ESAIM: PS 30 (2026) 354–371
https://doi.org/10.1051/ps/2026005

ESAIM: Probability and Statistics
www.esaim-ps.org

TOWARDS HISTORY-AWARE SENSITIVITY ANALYSIS FOR TIME SERIES

Mouad Yachouti[1,*], Guillaume Perrin[2] and Josselin Garnier[1]

Abstract. Explaining the outcome of dynamical systems is non-trivial due to the temporal nature and correlation of the variables. In this work, we propose a novel framework of history-aware sensitivity analysis for stationary time-series to quantify different memory effects and clarify their roles. For this purpose, we decompose the output time series into non-correlated components, namely the instantaneous component and the memory components. The latter are sorted in decreasing order of variance to reflect the importance of the variables. We highlight the compensation phenomena between the resulting components and illustrate them in the case of independent variables in a linear setting. To enable history-aware explanations, variance-based sensitivity indices are derived from the obtained decomposition. We demonstrate the effectiveness of our methodology in providing insights to explain output time-series in both synthetic and real-world cases.

Mathematics Subject Classification. 37M10, 60G10, 68T05.

Received September 12, 2025. Accepted April 28, 2026.

1. Introduction

Large amounts of historical data are collected to monitor complex dynamical systems, especially in industry and other mission-critical applications, such as in nuclear industry, finance, manufacturing, *etc.* Numerical models have been extensively used to mimic the behavior of these systems, notably thanks to the success of black-box models such as Gaussian processes [1, 2] and deep learning models [3]. Explaining a system's outcome or their surrogates based on the input variables is a crucial need; detecting the most influential variables improves the understanding of the system and its dynamic and, moreover, allows effective maintenance policies and facilitate the compliance with current legislation such as General Data Protection Regulation (GDPR) or regulations to ensure nuclear safety. To this end, rich frameworks, including uncertainty quantification (UQ), explainable AI (XAI), and Sensitivity analysis (SA) among others, have been developed in the last two decades to tackle this problematic and build confidence.

In practice, performing such techniques to assess variables importance for dynamical models is a non-trivial task and requires a history-aware methodology [4] due to the functional nature of the variables and the response, the temporal correlation of the inputs (for example, the presence of a daily periodicity in the data), or lag or delay effects (*i.e.* the fact that the impact of an event at a time t is observed at a time $t+\tau$ for a non-negligible

Keywords and phrases: Sensitivity analysis, time series, explainability.

[1] CMAP, CNRS, École polytechnique, Institut Polytechnique de Paris, 91120 Palaiseau, France.
[2] COSYS, Université Gustave Eiffel, 14-20 Boulevard Newton, 77447 Marne-la-Vallée, France.

* Corresponding author: mouad.yachouti@polytechnique.edu

response time τ) which is frequent in economic, medical, and environmental data, *etc.* In this paper, we assume that we are provided with a sample of input multivariate time series and the corresponding sample of outputs. We aim at explaining the latter based on the input variables by quantifying different memory effects of their underlying dynamics, notably, the instantaneous and the linear memory effects.

To extract meaningful knowledge, SA studies how input uncertainties are attributed to the uncertainty observed on the output of a model [5]. In particular, the framework of variance-based SA [6] and the so-called Sobol indices [7] allow to decompose the output's variance into different parts and attribute each one to individual (or combined) input variances. For multivariate or time dependent models, performing SA on separate scalar outputs, using univariate Sobol indices for instance, can provide some information on the evolution of the sensitivity over time. This approach, however, introduces redundancy and may miss some dynamical features. To overcome this, the work of [8] on outputs that are functions and later the work of [9] that focused on time dependent outputs both introduced frameworks that consist in extracting orthogonal features (based on principal component analysis) and then performing SA on the most informative components individually; for each component, a global sensitivity index can be derived. In a more general setting, in [10], new generalized Sobol indices are constructed for multivariate or functional outputs with updated pick and freeze estimators. Building on [10] and [9], [4] focused on the study of global SA for time dependent models of the form $Y_t = f(X, t)$ and proposed generalized Sobol indices that are equivalent to the work of [9]. The main contribution of the work [4] resides in improving computational efficiency and avoiding redundancy through a combination of Polynomial chaos [11] and Karhunen-Loeve expansions. Finally, in the context of dependent variables, [12] alleviates the assumption of independence of variables by using copulas [13] and proposes a method for both time dependent models $Y_t = f(X, t)$ and models defined by means of stochastic process $Y_t = f(X_t)$.

Our work is also related to HOFD (hierarchically orthogonal functional decomposition) introduced in [14] to generalize the Hoeffding-Sobol decomposition in case of dependent variables and mGS-PCE (modified Gram–Schmidt based polynomial chaos expansion) proposed in [15]. Both methods are iterative and leverage the ability of the Gram–Schmidt process to handle dependency and correlation between variables. In this work, this process is used to handle, as a first step, the temporal correlation between the instantaneous variables and the memory components, and, as a second step, to handle correlation among the memory components.

To the best of our knowledge, explaining outputs of the form $Y_t = f(X_s, s \leq t)$ which is the setting of this work remained an open question in the literature around SA or XAI methods; the setting often considered is time-series inputs and a scalar output (in the case of regression or classification tasks for instance). We propose then a methodology to perform a quantitative variance-based SA in attempt to make a step towards a generalization of the above mentioned frameworks. Our approach consists in decomposing the output time-series into a sum of an instantaneous component, a lag component, and a residual component; the purpose of the first two components is to quantify and clarify the roles of the memory effects of the inputs on the output. The components are constructed in a specific order to ensure an orthogonality property and avoid redundancy between variables in case of dependency. To do so, we apply a Gram–Schmidt process to make the variables orthogonal to the already captured effects. Although this construction holds under weak assumptions (for example, it does not require independence), in this work, we restrict ourselves to stationary processes for two main reasons: first, the ability to use sample covariance and variance estimators, and, second, the constant variance under this assumption reflects in a global fashion the behaviour of the resulting components. We evaluate the ability of the proposed method to approximate the true quantities on a toy example and, then, show its performance on a synthetic controlled case and real-world cases where the stationarity and the independence assumptions do not necessarily apply.

Our contributions are summarized as follows:

- we propose a two-stage approach to quantify memory effects;
- we distinguish the instantaneous effects from the lag effects and attribute to each one a component to reflect the captured dynamic;
- based on the modelled components, we derive simple variance-based indices to measure history-aware variables importance and provide useful post-hoc global explanations.

2. Proposed approach

We consider the scenario where the response variable Y is a function indexed by the time t with values in $\mathcal{Y} \subset \mathbb{R}$. This function is assumed to be the output of a function f depending on the history of $D \geq 1$ observed variables gathered in $X = (X^1, \ldots, X^D)$ such that:

$$\forall t \in \mathbb{N},\ Y_t = f\left((X_s)_{s \leq t}, \varepsilon_t\right), \tag{2.1}$$

where ε_t is an error component (to model observation errors or the fact that Y may also depend on additional but not observed variables for instance). The processes $X^1, \ldots, X^D$ can be mutually dependent. In this work, we make the assumption of stationarity. In particular, under this assumption, the means and the variances of the processes remain constant, and the covariance function does not depend on the time origin. Without loss of generality, the means are assumed to be zero. In practice, the underlying model f is unknown and we are only provided with a sample of $N > 1$ input-output pairs, noted $\mathcal{D} = \{(x_0, y_0), \ldots, (x_{N-1}, y_{N-1})\}$ so that for each $0 \leq t \leq N-1$, $(x_t, y_t) \in \mathbb{R}^D \times \mathcal{Y}$ is the available observation of (X_t, Y_t). This set is also assumed regularly-sampled.

Given the output Y_t, we seek to construct a decomposition of the form:

$$Y_t = Y_t^{\text{inst}} \oplus Y_t^{\text{lag}} \oplus Y_t^{\text{res}}, \tag{2.2}$$

where

- Y_t^{inst} is the instantaneous component, ie. a function of the instantaneous input $X_t = (X_t^1, \ldots, X_t^D)$,
- Y_t^{lag} is the linear memory component, ie. a linear function of lagged variables $X_{t-1:t-K} = (X^1_{t-1:t-K}, \ldots, X^D_{t-1:t-K})$ where for each $1 \leq d \leq D$ the notation $X^d_{t-1:t-K}$ refers to the vector gathering the observations of X^d between $t-K$ and $t-1$ in decreasing order of time, and K is a positive integer to control the lag window and is to be selected based on a cross-validation method described in Section 3.1.
- Y_t^{res} is a residual that takes into account non-linear memory effects, unobserved variables, noise, etc,
- $\oplus$ denotes an orthogonal sum when considering covariance as inner product.

In case of independent inputs, the Hoeffding-Sobol [7, 16] decomposition can be applied to decompose the instantaneous component. Efforts have been made to propose generalizations of this decomposition to the case of dependent variables [17, 18] or others indices based on tools from game theory such as the Shapley values [19] or Proportional values (PV). The indices inspired from PV are called Proportional Marginal Effects (PME) [20] and allocate to each variable its proportional share of the output's variance while Shapley-based tools rely on an egalitarian allocation. We refer to [20] for further details; we show in Section 4.3 the results of PME in real world cases to provide global explanations of the instantaneous components.

To attribute to each variable the memory effects it carries, the memory term Y_t^{lag} is decomposed as a sum of D orthogonal components:

$$Y_t^{\text{lag}} = Y_t^{\text{lag},1} \oplus \cdots \oplus Y_t^{\text{lag},D}, \tag{2.3}$$

such that the component $Y_t^{\text{lag},d}$ quantifies the effects of the d-th variable that cannot be captured by the other components.

To achieve this second decomposition, we propose to proceed iteratively to ensure the orthogonality property, by constructing the instantaneous component first, then, transforming suitably the lagged variables to eliminate the effects already quantified.

2.1. Variance-based time series explainability

The orthogonality obtained by construction ensures that

$$\mathrm{Var}(Y_t) = \mathrm{Var}(Y_t^{\mathrm{inst}}) + \mathrm{Var}(Y_t^{\mathrm{lag}}) + \mathrm{Var}(Y_t^{\mathrm{res}}). \tag{2.4}$$

As the focus of this work is on separating the instantaneous and lag contributions, we define a total instantaneous index as:

$$S^{\mathrm{inst}} := \frac{\mathrm{Var}(Y_t^{\mathrm{inst}})}{\mathrm{Var}(Y_t)}, \tag{2.5}$$

and a total lag index as:

$$S^{\mathrm{lag}} := \frac{\mathrm{Var}(Y_t^{\mathrm{lag}})}{\mathrm{Var}(Y_t)}. \tag{2.6}$$

We also define for each $d \in \{1, \dots, D\}$ the individual lag index of variable $X^d_{t-1:t-K}$ as:

$$S_d^{\mathrm{lag}} := \frac{\mathrm{Var}(Y_t^{\mathrm{lag},d})}{\mathrm{Var}(Y_t^{\mathrm{lag}})}. \tag{2.7}$$

The indices defined above benefit from different properties. They are constant in case of stationarity, positive thanks to the positivity of the variance, and the individual lag indices sum to one. Moreover, since the lag components are obtained iteratively, the lag indices should be interpreted according to the order of variables. For example, the index S_d^{lag} represents the part of variance explained by (or the effects of) the variable d without the effects of the same variable already taken into account in S^{inst} and S_i^{lag} for $i \leq d-1$. Finally, if furthermore a decomposition of the instantaneous variance is provided (based on PME for instance) and individual instantaneous indices are available, the latter can be grouped with the individual lag indices into holistic indices that reflect, regardless of their temporal nature, the effects of the inputs on the output as the traditional sensitivity indices.

2.2. Memory effects quantification

In this section, we describe the construction methodology of the decomposition of equation (2.2) which we recall below:

$$Y_t = Y_t^{\mathrm{inst}} \oplus Y_t^{\mathrm{lag}} \oplus Y_t^{\mathrm{res}}. \tag{2.8}$$

First, the instantaneous component Y^{inst} is modelled as a polynomial function of the instantaneous input X_t (see Sect. 2.2.1). Then, the lag component Y^{lag} is decomposed as a sum of D lag components $Y^{\mathrm{lag},\,1}, \dots, Y^{\mathrm{lag},\,\mathrm{D}}$, constructed iteratively to guarantee orthogonality, such that each component $Y^{\mathrm{lag},\,\mathrm{d}}$ is a linear function of the associated variable X^d. Finally, the residual component Y^{res} is obtained by subtracting the former components Y^{inst} and Y^{lag} from the true output Y. Learning the dynamic of the residual component is out of the scope of this work.

This modelling choice is made based on a performance-complexity trade-off while also allowing for a straightforward analysis and easily interpretable components. The proposed framework is flexible enough so that the polynomial model can be replaced by a complex model such as a Gaussian Process Regression or a neural network at the cost of a higher complexity. The lag components can also be replaced by models tailored for time-series regression (recurrent neural networks for instance) to take notably into account non-linear effects.

However, although more complex models may allow for a better representativity, they suffer from a lack of interpretability, which is a major drawback for the purpose of sensitivity analysis.

2.2.1. Instantaneous component

The aim of the instantaneous component Y_t^{inst} is to capture the information carried by the instantaneous variables that are in X_t. A large variety of models, from linear models to complex neural networks, can be used to approach the output Y_t depending on the complexity of the data. In this work, we impose a polynomial form to Y_t^{inst}:

$$Y_t^{\text{inst}} = \sum_{p}^{P_r} \beta_p \psi^p(X_t), \tag{2.9}$$

where $\beta := (\beta_1, \ldots, \beta_{P_r})$ is the model's parameter, $\psi(X_t) := (\psi^1(X_t), \ldots, \psi^{P_r}(X_t))$ is the vector gathering polynomial functions of the instantaneous variables, where the total order is less than or equal to a specified degree r, and $P_r = \binom{D+r}{D} - 1$ is the number of polynomial terms. One can also consider as sparse analog, based on hyperbolic index sets for instance, as suggested in [21].

To enhance the interpretability of this component for further analysis, we choose orthogonal polynomials: if we denote $m(X_t) := (m^1(X_t), \ldots, m^{P_r}(X_t))$ a vector of monomials, and L the triangular matrix resulting from the Cholesky decomposition of the covariance matrix of $m(X_t)$, we have that $\psi(X_t)$ is the solution of:

$$L\psi(X_t) = m(X_t). \tag{2.10}$$

This approach is equivalent to Polynomial Chaos expansion [11] in case of independent inputs.

2.2.2. Lag component

In real-world cases, the effects of an explanatory variable on the measured output may be spread over time or observed with a delay. A class of models that enables to model variables influence over time on a system's output is represented by distributed lag models [22]. The choice of this particular class implies a response time between the inputs in the present and their effects on the outputs in the future.

The lag components are chosen as linear distributed lag models given by:

$$\forall d \in \{1, \ldots, D\},\ Y_t^{\text{lag},d} = \sum_{s=1}^{K} H_s^d \tilde{X}_{t-s}^{d,t}, \tag{2.11}$$

where $H^d = (H_1^d, \ldots, H_K^d)$ is a vector parameter called lag weights or filter, K is a positive integer to control the lag window and for each $d \in \{1, \ldots, D\}$, $\tilde{X}_{t-s}^{d,t}$ is a linear transformation of X_{t-s}^d to ensure the orthogonality property. In fact, lag effects cannot be extracted based on raw variables since this will induce redundancy and will not result in the sought orthogonality. To this end, we transform the variables $(X^1, \ldots, X^D)$ into $(\tilde{X}^1, \ldots, \tilde{X}^D)$ iteratively: each input variable is made orthogonal to the instantaneous component and to the already fitted lag components using the Gram–Schmidt procedure that follows the sequence:

$$\begin{aligned} d \in \{1, \ldots, D\},\ \tilde{X}_{t-s}^{d,t} := X_{t-s}^d & \\ & - \frac{1}{\text{Var}(Y_t^{\text{inst}})} \text{Cov}\left(X_{t-s}^d, Y_t^{\text{inst}}\right) Y_t^{\text{inst}} \\ & - \sum_{i=1}^{d-1} \frac{1}{\text{Var}(Y_t^{\text{lag},i})} \text{Cov}\left(X_{t-s}^d, Y_t^{\text{lag},i}\right) Y_t^{\text{lag},i}. \end{aligned} \tag{2.12}$$

Note that the variables thus transformed are characterized by both the present time t *via* Y_t^{inst} and the lags $t-s$ *via* the lagged variables $X_{t-s}^{d,t}$ for $s = 1, \dots, K$.

Although different variants of distributed lag models exist to take into account non-linearities, we restrict ourselves in this work to only quantify linear history effects to obtain easily interpretable lag components with a reasonable complexity. Non-linearities are expected to be partially taken into account *via* the instantaneous component thanks to temporal correlation. We conduct in Section 2.2 a theoretical analysis on a linear case to illustrate to which extent temporal correlation (or memory) allows to capture a part of history.

Finally, such strategy inevitably relies on the order between variables. We propose then to sort them based on the variances of the resulting memory components in a decreasing order, ie. in a way such that $\mathrm{Var}(Y_t^{\mathrm{lag},1}) \geq \cdots \geq \mathrm{Var}(Y_t^{\mathrm{lag},D})$, to impose a hierarchy between variables and reflect their importance.

This order can be obtained by computing at each iteration of the Gram–Schmidt process the variances of the lag components of all the remaining variables and the variable associated to the lag component with the highest variance is fixed so that the process results in an appropriate ordering of variables. In total, $\sum_{i=1}^{D} D - i + 1 = \frac{D(D+1)}{2}$ possible lag components are computed.

2.2.3. Residual component

For a given input, if the true output is known (for example if we have access to the true model f, which we remind that we assume in this work that we do not have access to) it is straightforward to obtain Y_t^{res} by subtracting Y_t^{inst} and Y_t^{lag} from Y_t. Otherwise, the residual component can be modeled too during training. Since quantifying it does not allow further analysis in our case, we restrict ourselves to obtain its variance using train data which is a sufficient indicator whether the former components approximate the true output with accuracy.

3. Estimation and Theoretical analysis

3.1. Practical estimation

The first step of the method is to compute the orthogonal polynomials gathered in $\psi(X_t)$, which can be obtained by a Cholesky decomposition of the covariance matrix of monomials gathered in $m(X_t)$ as explained in Section 2.2. In practice, given the sample $\mathcal{D}$, we compute the monomials for every input and perform a QR decomposition of the resulting matrix. Proceeding this way allows a better numerical stability and avoids solving the system (2.10). The complexity of this step is $\mathcal{O}(NP_r^2)$.

In case of independent variables, using the former normalization of $\psi(X_t)$, the parameter β can be obtained immediately by projection:

$$\beta = \mathrm{Cov}\left(\psi(X_t), Y_t\right). \tag{3.1}$$

For versatility, β is obtained by ordinary least squares:

$$\beta \approx \hat{\beta} \in \arg\min_{b \in \mathbb{R}^{P_r}} \sum_{t=K}^{N-1} (y_t - \psi(x_t)^\top b)^2, \tag{3.2}$$

which has a complexity of $\mathcal{O}(P_r^2(N-K) + P_r^3)$ when using the closed-form solution.

Analogically to the instantaneous parameter estimation, if the input variables are assumed independent, the filters can be obtained by projection:

$$\mathrm{Var}(\tilde{X}_{t-1:t-K}^{d,t}) H^d = \mathrm{Cov}\left(\tilde{X}_{t-1:t-K}^{d,t}, Y_t\right). \tag{3.3}$$

However, only estimators of the covariance and the variance are available through the sample $\mathcal{D}$ and the independence assumption does not always hold. Hence, assuming that the input time-series are second-order stationary, we replace the covariances and variances with sample covariance and sample variance to compute the transformed data following the sequence (2.12). We refer to Appendix B for the general formula of such estimators.

For versatility, the filters are estimated iteratively by a penalized least squares to handle notably the case of dependent inputs:

$$H^d \approx \hat{H}^d \in \arg\min_{h\in\mathbb{R}^K} \sum_{t=K}^{N-1} (y_t - \tilde{x}^{d\top}_{t-1:t-K} h)^2 + \lambda \|\nabla h\|_2^2 \tag{3.4}$$

where we remind that $\tilde{x}^d_{t-1:t-K} = (\tilde{x}^d_{t-1}, \ldots, \tilde{x}^d_{t-K})$ and ∇h is the discrete gradient of h defined by:

$$\nabla h = (h_2 - h_1, \ldots, h_K - h_{K-1})\,. \tag{3.5}$$

The objective of this penalization is to avoid ill-posed problems and promote the smoothness of the filters obtained. Similarly as above, the closed-form solution has complexity $\mathcal{O}(K^2N + K^3)$ or $\mathcal{O}(K^2N)$ since $N >> K$.

The hierarchy between lag components is obtained by brute force by estimating and comparing all the variances at each iteration of the procedure which results in a total of $\frac{D(D+1)}{2}$ lag variances to compute.

Finally, we list the hyper-parameters the method relies on:

- the lag window K (responsible for the memory of the filters),
- the maximum degree r of the instantaneous component (reflects the complexity of the instantaneous model),
- the penalization parameter λ to select smooth filters in case of ill-posed problem.

For their selection, we perform an expanding window cross-validation[1] where the train set is expanding at each step and the size of the validation set is fixed. We could also choose a hyperbolic scheme for truncating the polynomial expansion of the instantaneous component (*i.e.* a truncation strategy based on q-norms with $0 < q < 1$) [21]. As for the lag window K, the best value that can be selected achieves a trade-off between temporal correlation and the lag-window of the filters. If the correlation time of the inputs is large compared to the lag-window of the filters, the instantaneous component will capture history through correlation and thus a large K will not be needed to compensate this. Conversely, if the filter's history is large compared to the correlation time, the instantaneous component will hardly capture history and a large value K needs to be set to allow the lag components to capture all the information carried by the filters. The following section illustrates the compensation phenomena that can occur between the instantaneous and the lag component in a linear theoretical framework.

3.2. Theoretical analysis: Instantaneous versus Lag compensation

For a better understanding of the proposed methodology and a better interpretation of the proposed decomposition, we conduct a theoretical analysis on the linear case, namely, the case where the true output Y_t can be written as:

$$Y_t = \sum_{d=1}^{D} \beta^*_d X^d_t + \sum_{d=1}^{D} H^{*,d^\top} X^d_{t-1:t-K} \tag{3.6}$$

and where:

[1] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html

- the first term carries only instantaneous effects;
- the second term only carries history effects;
- β^* is a scalar and $H^{*,\mathrm{d}} = (H_1^{*,\mathrm{d}}, \dots, H_K^{*,\mathrm{d}})$ is the vector gathering the lag weights associated to the lag variables $X^d_{t-1:t-K} = (X^d_{t-1}, \dots, X^d_{t-K})$.
- the input time-series are Gaussian processes.

In such a setting, linearity implies that the output is also a Gaussian process. The Gaussian assumption allows to conduct a theoretical reasoning regardless of the dependence nature of input variables. However, for the sake of simplicity, we consider inputs that are independent auto-regressive (AR) models of order 1 for their suitable statistical properties[2]:

$$\forall d \in [|1, D|],\ X_t^d = \rho_d X_{t-1}^d + \varepsilon_t^d \text{ where } \left(\varepsilon_t^d\right)_t \text{ are drawn i.i.d from } \mathcal{N}(0, 1-\rho_d^2).$$

Under these assumptions, the instantaneous component Y_{inst} defined by equation (2.9) with a degree $r = 1$ is precisely the conditional expectation of Y_t knowing $\{X_t^1, \dots, X_t^D\}$ which can be obtained using the properties of Gaussian conditioning as:

$$Y_t^{\mathrm{inst}} = \mathbb{E}\left[Y_t | X_t^1, \dots, X_t^D\right] = \beta^\top X_t, \tag{3.7}$$

where β is the solution to (3.1) and its closed-form expression is:

$$\beta = \left(\beta_d^* + \sum_{s=1}^{K} H_s^{*,d}\, \rho_d^s\right)_{d=1}^{D} \tag{3.8}$$

This result highlights the fact that long memory effects can be taken into account *via* the temporal correlation of the variables. Thus, the modelled lag effects are the remaining effects that are not modelled by the instantaneous variable. For instance, if the window memory of the filters is significantly large compared to the correlation time[3] τ of the input processes, the instantaneous component can hardly capture the information of the filters. In the opposite case where the correlation time is large, the instantaneous component will dominate.

Similarly, we have that the first lag component, defined by (2.11) and (2.12), is the conditional expectation of Y_t knowing $\tilde{X}^{1,t}_{t-1:t-K}$ which is obtained, again, by Gaussian conditioning as:

$$Y_t^{\mathrm{lag},1} = \mathbb{E}\left[Y_t | \tilde{X}_{t-1}^1, \dots, \tilde{X}_{t-K}^1\right] = {H^1}^\top \tilde{X}^{1,t}_{t-1:t-K}, \tag{3.9}$$

such that:

$$\tilde{X}^{1,t}_{t-s} := X^1_{t-s} - \rho_1^s \frac{\beta_1}{\sum_{d=1}^D \beta_d^2} Y_t^{\mathrm{inst}} \text{ and } H^1 = \tilde{\Sigma}_1^{-1} C_1, \tag{3.10}$$

where:

$$C_1 = \mathrm{Cov}\left(\tilde{X}^{1,t}_{t-1:t-K}, Y_t\right) = \left(\sum_{l=1}^{K} H_l^{*,1}\, (\rho_1^{|l-s|} - \rho_1^{l+s})\right)_{s=1}^{K}, \tag{3.11}$$

[2]In particular, in our setting, the mean is equal to zero, the variance is equal to one, and the covariance between t and $t+s$ is of the form ρ^s.

[3]The correlation time τ is defined as the time step at which the autocorrelation function drops below $1/e$ and reflects the memory of the process. For an AR(1) process of parameter ρ, we have $\tau = -1/\log(\rho)$.

and

$$\tilde{\Sigma}_1 = \mathrm{Var}(\tilde{X}^{1,t}_{t-1:t-K}) = \left[\rho_1^{|i-j|} - \frac{\beta_1^2}{\sum_{k=1}^d \beta_k^2}\rho_1^{i+j}\right]_{1\leq i,j\leq K}. \tag{3.12}$$

In the univariate case, observe that C_1 is the product of $\tilde{\Sigma}_1$ and $H^{*,1} = (H_1^{*,1}, \ldots, H_K^{*,1})$, and thus the lagged model reduces to:

$$Y_t^{\mathrm{lag},1} = {H^{*,1}}^\top \tilde{X}^{1,t}_{t-1:t-K}. \tag{3.13}$$

In the multivariate case, by taking $|\rho| \longrightarrow 0$ for short memory processes or $|\rho| \longrightarrow 1$ for long memory processes, expanding the terms depending of ρ to the order 2 allows to observe other compensation phenomena. We refer to Section 4.1 for an illustration and numerical results in a such linear setting. Similar compensation phenomena can be observed in a non-linear setting; we refer to Section 4.2 for a numerical illustration.

4. Experiments

We illustrate the properties of the method on two synthetic data sets and evaluate its performance on 3 real-world data sets of different sizes and numbers of variables. We recall that, to the best of our knowledge, no existing approaches in the literature address the problem of explaining outputs of the form $Y_t = f(X_s, s \leq t)$. Therefore, there are no established baselines or comparative benchmarks for a direct comparison.

For an effective application, the hyper-parameters need to be carefully selected. For the real data experiments, the hyper-parameters are selected in the following grid based on the cross-validation procedure mentioned above such that the train and validation sets are separated by a gap of 100 time steps:

- the lag window $K \in [10, 25, 50, 75, 100]$;
- the maximum degree $r \in [1, 2, 3, 4, 5]$;
- the penalization parameter $\lambda \in [0, 1, 10^2, 10^4, 10^6, 10^8]$, with no penalization if $\lambda = 0$.

Training times of the synthetic experiment are reported in Appendix A.2.

4.1. Synthetic linear setting

We consider a first toy data set which we refer to as the *linear case*. The assumptions made in Section 3.2 are fulfilled to study empirically the convergence of the filters' estimators to their true quantities based on the closed-form expressions of the section above. The inputs are AR(1) processes and the output Y_t is defined as:

$$Y_t := \sum_{d=1}^{D} \beta_d^* X_t^d + \sum_{d=1}^{D}\sum_{s=1}^{K} H_s^{\mathrm{lin},d} X_{t-s}^d \tag{4.1}$$

where $D = 3$ (*i.e.*, three variables), β^* is a fixed scalar and H^{lin} are the filters shown on Figure 1 and defined in Appendix A. We consider two scenarios: short correlation inputs and long correlation inputs where the auto-regressive parameters ρ_1, ρ_2, ρ_3 are set to obtain (respectively) correlation times ranging approximately around 3 time units in the short correlation case and around 20 time units in the long correlation case.

For each linear scenario, we apply the methodology on a subset of size N and vary N from 200 to 100 000. We save the estimated filters and record the RMSE (root mean squared error) of the estimators over 25 iterations. Since we are interested in assessing the convergence, the filters saved are the filters obtained at the first iteration after fitting the instantaneous model, ie. the filters of all variables are estimated as if the latter were the first ordered variable. Figures 2.(A) and 3.A show the estimated filters and their 95% confidence intervals for $N = 1000$, and Figures 2.B and 3.B show the estimated filters and their 95% confidence intervals for $N = 100\,000$.

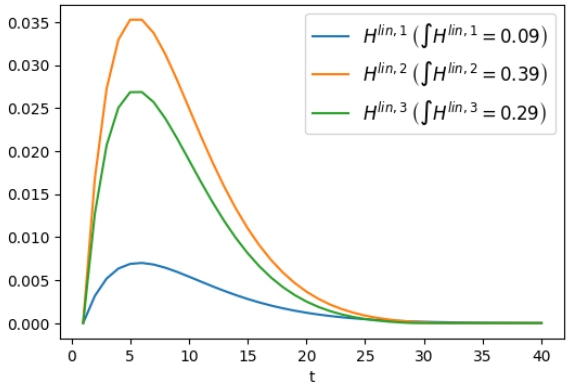


FIGURE 1. True Linear filters applied in the linear case and their integrals. The higher the integral, the greater the influence of the corresponding variable.

We observe that the estimated filters in the short correlation scenario are smoother and have reduced confidence intervals compared to the filters obtained in the long correlation setting. Moreover, Figure 2.C shows that in the first setting the convergence is quicker than in the setting of Figure 3.C. Finally, in virtue of the compensation analysis done in Section 3.2, Figure 2.D shows that the lag component is preponderant and control the global dynamic of the output while in the long correlation setting we observe on Figure 3.D that the instantaneous component tends to be in advance and that the lag component contributes in capturing the delay effects and correcting the former component.

As the inputs X^1, X^2 and X^3 have the same amplitude (their variance is equal to 1), looking at the filters informs us about the sensitivity hierarchy, in the sense that, the higher the filter's integral for a variable, once normalized by the corresponding correlation time, the more influence we expect that variable to have on the output. This intuition is confirmed by the individual lag indices shown on Figures 4a and 4b on which we see that the variable with the most influence on the lag component is X^2 followed by X^3 and then X^1 which is the same order implied by the integrals of linear filters shown on Figure 1. Even if the instantaneous component is more influential in the long correlation case than in the short correlation case, Figures 4a and 4b show that the individual lag indices report the same amount of lag importance in both settings.

4.2. Synthetic general setting

In this section, we consider a second data set that we call the *general case*. Compared to the linear case, the general case consists of a non-linear short memory component in addition to a linear memory component, which allows to observe the behaviour of the method facing non-linearity and to illustrate the similar compensation phenomena. We refer to Appendix A for details about this synthetic design.

The inputs are three AR(1) processes such that their auto-regressive parameters are set to obtain correlation times around 4 time units and non-linear effects up to order 3 of these variables are incorporated in the output. Since non-linearities are involved, a closed-form for the filters is not available; we record then the R2-score and the variances obtained to assess the convergence of the method in this case over 25 repetitions and a varying data set size N between 200 and 10 000. The aforementioned cross-validation is performed and the best lag window selected is 25.

Figure 5 shows the filters that are applied to the lag variables while Figure 7.A shows the estimated filters. We observe that the filters estimators recover the shape of the linear filters without being equal because of the compensation phenomena. We observe also on Figure 7.B that the instantaneous component is in advance compared to the lag component. Finally, Figures 7.C and 7.D illustrate the convergence of variances and the R2 scores. We observe in particular that we do not recover the total variance, nor an R2-score equal to 1 which

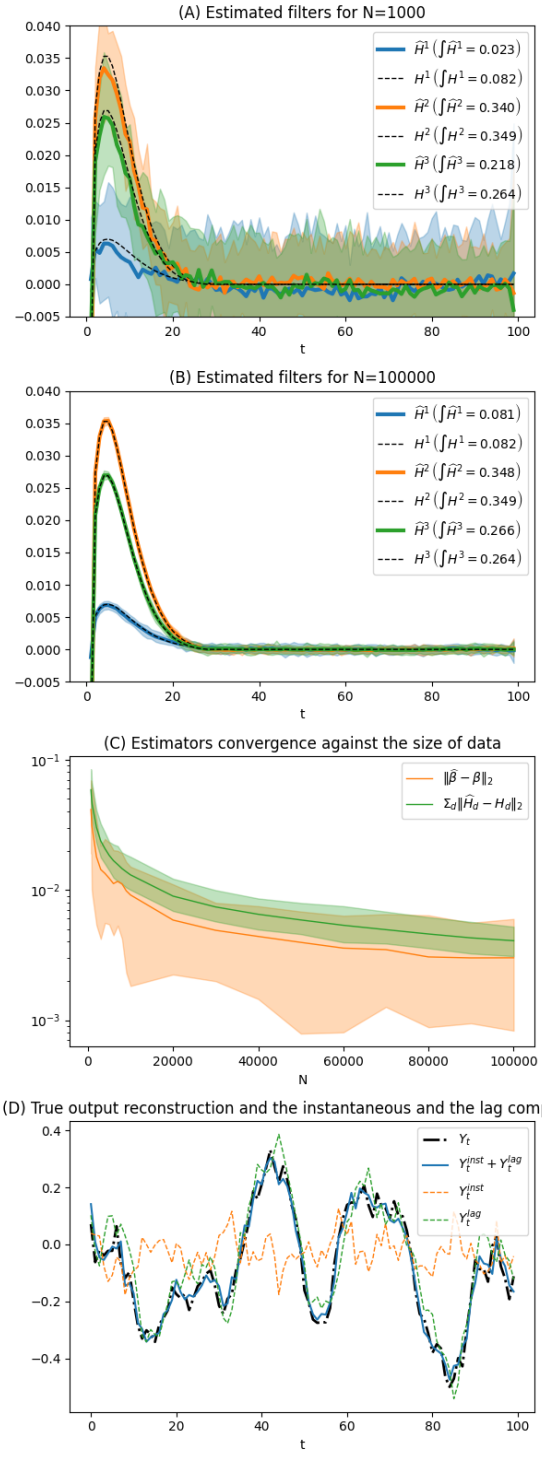


FIGURE 2. Synthetic experiment results in the short correlation setting.

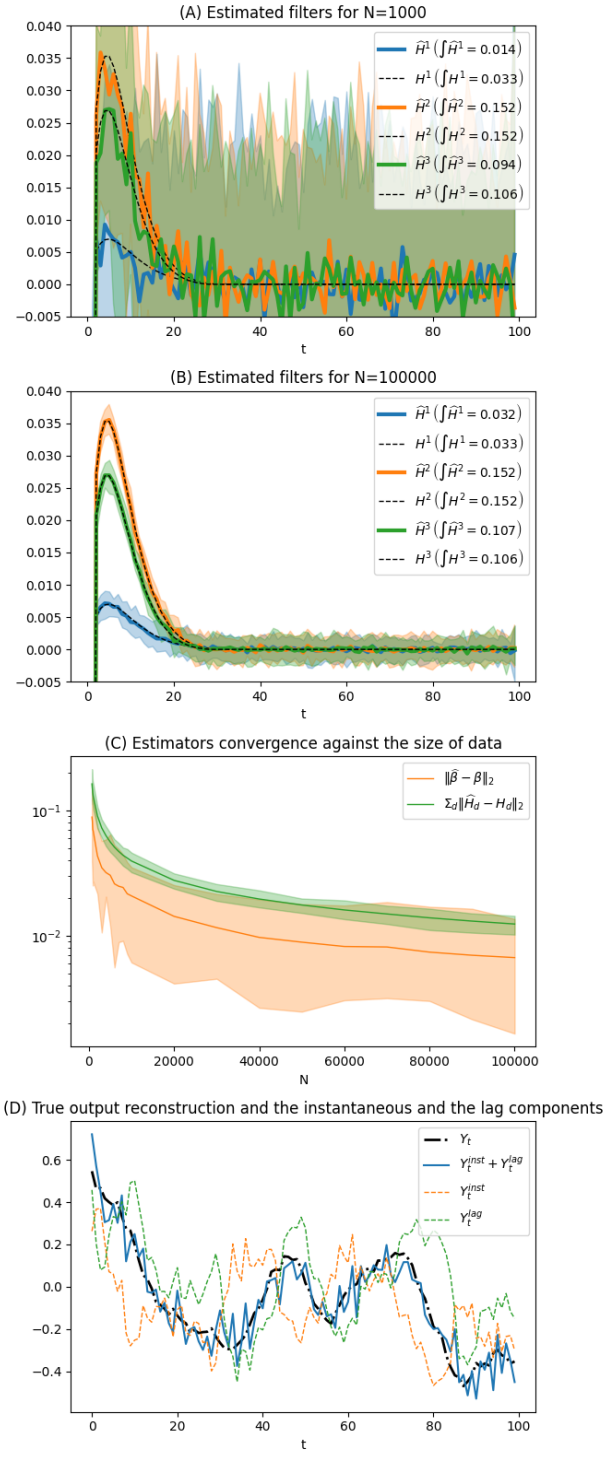
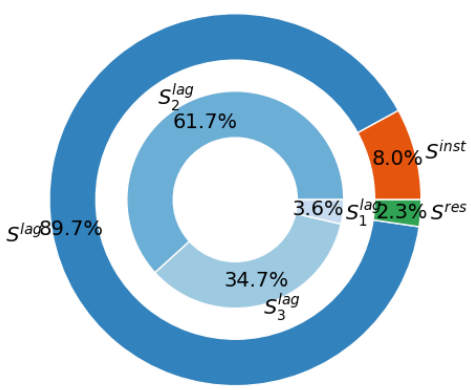
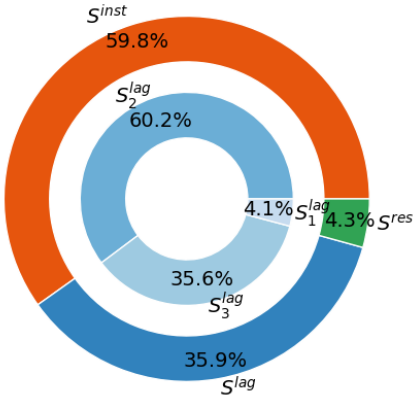


FIGURE 3. Synthetic experiment results in the long correlation setting.



(A) Short correlation setting



(B) Long correlation setting

FIGURE 4. Pie charts of indices: instantaneous, total lag and residual indices in the outer donut, individual lag indices in the inner donut.

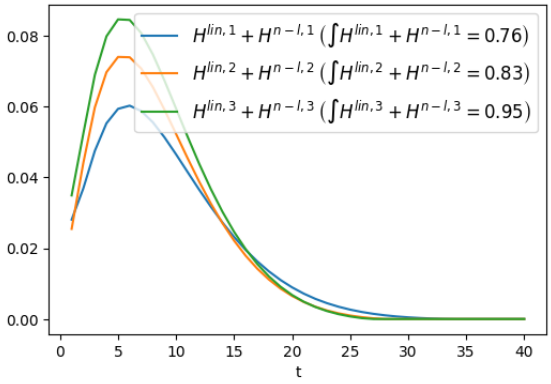


FIGURE 5. Filters that are applied to linear variables in the general setting.

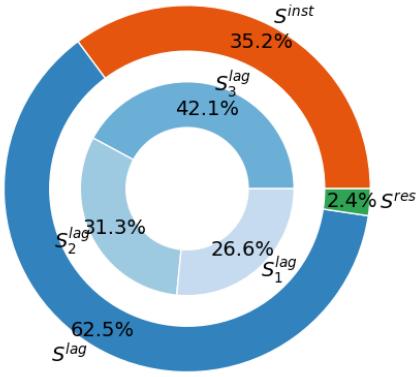


FIGURE 6. Donut chart of indices in the general setting.

is explained by the fact that: 1)method's parameters and filters are estimated through a least squares instead of a projection approach, and, 2) the best penalization parameter selected is not zero. Finally, the donut chart of Figure 6 shows that the order of individual lag indices is the same as the order of integrals of filters that are applied to the linear variables shown in Figure 5.

4.3. Real applications datasets

In this section, we apply the methodology to real-world data sets to detect the lag effects and quantify to how much extent they allow a better explanation of the output.

We consider three data sets:

- **AirQuality**[4] is a data set provided by UCI Machine Learning repository [23] that has been first introduced in [24]. It consists of hourly measured pollutants data in addition to meteorological data with the same time sampling for different stations across multiple sites in Beijing. In this work, we only focus on one station. Data was split into two sets of size 3000: a train set between March and September 2013 and a test set between March and September 2014. We select `O3` (Ozone concentration) as the target variable to explain

[4] https://archive.ics.uci.edu/dataset/501/beijing+multi+site+air+quality+data

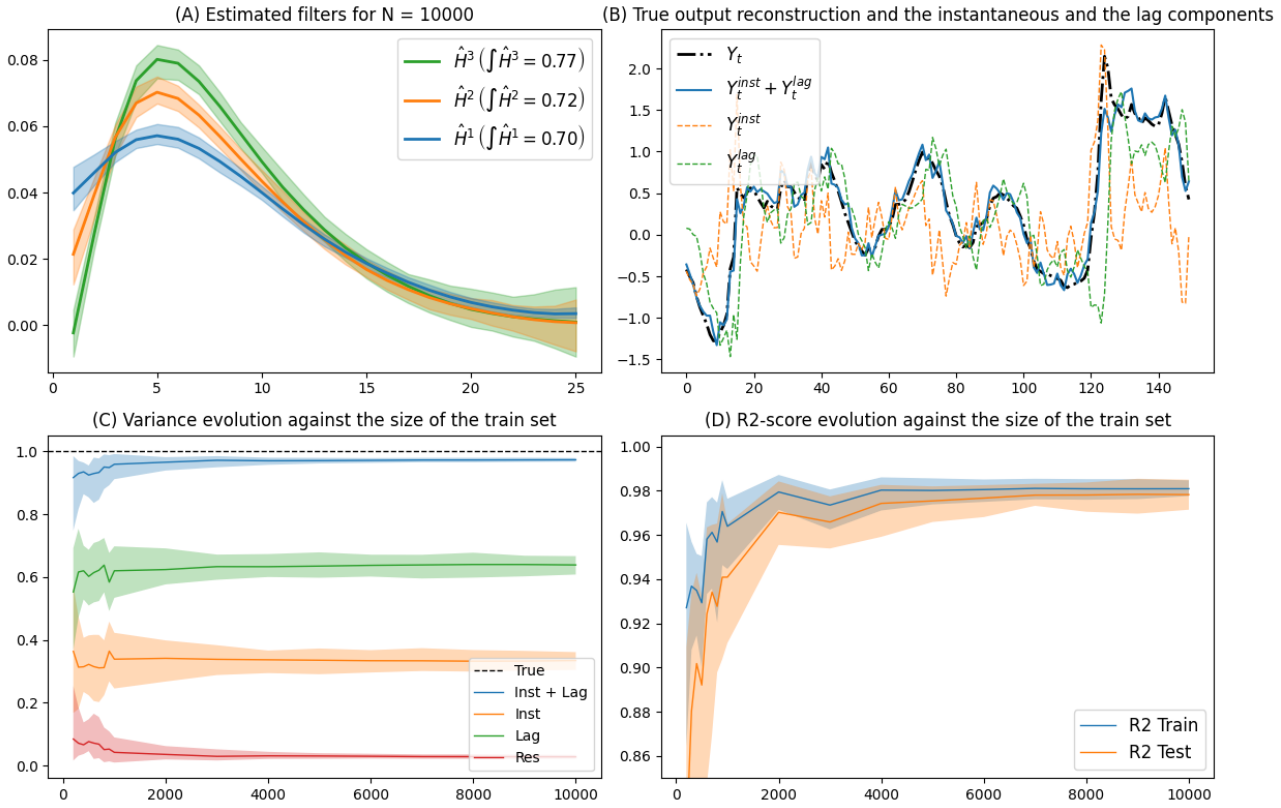


FIGURE 7. General synthetic experiment summary.

based on six variables: `NO2` (nitrogen dioxide concentration), `CO` (carbon monoxide), `TEMP` (temperature), `PRES` (pressure), `DEWP` (dew point temperature) and `WSPM` (wind speed).

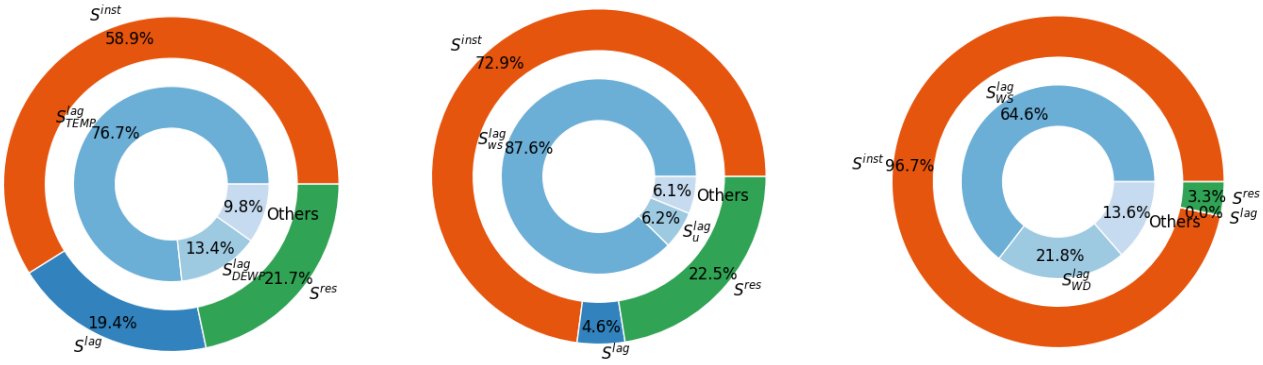
- **GEFCom2012**[5] is a data set introduced in [25] of hourly measured wind power production with weather forecasting variables for different wind farms. From one farm (chosen arbitrarily), we select data as follows: the first 4415 time steps are selected as the training set and the last 4415 as the test set. The target is the wind power production; we study the influence of three weather variables: `ws` (wind speed), `u` (the zonal wind component) and `v` (the meridional wind component).
- **WindPower** is a data set with a higher resolution than the other data sets (every 10 min). The train and test sets are of size 30 000. We focus in this work on quantifying the potential delay observed on the power production of a wind farm (target) taking into account the wind speed, its direction and the temperature. In other words, we ask ourselves to how much extent the wind starting at a time t is responsible for the rotation of the wind turbine (and thus the production) at the time $t + K$ where K can be seen as the duration the turbine takes to start or the delay induced by the inertia of the turbine. Data is available upon request and we refer to [26] for extensive details.

Since only a small subset of values is missing for the AirQuality and WindPower data sets, a linear interpolation is performed to impute them. Train and test sets are selected so that they have the same distribution and the stationary property is relatively fulfilled (the study of non-stationary time series being left for future work). Data are standardized and the best hyper-parameters are selected as explained in Section 3.1.

[5] http://blog.drhongtao.com/2016/07/gefcom2012-load-forecasting-data.html

TABLE 1. Data size, selected hyper-parameters, training times and performance metrics.

	D	N	K	r	λ	Training time (s)	R^2_{test}	R^2_{train}	R^2_{inst}	R^2_{lag}
AirQuality	6	3000	50	2	10^2	0.441	0.718	0.787	0.589	0.199
GEFCom	3	4415	75	3	10^4	0.332	0.735	0.772	0.729	0.051
WindPower	3	30 000	10	3	0	1.636	0.951	0.967	0.967	0.0002



(A) AirQuality (B) GEFCom (C) WindPower

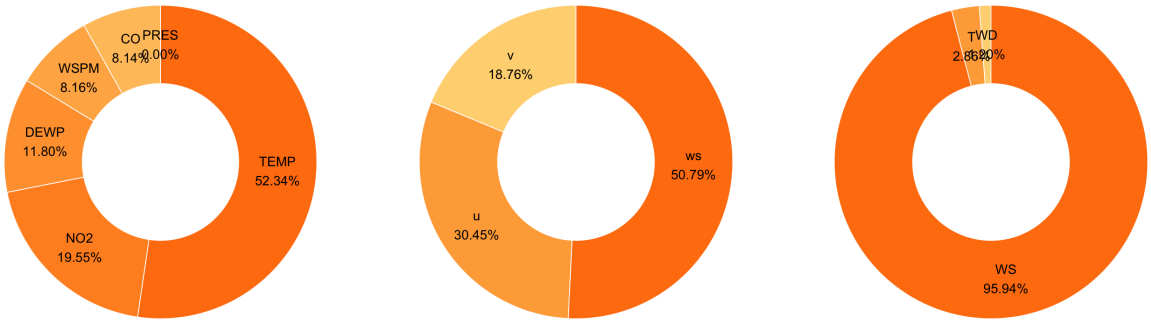
FIGURE 8. Pie charts of indices: instantaneous, total lag and residual indices in the outer donut, individual lag indices in the inner donut.

We use the R2-score to assess the validity of the resulting model and report the recorded scores in Table 1 on which we read that the model produces good results in reconstructing the output and to predict new data. Moreover, training times remain reasonable with respect to data size. We compute for each dataset the total indices and the individual lag indices defined in Section 2.1 and visualize them in the pie charts of Figures 8a, 8b and 8c that shows the instantaneous effects are responsible for a substantial part in explaining the output time-series in the three application cases; the lag effects have an influence on the output in the AirQuality and GEFCom cases and no influence for WindPower data. These results are consistent with the R2-scores of Table 1. We notice in particular that the lag component helps improving the results obtained by the instantaneous component of around 34% gain from R^2_{inst} to R^2_{train} for AirQuality data for example and in a lesser extent on GEFCom data. For WindPower data however, taking into account the lag effects does not improve (nor deteriorate) the performance of the instantaneous model since the latter captures all the required effects to model the target variable. Figure 8a shows that the variables with the most effects on the ozone production history-wise are the temperature (`TEMP`) and the dew point temperature (`DEWP`) with effects due up to 76.7% and 13.4% respectively. These two variables are known to be related for instance to solar radiation and humidity which seem to have an effect on the ozone production[6]. As for the instantaneous effects, PME results on Figure 9a show `TEMP` and `NO2` as the most important variables which can help ground-level ozone formation. For the wind power production cases, wind speed has a large impact on both the instantaneous component and the lag one as shown in Figures 8b, 8c and Figures 9b and 9c.

5. CONCLUSION AND FUTURE WORK

This work proposes a novel methodology for history-aware sensitivity analysis based on a decomposition of the output time series into three non-correlated components: an instantaneous component, a lag component, and

[6]See for e.g. https://www.epa.gov/air-trends/trends-ozone-adjusted-weather-conditions



(A) AirQuality (B) GEFCom (C) WindPower

FIGURE 9. Donut charts of PME applied to the instantaneous component.

a residual component. This approach clarifies the roles of memory effects of the input variables. We demonstrate its effectiveness and its ability to provide plausible explanations in both synthetic and real cases in a reasonable amount of time.

Future work will focus on enhancing the interpretability of the instantaneous component and extending the framework to other models and quantities of interest (such as failure probabilities), handling non-stationary (for instance, handling the dependency to the seasons) and treating cases with irregularly-sampled time series (in case of missing data for *e.g.*, *etc.*).

ACKNOWLEDGMENTS

Wind power data is provided by Energy4Climate Interdisciplinary Center (E4C) of Institut Polytechnique de Paris, for which the authors would like to express their thanks.

FUNDING

This work was supported by the Chair Stress Test, RISK Management and Financial Steering, led by Ecole Polytechnique, its Foundation and sponsored by BNP Paribas.

CONFLICTS OF INTEREST

The authors have no competing interests to declare that are relevant to the content of this article.

DATA AVAILABILITY STATEMENT

The source code of the experiments is available online in a GitHub repository: https://github.com/ymouad/Towards-History-aware-Sensitivity-Analysis-For-Time-Series [27].

REFERENCES

[1] G. Perrin, Adaptive calibration of a computer code with time-series output. *Reliabil. Eng. Syst. Saf.* **196** (2020) 106728.

[2] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson and S. Aigrain, Gaussian processes for time-series modelling. *Philos. Trans. Roy. Soc. A: Math. Phys. Eng. Sci.* **371** (1984) 20110550.

[3] P. Lara-Benítez, M. Carranza-García and J.C.R. Santos, An experimental review on deep learning architectures for time series forecasting. *Int. J. Neural Syst.* (2020) 2130001.

[4] A. Alexanderian, P.A. Gremaud and R.C. Smith, Variance-based sensitivity analysis for time-dependent processes. *Reliabil. Eng. Syst. Saf.* **196** (2020) 106722.

[5] A. Saltelli, Sensitivity analysis for importance assessment. *Risk Anal.* **22** (2002) 579–590.

[6] K. Chan, A. Saltelli and S. Tarantola, Sensitivity analysis of model output: Variance-based methods make the difference, in *Winter Simulation Conference Proceedings* (1997) 261–268.

[7] I.M. Sobol, Sensitivity estimates for nonlinear mathematical models (1993) https://api.semanticscholar.org/CorpusID:115460399
[8] K. Campbell, M.D. McKay and B.J. Williams, Sensitivity analysis when model outputs are functions. *Reliabil. Eng. Syst. Saf.* **91** (2006) 1468–1472.
[9] M. Lamboni, H. Monod and D. Makowski, Multivariate sensitivity analysis to measure global contribution of input factors in dynamic models. *Reliabil. Eng. Syst. Saf.* **96** (2011) 450–459.
[10] F. Gamboa, A. Janon, T. Klein and A. Lagnoux, Sensitivity analysis for multidimensional and functional outputs. *Electron. J. Statist.* **8** (2014) 575–603.
[11] R. Ghanem and P. Spanos, Stochastic Finite Elements: A Spectral Approach. Civil, Mechanical and Other Engineering Series. Dover Publications (2003).
[12] S. Song, Z. Bai, H. Wei and Y. Xiao, Copula-based methods for global sensitivity analysis with correlated random variables and stochastic processes under incomplete probability information. *Aerospace Sci. Technol.* **129** (2022) 107811.
[13] R. Nelsen, An Introduction to Copulas. Springer Series in Statistics. Springer New York (2007).
[14] G. Chastaing, C. Prieur and F. Gamboa, Generalized Sobol sensitivity indices for dependent variables: numerical methods (2014) https://arxiv.org/abs/1303.4372.
[15] Z. Liu and Y. Choe, Data-driven sensitivity indices for models with dependent inputs using the polynomial chaos expansion. *Struct. Saf.* **88** (2018) 101984.
[16] W. Hoeffding, A class of statistics with asymptotically normal distribution. *Ann. Math. Statist.* **19** (1948) 293–325.
[17] G. Chastaing, F. Gamboa and C. Prieur, Generalized Hoeffding–Sobol decomposition for dependent variables – application to sensitivity analysis. *Electron. J. Statist.* **6** (2011) 2420–2448.
[18] G. Li, H. Rabitz, P. Yelvington, O. Oluwole, F. Bacon, C. Kolb and J. Schoendorf, Global sensitivity analysis for systems with independent and/or correlated inputs. *J. Phys. Chem. A* **114** (2010) 6022–6032.
[19] A.B. Owen and C. Prieur, On Shapley value for measuring importance of dependent inputs. *SIAM/ASA J. Uncertainty Quantif.* **5** (2017) 986–1002.
[20] M. Herin, M. Il Idrissi, V. Chabridon and B. Iooss, Proportional marginal effects for global sensitivity analysis. *SIAM/ASA J. Uncertainty Quantif.* **12** (2024) 667–692.
[21] G. Blatman and B. Sudret, Adaptive sparse polynomial chaos expansion based on least angle regression. *J. Computat. Phys.* **230** (2011) 2345–2367.
[22] A. Gasparrini, B. Armstrong and M.G. Kenward, Distributed lag non-linear models. *Statist. Med.* **29** (2010) 2224–2234.
[23] M. Kelly, R. Longjohn and K. Nottingham, The UCI machine learning repository. http://archive.ics.uci.edu.
[24] S. Zhang, B. Guo, A. Dong, J. He, Z. Xu and S.X. Chen, Cautionary tales on air-quality improvement in Beijing. *Proc. Roy. Soc. A: Math. Phys. Eng. Sci.* **473** (2017) 20170457.
[25] T. Hong, P. Pinson and S. Fan, Global energy forecasting competition 2012. *Int. J. Forecast.* **30** (2014) 357–363.
[26] D. Bouche, R. Flamary, F. d'Alché Buc, R. Plougonven, M. Clausel, J. Badosa and P. Drobinski, Wind power predictions from nowcasts to 4-hour forecasts: a learning approach with variable selection. *Renewable Energy* **211** (2023) 938–947.
[27] M. Yachouti, G. Perrin and J. Garnier, Python code for "Towards history-aware sensitivity analysis for time series". https://github.com/ymouad/Towards-History-aware-Sensitivity-Analysis-For-Time-Series.

Appendix A. Details on the synthetic experiment

A.1 Synthetic dataset design

The definition of Y_t is:

$$Y_t := \sum_{d=1}^{D} \beta_d^* X_t^d + \sum_{d=1}^{D}\sum_{s=1}^{K} H_s^{\text{lin},d} X_{t-s}^d \text{ in the linear case,} \tag{A.1}$$

$$Y_t := \sum_{p=1}^{P_r}\sum_{s=0}^{5} H_s^{\text{n-l},p}\, m^p(X_{t-s}) + \sum_{d=1}^{D}\sum_{s=1}^{K} H_s^{\text{lin},d} X_{t-s}^d \text{ in the general case} \tag{A.2}$$

where β^* is a scalar. The chosen form of the linear filters is:

$$H_t^{\text{lin},d} := \frac{1}{\omega_d\sqrt{1-\zeta_d^2}} C_{\gamma_{d,1}} \exp\left(-650\,\zeta_d\,\omega_d t\right) \sin\left(\frac{650}{\omega_d\sqrt{1-\zeta_d^2}}\, t\right) \mathbf{1}_{t\in[|1,K|]}, \tag{A.3}$$

and the non-linear filters are:

$$H_t^{\text{n-l},d} := \frac{1}{\omega_d\sqrt{1-\zeta_d^2}} C_{\gamma_{d,2}} \exp\left(-t\right) \mathbf{1}_{t\in[|0,5|]}, \tag{A.4}$$

where ω_d, ζ_d, $\gamma_{d,1}$ and $\gamma_{d,2}$ (respectively) are set to obtain filters shown in Figures 1 and 5, and $C_{\gamma_{d,1}}$ and $C_{\gamma_{d,2}}$ are normalizing constants such that the integrals of the corresponding filters sum (respectively) to $\gamma_{d,1}$ and $\gamma_{d,2}$.

A.2 Training times

Figures below show the training times recorded for the synthetic experiments. We recall that the number of variables is $D = 3$. Results were averaged over 25 iterations over different generated data sets of size N ranging from 200 to up to 100000 in the linear setting and up to 10000 in the general setting. In the latter, obtaining the average time taking into account the cross-validation step to select the hyper-parameters is straightforward by multiplying the average training time by the total number of possible combinations. We observe that training times remain reasonable against data size. They should increase by adding more variables since a loop over all the remaining variables at each iteration of the Gram–Schmidt process is performed to select the component with the higher variance, which results in a total of $\frac{D(D+1)}{2}$ iterations for D variables.

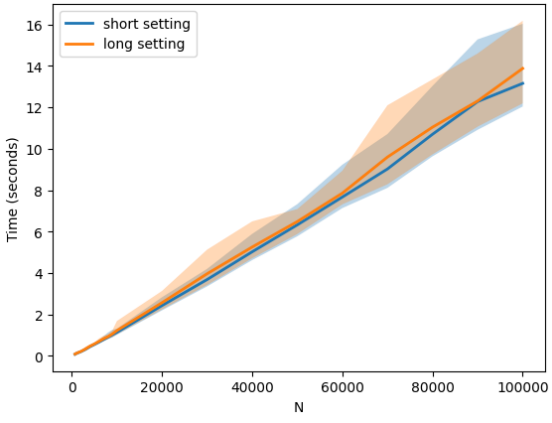


Figure A.1. Training time against the size of data in the linear case.

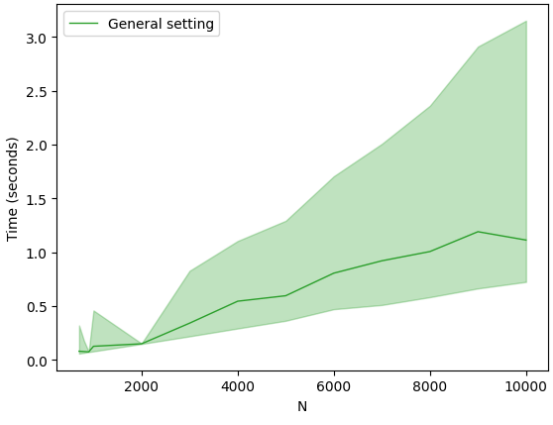


Figure A.2. Training time against the size of data in the general case.

Appendix B. Covariance estimation

Let $Z = (Z_t)_t$ be a stationary time-series and $\gamma_h := \mathrm{Cov}\,(Z_t, Z_{t+h}) = \mathrm{Cov}\,(Z_0, Z_h)$ its covariance function. Assuming without loss of generality that Z is centred, the sample covariance given by:

$$\hat{\gamma}_h := \frac{1}{N-h} \sum_{i=1}^{N-h} Z_i Z_{i+h},$$

is an unbiased estimator of the covariance function at time h. Based on ergodicity and/or dependence structure assumptions (short-term temporal correlation for example) it can be proven that this estimator is $\sqrt{N}$-consistent (see for *e.g.*, Martingale Limit Theory and its Application, Hall 1980 or Time Series: Theory and Methods, Davis 1991). The variance is the particular case where $h = 0$.