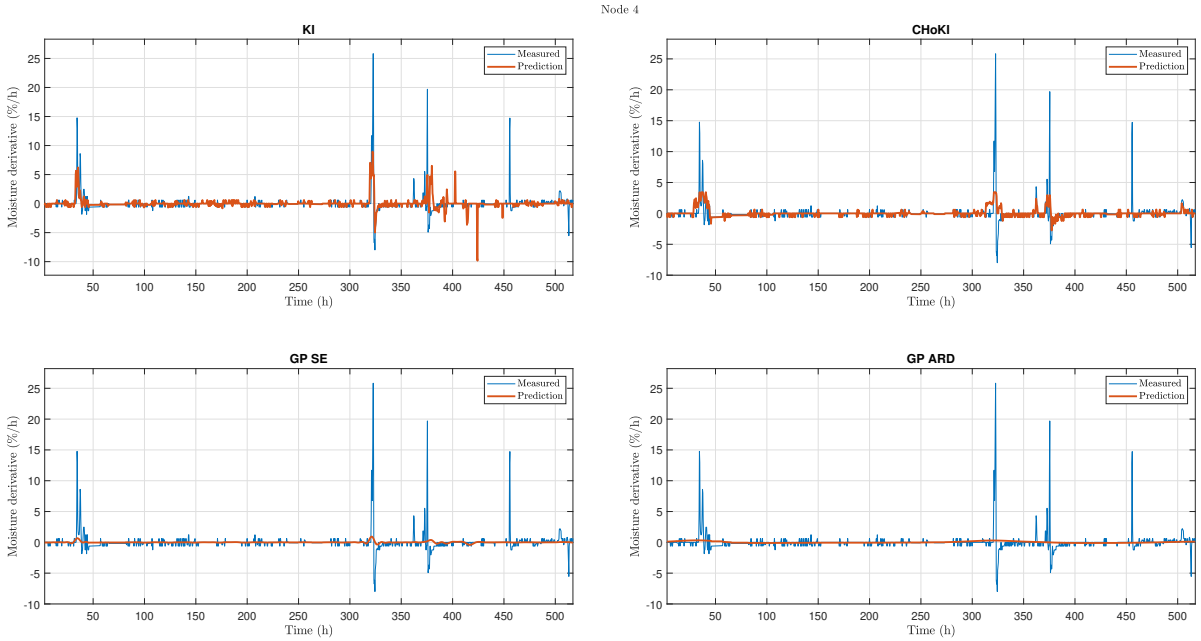


Figure 16: Numerical derivative of the moisture, both for the measured and predicted values, using the four methods, over unseen data of node 4

Concerning temperature estimation, the performance of the proposed methods and the GP-ARD is quite similar, and the estimations are quite similar to the measured data. Please remember that the real temperature at those positions is unknown. The GP-SE produces estimations that are far from the measured data and, probably, far from the real variable. The reason behind this last conclusion is that temperature diffuses in the soil, and the same kind of variations is expected in all positions, provided that the temperature sensors are quite reliable, as explained in Section 4. The bias observed in Fig. 14f could be explained by a bad calibration of node 9.

On the other hand, by observing Figures 14a, 14c and 14e one might conclude that none of the methods is useful to make moisture predictions over unseen coordinates. However, and by taking into account that the moisture data collected by the sensors is quite local (see Section 4), the absolute value is not that important. Then, one might consider that the performance of the estimator is adequate if the trends are captured.

So, paying again special attention to those figures, one can realize that the trends are not well-captured by GP-SE. GP-ARD is able to catch the low-frequency trends, but fails to capture sudden changes, as those provoked by rains or irrigation. The proposed methods clearly outperform those based on GPs. This can be better noticed in Fig. 16, where the errors between the derivative of the measured data and the derivative of the predictions made with the four non-parametric methods is shown.

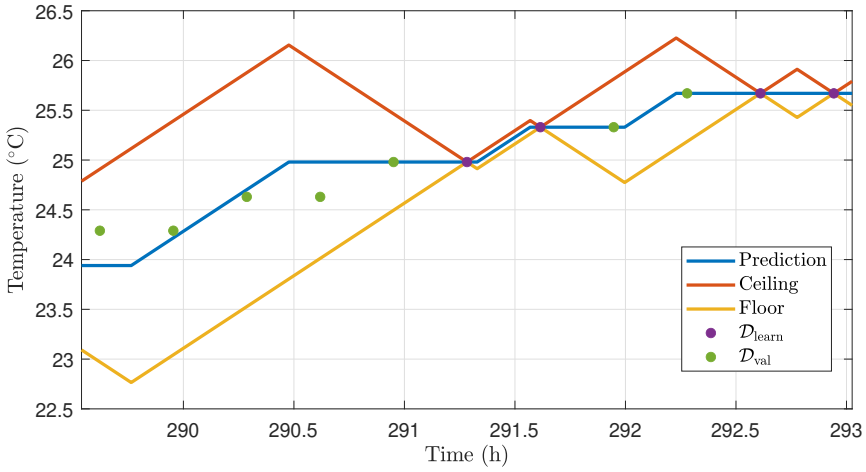


Figure 17: Prediction error bound, determined by the enclosure, obtained over a validation process

Strengths and limitations. The main strengths of the proposed methodology are its simplicity and its scalability to higher dimensional spaces. The method has also proven itself adequate for estimation and prediction. It fails when predicting the moisture over unseen coordinates, but is able to produce an adequate estimation of the trends. No previous expertise is required to train the method or making predictions with it.

An important issue of the method is related to the impact of the outliers to the estimated value of the Lipschitz constants. Filtering outliers has shown positive results in the posterior prediction stage. If not, Lipschitz constant would be much larger, and the deterministic bounds of the prediction of the variables would be rather large and, possibly, useless. Again, the fact the moisture data is quite local makes that, at the same instant, two nodes that are physically close, might collect very different value for the moisture and, then, the Lipschitz constant required to fit the data grows.

An illustration of the deterministic bounds on the uncertainty provided by the CHoKI method (see Problem 1) is shown in Figure 17. Using the validation setup mentioned in Section 5.1, 60% of the data is removed from $\mathcal{D}_{\text{learn}}$. The CHoKI constant are estimated, and the prediction is computed. Figure 17 also shows the enclosure obtained, over a three-hour-long prediction process over node number 11.

Future works. By visual inspection of the moisture data set in Figure 5, one can observe that the dynamical behaviour of the variable is completely different when the field is being irrigated than in the evapotranspiration phase. It will be beneficial to have two different Lipschitz constants, one when the system is irrigated and one when it is not. How to distinguish rain from irrigation is something to be analysed in the future.

Validating the method with data from other seasons is also sought. Although the validation data used in this study was collected in March, the non-parametric nature of the proposed method allows it to adapt effectively to different temporal scenarios, including other seasons. The method's reliance on real-time, online temperature measurements en-

sures continuous adjustment, minimizing the risk of overfitting to data from a specific period.

7. Conclusion

This work described an application of the kinky inference machine learning algorithm, which has been adapted to learn and to predict spatio-temporal variability of both soil moisture and temperature in agricultural fields.

The work also analyses the deployment of a monitoring network for soil temperature and moisture in a sugarcane field in Paraguay. The results illustrate the performance of the algorithm over the observed data. Strengths and limitations, as well as a comparison with other estimation methods in the literature, have been presented.

Some future modifications of the proposed method were given in the Discussion. Apart from those, we intend to explore a distributed formulation of the algorithms, to avoid the concentration of data into a single device. Furthermore, we will exploit this information to devise irrigation strategies and decision making procedures, thanks to the information obtained from the suggested method. Finally, it seems crucial to explore the spatial position in which the sensors are deployed in order to maximize the performance of the methods.

Acknowledgments

Omitted for blind-review process.

References

[1] J. Dong, J. G. Burnham, B. Boots, G. Rains, F. Dellaert, 4D crop monitoring: Spatio-temporal reconstruction for agriculture, in: IEEE International Conference on Robotics and Automation, IEEE, 2017, pp. 3878–3885. doi:10.1109/ICRA.2017.7989447.

[2] D. Corwin, S. Lesch, Characterizing soil spatial variability with apparent soil electrical conductivity: I. Survey protocols, Computers and Electronics in Agriculture 46 (1-3) (2005) 103–133. doi:10.1016/j.compag.2004.11.002.

[3] H. M. Ibrahim, D. R. Huggins, Spatio-temporal patterns of soil water storage under dryland agriculture at the watershed scale, Journal of Hydrology 404 (3-4) (2011) 186–197. doi:10.1016/j.jhydrol.2011.04.029.

[4] J. Lloret, S. Sendra, L. Garcia, J. M. Jimenez, A wireless sensor network deployment for soil moisture monitoring in precision agriculture, Sensors 21 (21) (2021) 7243. doi:10.3390/s21217243.

[5] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat, P. Nillaor, IoT and agriculture data analysis for smart farm, Computers and Electronics in Agriculture 156 (2019) 467–474. doi:10.1016/j.compag.2018.12.011.

[6] S. N. Patil, M. B. Jadhav, Smart agriculture monitoring system using IoT, International Journal of Advanced Research in Computer and Communication Engineering 8 (4) (2019) 116–120. doi:10.17148/IJARCCE.2019.8419.

[7] T. Van Waterschoot, G. Leus, Distributed estimation of static fields in wireless sensor networks using the finite element method, in: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2012, pp. 2853–2856. doi:10.1109/ICASSP.2012.6288512.

[8] A. Mondal, D. Khare, S. Kundu, S. Mondal, S. Mukherjee, A. Mukhopadhyay, Spatial soil organic carbon (soc) prediction by regression kriging using remote sensing data, The Egyptian Journal of Remote Sensing and Space Science 20 (1) (2017) 61–70. doi:10.1016/j.ejrs.2016.06.004.

[9] F. Sasso, A. Coluccia, G. Notarstefano, An empirical Bayes approach for distributed estimation of spatial fields, in: 2018 European Control Conference (ECC), IEEE, 2018, pp. 2206–2211. doi:10.23919/ECC.2018.8550231.

[10] S. Martinez, Distributed interpolation schemes for field estimation by mobile sensor networks, IEEE Transactions on Control Systems Technology 18 (2) (2009) 491–500. doi:10.1109/TCST.2009.2017028.

[11] D. Yan, Q. Zhong, Y. Sui, Spatial Kalman filters and spatial-temporal Kalman filters, in: 2014 12th International Conference on Signal Processing (ICSP), IEEE, 2014, pp. 1902–1905. doi:10.1109/ICOSP.2014.7015323.

[12] N. Cressie, C. K. Wikle, Space-time Kalman filter, Wiley StatsRef: Statistics Reference Online (2014). doi:10.1002/9781118445112.stat07813.

[13] J. Cortés, Distributed kriged Kalman filter for spatial estimation, IEEE Transactions on Automatic Control 54 (12) (2009) 2816–2827. doi:10.1109/TAC.2009.2034192.

[14] P. K. Shit, G. S. Bhunia, R. Maiti, Spatial analysis of soil properties using GIS based geostatistics models, Modeling Earth Systems and Environment 2 (2) (2016) 1–6. doi:10.1007/s40808-016-0160-4.

[15] N. Cressie, C. K. Wikle, Statistics for spatio-temporal data, John Wiley & Sons, 2015. doi:10.1111/j.1541-0420.2012.01835.x.

[16] W. Stepniewski, H. Sobczuk, M. Widomski, Encyclopedia of Agrophysics, Springer Netherlands, Dordrecht, 2011, Ch. Diffusion in Soils, pp. 214–220. doi:10.1007/978-90-481-3585-1-273.

[17] T. R. Tenreiro, M. García-Vila, J. A. Gómez, J. A. Jimenez-Berni, E. Fereres, Water modelling approaches and opportunities to simulate spatial water variations at crop field level, Agricultural Water Management 240 (2020) 106254. doi:10.1016/j.agwat.2020.106254.

[18] K. Eldrandaly, A. Abdelmouty, Spatio-temporal interpolation: Current practices and future prospects, International Journal of Digital Content Technology and its Applications 11 (06) (2017) 2017.

[19] L. Li, P. Revesz, A comparison of spatio-temporal interpolation methods, in: International Conference on Geographic Information Science, Springer, 2002, pp. 145–160.

[20] T. Gantala, K. Balasubramaniam, Implementing data-driven approach for modelling ultrasonic wave propagation using spatio-temporal deep learning (sdl), Applied Sciences 12 (12) (2022) 5881.

[21] F. Amato, F. Guignard, S. Robert, M. Kanevski, A novel framework for spatio-temporal prediction of environmental data using deep learning, Scientific reports 10 (1) (2020) 22243.

[22] H. Jia, M. Zheng, P. Wang, T. Li, X. Zheng, Big data-driven spatio-temporal heterogeneity analysis of beijing's catering service industry during the covid-19 pandemic, Scientific Reports 14 (1) (2024) 721.

[23] N. Cressie, The origins of kriging, Mathematical geology 22 (3) (1990) 239–252. doi:10.1007/BF00889887.

[24] C. K. Williams, C. E. Rasmussen, Gaussian processes for machine learning, Vol. 2(3), MIT press Cambridge, MA, 2006. doi:10.7551/mitpress/3206.001.0001.

[25] C. K. Gasch, T. Hengl, B. Gräler, H. Meyer, T. S. Magney, D. J. Brown, Spatio-temporal interpolation of soil water, temperature, and electrical conductivity in 3D+ T: The Cook Agronomy Farm data set, Spatial Statistics 14 (2015) 70–90. doi:10.1016/j.spasta.2015.04.001.

[26] J. E. Whitman, H. Maske, H. A. Kingravi, G. Chowdhary, Evolving Gaussian processes and kernel observers for learning and control in spatiotemporally varying domains: With applications in agriculture, weather monitoring, and fluid dynamics, IEEE Control Systems Magazine 41 (1) (2021) 30–69. doi:10.1109/MCS.2020.3032801.

[27] J. M. Manzano, L. Orihuela, E. Pacheco, M. Pereira, Results on spatio-temporal estimation of temperature and soil moisture in La Colmena (Paraguay), in: 7th AGRICONTROL, IFAC, 2022, p. Accepted.

[28] Z. Hong, Z. Kalbarczyk, R. K. Iyer, A data-driven approach to soil moisture collection and prediction, in: 2016 IEEE International Conference on Smart Computing (SMARTCOMP), IEEE, 2016, pp. 1–6.

[29] J. Li, A. D. Heap, Spatial interpolation methods applied in the environmental sciences: A review, Environmental Modelling & Software 53 (2014) 173–189.

[30] J.-P. Calliess, Conservative decision-making and inference in uncertain dynamical systems, Ph.D. thesis, University of Oxford, Oxford (2014).

[31] M. Milanese, C. Novara, Set membership identification of nonlinear systems, Automatica 40 (6) (2004) 957–975. doi:10.1016/j.automatica.2004.02.002.

[32] G. Beliakov, Interpolation of Lipschitz functions, Journal of computational and applied mathematics 196 (1) (2006) 20–44. doi:10.1016/j.cam.2005.08.011.

[33] J. M. Nadales, J. M. Manzano, A. Barriga, D. Limon, Efficient FPGA parallelization of Lipschitz interpolation for real-time decision making, IEEE Transactions on Control Systems Technology (2022). doi:10.1109/TCST.2021.3136616.

[34] L. A. Richards, Capillary conduction of liquids through porous mediums, Physics 1 (5) (1931) 318–333.

[35] J. Qin, S. Liang, K. Yang, I. Kaihotsu, R. Liu, T. Koike, Simultaneous estimation of both soil moisture and model parameters using particle filtering method through the assimilation of microwave signal, Journal of Geophysical Research: Atmospheres 114 (D15) (2009). doi:10.1029/2008JD011358.

[36] J.-P. Calliess, S. J. Roberts, C. E. Rasmussen, J. Maciejowski, Lazily adapted constant kinky inference for nonparametric regression and model-reference adaptive control, Automatica 122 (2020) 109216. doi:10.1016/j.automatica.2020.109216.

[37] J. M. Manzano, D. Muñoz de la Peña, J.-P. Calliess, D. Limon, Componentwise Hölder inference for robust learning-based MPC, IEEE Transactions on Automatic Control (2021) 5577–5583doi:10.1109/TAC.2021.3056356.

[38] J. M. Manzano, J. Calliess, D. Muñoz de la Peña, D. Limon, Online learning robust MPC: an exploration-exploitation approach, IFAC-PapersOnLine 53 (2) (2020) 5292–5297. doi:10.1016/j.ifacol.2020.12.1210.

[39] J. M. Manzano, D. Limon, D. Muñoz de la Peña, J.-P. Calliess, Robust learning-based MPC for nonlinear constrained systems, Automatica 117 (2020) 108948. doi:10.1016/j.automatica.2020.108948.

[40] D. Aranda, A. Tapia Córdoba, P. Millan Gata, Calibración y caracterización de sensores capacitivos de bajo coste para la monitorización de humedad de suelo, in: XLIII Jornadas de Automática, Universidade da Coruña. Servizo de Publicacións, 2022, pp. 479–485.

[41] J. de Carvalho Silva, J. J. Rodrigues, A. M. Alberti, P. Solic, A. L. Aquino, LoRaWAN—A low power WAN protocol for Internet of Things: A review and opportunities, in: 2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech), IEEE, 2017, pp. 1–6.

[42] D. Chicco, M. J. Warrens, G. Jurman, The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation, Peerj computer science 7 (2021) e623.