SINDy-SA framework: enhancing nonlinear system identification with sensitivity analysis

Gustavo T. Naozuka · Heber L. Rocha · Renato S. Silva · Regina C. Almeida

Received: 1 February 2022 / Accepted: 25 July 2022 / Published online: 30 August 2022
© The Author(s), under exclusive licence to Springer Nature B.V. 2022

Abstract Machine learning methods have revolutionized studies in several areas of knowledge, helping to understand and extract information from experimental data. Recently, these data-driven methods have also been used to discover structures of mathematical models. The sparse identification of nonlinear dynamics (SINDy) method has been proposed with the aim of identifying nonlinear dynamical systems, assuming that the equations have only a few important terms that govern the dynamics. By defining a library of possible terms, the SINDy approach solves a sparse regression problem by eliminating terms whose coefficients are smaller than a threshold. However, the choice of this threshold is decisive for the correct identification of the model structure. In this work, we build on the SINDy method by integrating it with a global sensitivity analysis (SA) technique that allows to hierarchize terms according to their importance in relation to the desired quantity of interest, thus circumventing the need to define the SINDy threshold. The proposed SINDy-SA framework also includes the formulation of different experimental settings, recalibration of each identified model, and the use of model selection techniques to select the best and most parsimonious model. We investigate the use of the proposed SINDy-SA framework in a variety of applications. We also compare the results against the original SINDy method. The results demonstrate that the SINDy-SA framework is a promising methodology to accurately identify interpretable data-driven models.

Keywords Sparse identification · Sensitivity analysis · Model selection · Differential equations · Data-driven methods

1 Introduction

Machine learning methods have been commonly used to understand behaviors, recognize patterns and make predictions from experimental data. Furthermore, another application of these methods, which has become popular in recent years, is the structural identification of mathematical models [2, 6, 7, 20, 25, 29, 35, 37]. These models, in turn, help to interpret the dynamics and allow the use of tools for mathematical analysis. Recently, Brunton et al. [2] have developed the sparse identification of nonlinear dynamics (SINDy) method, combining sparsity-promoting techniques and machine learning with nonlinear dynamical systems to discover governing equations from noisy measurement data. The only assumption about the structure of the
mathematical model is that there are only a few important terms that govern the dynamics. Thus, by defining a prior set of possible functions, the authors solved a sparse regression problem with the objective of determining the smallest number of terms in this set, for each equation of the dynamical system, required to accurately represent the data. The SINDy method has been applied to solve many system identification problems, such as the dynamics of COVID-19 global transmission [17], the Duffing oscillator [23], and empirical data from measles, varicella, rubella, and chickenpox datasets [12, 13].

Several other related approaches have been published in the literature in order to recover the governing equations from data. França et al. [7] combined several machine learning techniques to improve the robustness to noise in system identification problems. Cortiella et al. [6] proposed an iterative sparse-regularized regression method to recover nonlinear dynamical systems from noisy measurement data. The work aims to improve the accuracy and robustness of SINDy in the presence of noisy state measurements, by developing a reweighted $\ell_1$-regularized least-squares solver. Niven et al. [25, 26] used an inverse Bayesian method for system identification from time-series data and compared two Bayesian methods, based on the joint maximum a posteriori and variational Bayesian approximation, to the SINDy approach. These methods are also used to quantify the variances of the model parameters. Maddu et al. [20] proposed a statistical learning framework based on group-sparse regression to discover mathematical models from data. This framework can enforce conservation laws, ensure model equivalence and symmetries, by using the group iterative hard thresholding algorithm and stability selection. Wang et al. [35, 36] presented a variational framework for system identification of partial differential equations, based on stepwise regression. The authors also addressed the influences of variable fidelity and noise in the measurement data. Yang et al. [37] presented a machine learning framework for Bayesian system identification from noisy, sparse, and irregular observation data. This framework uses differentiable programming and performs Bayesian inference using Hamiltonian Monte Carlo sampling. Rudy et al. [29] proposed a sparse regression method to extract partial differential equations from time-series measurements in the spatial domain. This method selects the nonlinear and partial derivative terms of the governing equations that best fit the data.

In addition, there are a number of extensions of the SINDy approach to improve its accuracy and robustness or to discover other types of mathematical models. Subramanian [33] incorporated a scientific machine learning approach in the context of clinical trials to discover governing equations of idiopathic pulmonary fibrosis disease progression. This approach combines machine learning techniques, statistical methodologies, and scientific computing tools such as bootstrap sampling, cubic spline interpolation, Bayesian inference, and SINDy to discover the dynamics and quantify the uncertainty in the model parameters. Hoffmann et al. [11] extended the SINDy method to vector-valued ansatz functions, in order to estimate effective reaction networks from observations. In the proposed “reactive SINDy” approach, each function represents a particular reaction process. Boninsegna et al. [1] also extended the SINDy method to discover stochastic dynamical systems of biophysical processes. Jiang et al. [15] proposed a SINDy-LM modeling method, in which the SINDy approach is used to discover nonlinear dynamical systems from observation data and the Levenberg-Marquardt algorithm is used to improve the accuracy of the mathematical model identified by the SINDy algorithm. Hirsh [10] developed the theory of existing methods and proposed new techniques to model spatiotemporal data. These data-driven methods include dynamic mode decomposition (DMD), a dimensionality reduction method for time-varying linear dynamics, the Hankel Alternative View of Koopman (HAVOK) algorithm, and an uncertainty quantification for sparse identification of nonlinear dynamics (UQ-SINDy) method. The modified SINDy approach uses compressed sensing and Bayesian statistics to discover governing equations from data and quantifies model uncertainties. Brunton et al. [3] generalized the SINDy algorithm to identify nonlinear dynamical systems with external inputs and feedback control. Mangan et al. [21] proposed an alternative data-driven technique to identify networked nonlinear dynamical systems by using the SINDy algorithm. This technique can be used to recover governing equations that have rational function nonlinearities with cross terms. Kaheman et al. [16] also developed a variant of the SINDy algorithm to identify implicit dynamics and rational nonlinearities, by using multiple optimization algorithms and a model selection approach. This variant can be used to
recover implicit differential equations and conservation laws from limited and noisy data. Quade et al. [28] proposed a conceptual framework to recover dynamical systems in response to abrupt changes from limited data. This framework first detects the abrupt change and then applies the SINDy method to update a previously identified mathematical model with the fewest changes.

Although the SINDy method is often used to identify the structure of the mathematical model, this identification depends on the suitable choice of a threshold, which is used in the process of eliminating terms from the governing equations. One way to bypass the difficulty of this choice is to define a set of values for the threshold, run the method for each defined value, and select the best model from the resulting set of models, given the experimental data [2,22]. However, the best value for the threshold may not belong to the defined prior set, resulting in incorrect identification of the dynamical system. Thus, to discover the most parsimonious model that best fits the data, one must run the SINDy method for a sufficiently large set of threshold values, which consequently increases the computational cost of solving the problem.

On the other hand, sensitivity analysis (SA) is a technique that allows ranking the parameters of a mathematical model according to their importance in relation to the desired quantity of interest [5,31]. In our case, the quantity of interest is associated with the formulation of the sparse regression problem, described in detail in Sect. 2.1. In this work, we propose the SINDy-SA method that leverages the original SINDy approach, replacing the need to determine the threshold with a global sensitivity analysis technique. In particular, we use the elementary effects method, a simple global SA approach, able to rank the model parameters according to their importance. We integrate the SINDy-SA method in a general framework that encompasses other issues associated with the problem of nonlinear system identification. Specifically, we design different experimental settings, varying, for example, the set of possible candidate functions for the governing equations. After applying the system identification method for each experiment, a model recalibration step is performed to improve the accuracy of the parameter estimates. We then compare the resulting set of models against the experimental data in two ways: (i) constructing the Pareto curve that balances the accuracy and complexity of the models, and (ii) employing model selection criteria to select the best model. Any suitable model selection method can be used. In this work, we employ three different information criteria: the first- and second-order Akaike and the Bayesian information criteria, which weigh both the goodness-of-fit and the number of model parameters given the data. The framework using our SINDy-SA method is evaluated in a variety of applications using simulated data from a tumor growth model, a prey-predator model, a pendulum motion equation, and a compartmental model, and the obtained results are compared with the original SINDy method. To this end, we replace the SINDy-SA component with the original SINDy within the proposed general framework. We anticipate that the SINDy-SA framework is able to correctly identify the true model in all considered applications, outperforming the framework with the original SINDy method.

This paper is organized as follows. In Sect. 2, we explain some preliminary concepts required to understand the development of the proposed SINDy-SA method. Moreover, we describe in detail the SINDy-SA method and its implementation and show our general framework for solving system identification problems. In Sect. 3, we present the results obtained for the different applications by employing the frameworks with our method and the original SINDy approach. Finally, in Sect. 4, we point out some final remarks.

2 Problem statement and the SINDy-SA framework

Before introducing the new SINDy-SA method, we present in Sect. 2.1 some preliminary concepts about the sparse regression problem. Section 2.2 details our proposed SINDy-SA approach and its implementation. Finally, Sect. 2.3 describes our developed general framework for solving system identification problems.

2.1 Problem statement

Consider the problem of determining the dynamics of a system of $n$ variables $x^T = \{x_1, x_2, \ldots, x_n\}, x(t) \in \mathbb{R}^n$. Under the assumption of spatial homogeneity, we assume that the rate of change of these variables is given by the following dynamical system:

$$\frac{d}{dt} x(t) = f(x(t)).$$

(1)
in which the vector function $f(x(t))$ that we want to determine defines the interplay among the state variables. We approach the problem of determining the mathematical model (1) using sparse regression techniques, based on the fact that $f(x(t))$ has only a few terms that govern the dynamics of the system. The sparse regression technique is built upon the following available measured data that display a temporal history of the state variable vector $x(t)$ at multiple time instants $t_1, t_2, \ldots, t_m$:

$$X = \begin{bmatrix} x^T(t_1) \\ x^T(t_2) \\ \vdots \\ x^T(t_m) \end{bmatrix} = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \ldots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \ldots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \ldots & x_n(t_m) \end{bmatrix}. \tag{2}$$

From these data, we determine the corresponding temporal history of the state variable derivatives $\dot{x}(t)$, usually approximated numerically, which is likewise arranged in the form of an $m \times n$ matrix denoted by $\dot{X}$. With these definitions, the sparse regression problem aims to determine the $\ell \times n$ sparse matrix of coefficients $\Xi = [\xi_1 \xi_2 \ldots \xi_n]$ so that

$$\dot{X} = \Theta(X) \Xi, \tag{3}$$

in which $\Theta(X)$ is the $m \times \ell$ matrix of candidate nonlinear functions from the columns of $X$, and $\ell$ is the basis dimension for the function library $\Theta(X)$. This meaningful library includes candidate functions (polynomials, trigonometric functions, etc.) for the right-hand side of Eq. (1) and is built from the modeler’s a priori knowledge of potential functions capable of describing the experimental data behavior.

The sparse regression problem (3) can be solved using classical regression methods such as least-squares, ridge, lasso, and elastic net [8,14]. The original SINDy approach is implemented using the sequential thresholded ridge regression (STRidge) method [2]. Specifically, the SINDy method requires the definition of a threshold for the values of the components of the sparse vector $\xi_j$ below which the corresponding library functions are eliminated from the set of possible functions. The chosen threshold, denoted here by $\lambda$, is decisive for the correct identification of the mathematical model. Thus, determining its best value can be a hard task, especially when the dynamical system has parameter values with very different orders of magnitude. In such cases, additional approaches to overcome this difficulty are required [2,22]. For more information, we refer the reader to the Supplementary Material (Section SM-1). After determining the sparse matrix $\Xi$, the desired identified system is obtained as

$$\dot{x} = f(x) = \Xi^T (\Theta(x^T))^T. \tag{4}$$

2.2 The proposed SINDy-SA approach

The original SINDy method depends on the proper choice of the threshold, necessary to eliminate candidate functions from the governing equations. Due to the difficulty of this choice and the consequent increase in computational cost to determine its best value, as suggested in Brunton et al. [2], we propose in this work to integrate the sparse regression problem with a global SA technique. By using a SA technique that is able to rank the coefficient terms according to their importance with respect to $\dot{X}$, the selected quantities of interest (QoIs), we overcome the need of using a pre-defined threshold to eliminate candidate functions. Instead, the non-influential terms can be eliminated and the accuracy of the regression verified. In particular, we use the Morris method, also known as the elementary effects (EE) method, due to its simplicity and low computational cost [31]. Moreover, the EE method assesses not only the overall importance of the coefficients but also their interactions and nonlinear effects. The implementation of the proposed approach was carried out in the Python programming language, based on the PySINDy module, a sparse regression package containing different implementations for the SINDy method [32]. The EE method was developed using the SALib module, composed of commonly used SA methods [9].

Our proposed SINDy-SA sparse regression method is an iterative process encompassing the ridge regression estimation, the error computation, and the SA technique. At every $\tau$-th iteration, non-influential terms are eliminated until only influential terms remain. Algorithm 1 presents a pseudocode of the SINDy-SA method for solving the sparse regression problem (3), in which $\hat{\Xi}_\tau$ describes the estimate for the sparse vectors of coefficients in the iteration $\tau$. Additionally, Figure SM-1 in the Supplementary Material illustrates a flowchart of our method, graphically representing each instruction of Algorithm 1. We detail the three main features of our SINDy-SA method in the following.
Algorithm 1 SINDy-SA

1: \( \tau \leftarrow 0; \)
2: repeat
3: Compute a least-squares solution with \( \ell_2 \)-regularization for \( \hat{\Xi}_\tau \);
4: Find the derivative \( \hat{\xi}_\tau \) using \( \hat{\Xi}_\tau \);
5: Calculate the SSE between the derivatives \( \hat{\xi} \) and \( \hat{\xi}_\tau \);
6: if \( \tau > 0 \) and \( \text{SSE}_\tau (\hat{\xi}, \hat{\xi}_\tau) > M_\tau + \varepsilon D_\tau \) then
7: return \( \hat{\Xi}_{\tau-1} \);
8: end if
9: Perform sensitivity analysis of model parameters;
10: if all parameters are important then
11: return \( \hat{\Xi}_\tau \);
12: else
13: Eliminate terms with less importance in relation to the QoI;
14: end if
15: \( \tau \leftarrow \tau + 1; \)
16: end repeat

Ridge regression

In our SINDy-SA approach, we start by assigning to each column \( \xi_k \) of \( \hat{\Xi} \) the corresponding least-squares solution with \( \ell_2 \)-regularization. Specifically, defining \( \hat{\xi}_k \) as the \( k \)-th column of \( \hat{\xi} \), we use the ridge regression [8,14]. Of note, we have evaluated commonly used statistical regression methods to identify nonlinear dynamical systems, and have chosen the ridge regression method for its robustness and accuracy. It is defined as:

\[
\xi_{k\tau} = \arg\min_{\xi_k} \left\| \hat{\xi}_k - \Theta(X)\xi_k \right\|_2^2 + \alpha \left\| \xi_k \right\|_2^2
\]

(5)

to estimate the coefficient values, balancing two different criteria: the residual sum of squares and a penalty term that has the effect of preventing overfitting. The tuning parameter \( \alpha \) serves to control the relative impact of these two terms on the coefficient estimates. Different \( \alpha \) values are likely to lead to different coefficient estimates and, ultimately, to different models. James et al. [14] suggest that a possible way to select a value for \( \alpha \) is to use the cross-validation method. By defining a prior set of \( \alpha \) values and computing the cross-validation error for each defined value, they select the value of \( \alpha \) with the smallest cross-validation error. Inspired by this approach, our proposed framework also defines a set of \( \alpha \) values that composes different simulation scenarios together with the possible set of candidate functions. Ultimately, the best identified model is chosen by further employing a model selection technique, as detailed in Sect. 2.3.

Error computation

After the ridge regression estimation, we then calculate the \( \tau \)-th estimate for the derivative \( \hat{\xi}_\tau \) based on the library \( \Theta(X) \) and the estimates for the sparse vectors of coefficients \( \hat{\Xi}_\tau \) obtained by the ridge regression, according to the following equation:

\[
\hat{\xi}_\tau = \Theta(X)\hat{\Xi}_\tau.
\]

(6)

Next, we calculate the sum of squared errors (SSE) between the derivatives \( \hat{\xi} \), measured or approximated from the data, and \( \hat{\xi}_\tau \), given by Eq. (6), through:

\[
\text{SSE}_\tau (\hat{\xi}, \hat{\xi}_\tau) = \sum_{k=1}^{n} \sum_{i=1}^{m} (\hat{\xi}_k (t_i) - \hat{\xi}_{k\tau} (t_i))^2,
\]

(7)

where \( \hat{\xi}_k (t_i) \in \hat{\xi} \) and \( \hat{\xi}_{k\tau} (t_i) \in \hat{\xi}_\tau \). We store the SSE values for each iteration of the SINDy-SA method in order to determine whether the error between the current and previous iterations has significantly increased. In the first iteration (\( \tau = 0 \)), no error comparison is performed. The iterative process continues until a significant increase in the SSE occurs, ending the process. Defining what is significant is a tricky task since SSE fluctuations can occur depending on the problem. To overcome this issue, we define an iteration window, denoted by \( W \), in which we collect statistics from part of the previous evolution of the SSE, which are used as a reference. In mathematical terms, we check the condition:

\[
\text{SSE}_\tau (\hat{\xi}, \hat{\xi}_\tau) > M_\tau + \varepsilon D_\tau,
\]

(8)

where \( M_\tau \) and \( D_\tau \) represent, respectively, the mean and the standard deviation of the errors of previous iterations belonging to \( W \), and \( \varepsilon \) is a scaling factor. Some adjustments are required when \( \tau < W \) and are given according to the conditions described in Table 1.

Note that, for the second iteration of the algorithm (\( \tau = 1 \)), the standard deviation \( D_1 \) would be zero considering only the error of the previous iteration. For this reason, we consider \( D_1 \) as 10% of the SSE obtained in the first iteration. The values of the parameters \( W \) and \( \varepsilon \) can be determined by observing the sequence of SSE values, numerically or graphically, and the goodness-of-fit of the resulting mathematical model simulation against the experimental data. In all our experiments, we set the window size \( W = 3 \) iterations, while the parameter \( \varepsilon \) varies for each experiment.

If the condition (8) is satisfied, the algorithm ends, and the coefficient estimates of the iteration \( \tau - 1 \),
whose error was considered significantly smaller compared to the iteration \( \tau \), are used for the construction of governing equations. Otherwise, the procedure continues and performs the SA of model parameters.

**Sensitivity analysis**

In the SA of model parameters, we firstly determine admissible variations for the non-zero coefficients, being \( \pm 20\% \) of the estimated value for the coefficient. Given one or more analysis time instants, we calculate the sensitivity indices \( \mu_{k,j}^{*} \) and \( \sigma_{k,i} \) that indicate the influence and nonlinear importance of the \( i \)-th parameter on the \( k \)-th QoI, respectively. The higher the values of these sensitivity indices, the more influential are the corresponding parameters, so that small variations in their values have a great impact on the estimation of \( \hat{X}_\tau \). For more details about the EE method, see Supplementary Material (Section SM-2). In order to facilitate sorting parameters by their order of importance with respect to QoI and to take into account both their direct and nonlinear effects, we combine the sensitivity indices into a single metric:

\[
S_{k,i} = \sqrt{\left(\mu_{k,j}^{*}\right)^2 + \left(\sigma_{k,i}\right)^2}.
\]

We develop a scoring scheme to integrate SA at multiple time instants. Each model parameter receives a score \( P_{k,i} \), depending on its rank of importance which is updated over the analysis times, starting from 0. Of note, \( k = 1, \ldots, n \) and \( i = 1, \ldots, d \), where \( n \) is the number of state variables, and \( d \) is the number of model parameters. Once, at a given time of analysis, the model parameters are ordered from the least important parameter \( \hat{\xi}_{k,1} \) to the most important parameter \( \hat{\xi}_{k,d} \), we ensure that the latter terms are kept by assigning \(-1\) to \( P_{k,d} \). Next, we increment an increasing score \( \pi \) to the total score \( P_{k,i} \) for the remaining parameters of the model. In this way, more influential parameters receive a lower score, and parameters with less influence receive a higher score. After analyzing all time instants, we check if \( P_{k,i} = -1 \) for all \( k = 1, \ldots, n \) and \( i = 1, \ldots, d \), meaning that all parameters are important. Otherwise, we eliminate the terms that have the highest total score \( P_{k,i} \), being able to eliminate more than one candidate function for each state variable. Algorithm 2 presents the pseudocode of the procedure for sensitivity analysis at multiple time instants.

If all parameters are considered important, the algorithm ends, in which case the coefficient estimates of the iteration \( \tau \) are used to construct the governing equations. Otherwise, the procedure returns to the first step calculating a least-squares solution with \( \ell_2 \) regularization for \( \hat{\Xi}_\tau \). It is important to observe that, in our approach, it is not necessary to specify a maximum number of iterations because eventually or all the model parameters are considered important or condition (8) is satisfied. Indeed, if a model parameter is erroneously eliminated according to SA, the SSE between the current and previous iterations tends to increase, and the condition (8) must be satisfied.
Algorithm 2 Sensitivity analysis at multiple time points
1: Initialize the total score \( P_{k,i} \leftarrow 0 \), with \( k = 1, \ldots, n \) and 
   \( i = 1, \ldots, d \);
2: \textbf{for each} time instant for analysis \textbf{do}
3: \quad \textbf{for each} state variable \( k = 1, \ldots, n \) do
4: \quad \quad Order the parameters associated with the state variable 
   \( k \) according to their combined sensitivity index \( S_{k,i} \);
5: \quad \quad \( P_{k,d} \leftarrow -1 \), where \( \hat{\xi}_{k,d} \) is the most important parameter 
   in relation to the QoI;
6: \quad \quad \( \pi \leftarrow 1 \);
7: \quad \quad \textbf{for each} parameter \( \hat{\xi}_{k,i} \), with \( i = d - 1, \ldots, 1 \) do
8: \quad \quad \quad \textbf{if} \( P_{k,i} \neq -1 \) \textbf{then}
9: \quad \quad \quad \quad \( P_{k,i} \leftarrow P_{k,i} + \pi \);
10: \quad \quad \quad \quad \( \pi \leftarrow \pi + 1 \);
11: \quad \quad \quad \textbf{end if}
12: \quad \quad \textbf{end for}
13: \quad \textbf{end for}
14: \textbf{end for}
15: \textbf{if} \( P_{k,i} = -1 \), for all \( k = 1, \ldots, n \) and \( i = 1, \ldots, d \) \textbf{then}
16: \quad All parameters are important;
17: \textbf{else}
18: \quad \textbf{return} the terms with the highest total score \( P_{k,i} \) (i.e., the 
   least importance in relation to the QoI);
19: \textbf{end if}

2.3 General framework for system identification

The proposed framework for solving the problem of identifying nonlinear dynamical systems is schemati-
cally represented in Fig. 1. It contains four distinct components that encompass the sequence of tasks required 
for ultimately obtaining the best identified model.

The first component is the Input block in which we gather the available experimental data set \( X \) and the modeler’s prior knowledge of a library \( \Theta(X) \) of candidate basis functions that have the potential to explain 
the data. Such prior knowledge about expected terms in \( f(x) \) plays a crucial role in the dynamical sys-
tem identification and supports the construction of the Experimental set. In this second component, we investi-
gate different potential configurations in the modeling by varying \( \Theta(X) \) and the regularization parameter \( \alpha \) of the ridge regression. Each experiment is then submitted to the model identification block, the central 
core of the framework. In this step, the system identification method (SINDy-SA or the original SINDy) 
is employed for each experimental configuration. In Fig. 1, the SINDy-SA block illustrates its main three 
steps described in Sect. 2.2: the ridge regression for estimating the sparse vectors of coefficients \( \hat{\xi}_\tau \), the 
computation of SSE between the derivatives \( \dot{X} \) and \( \dot{\hat{X}}_\tau \), and the sensitivity analysis to eliminate less important 
terms from the governing equations.

Once the mathematical model is identified, the coefficients learned by the optimizer are recalibrated to
improve accuracy. In this work, we use the Levenberg–Marquardt algorithm due to its robustness, as suggested in Jiang et al. [15]. After performing the model identification step for the $N$ experiments, we discover a set of at most $N$ data-driven models (Model set component). The best and most parsimonious data-driven model, which composes the Output block of the framework, is then selected using model selection techniques, as suggested in Kaheman et al. [16] and Mangan et al. [21]. In this step, we only consider candidate models whose goodness-of-fit between their simulations and the available data are computable. We use some of the most used model selection methods for comparing both nested and non-nested models: the Akaike Information Criterion (AIC), the second-order Akaike (AICc), and the Bayesian Information Criterion (BIC) [4]. The probability of the $i$-th model being the best given the data is obtained by each criterion weights $w_i$, which improve information about the selected model. More details on the information criteria and their weights are provided in the Supplementary Material (Section SM-3) for completeness.

An important remark still has to be made when the original SINDy is used as the system identification method in the developed framework because the need of defining the threshold $\lambda$. To this end, the construction of the experimental set must also include a value for $\lambda$. This means that, when the original SINDy is used in the proposed framework, the experimental set is defined by the triples $\{\Theta_k(X), \alpha_k, \lambda_k\}, k = 1, \ldots, N$. In this way, we also vary possible values for the SINDy threshold, as suggested in Brunton et al. [2].

### 3 Applications and results

In this section, we investigate the use of the proposed SINDy-SA framework in a variety of applications with different behaviors. We use simulated data from a prey–predator model (Sect. 3.1), a tumor growth model (Sect. 3.2), a pendulum motion equation (Sect. 3.3), and a compartmental model (Sect. 3.4). The results are compared against the original SINDy approach, as the system identification method within the proposed framework. We also investigate the proposed SINDy-SA framework in scenarios with limited and noisy data.

In all applications, we use the LSODA (Livermore Solver for Ordinary Differential Equations) method [27] to numerically simulate the true models and generate measurement data. It is also employed to solve the identified models and compare their simulations against the data. LSODA is a well-known numerical solver for ordinary differential equations that uses adaptive step sizes and automatically switches between the nonstiff Adams method and the stiff BDF method. These properties provide robustness to the framework.

We approximate the derivative $\dot{X}$ from the generated data using the second-order finite difference method. As described in Sect. 2.3, in order to verify if the system identification methods are capable of discovering known mathematical models, we define the experimental settings by choosing candidate functions that include the terms present in the true model with additional functions to be eliminated from the governing equations. In addition, we define the values for the regularization parameter $\alpha$ and the threshold $\lambda$ by varying different orders of magnitude. At the end of the process, the modeler may need to refine these choices and build new experimental settings.

#### 3.1 Prey–predator model

In our first application, we use the prey–predator model to generate the simulated data that will be used in the sparse identification problem. This model describes the evolution of interactions between prey and predator over time $t$, until $t = t_f = 20.0$, and is given by:

$$
\begin{align*}
\frac{dx}{dt} &= c_0 x + c_1 x y, \\
\frac{dy}{dt} &= c_2 y + c_3 x y, \\
x(0) &= x_0, \quad y(0) = y_0,
\end{align*}
$$

(10)

where $x(t)$ and $y(t)$ denote prey and predator populations, respectively. In this illustrative example, we use the prey birth rate $c_0 = 1.0$, the predator death rate $c_2 = -1.0$, and the interaction coefficients $c_1 = -1.0$ and $c_3 = 1.0$, and the initial conditions are set equal to $(x_0, y_0) = (4.0, 1.0)$. The training set is built by simulating the model and collecting the numerical solutions at $m = 200$ equally distributed time points. Using these collected solutions, we approximate the derivative $\dot{X}$ and build a library $\Theta(X)$ consisting only of polynomials, varying their maximum degree $P_i, i \in \{2, 3, 4, 5\}$. For example, if the maximum degree of the polynomial is $P_2$ with two state variables representing prey and predator populations, there are six candidate functions so that...
Table 2 | Behavior of the SINDy-SA method for the prey–predator model (10), using the maximum degree of the polynomial $P_2$ and the tuning parameter $\alpha = 10^{-3}$. We present $\hat{\mathbf{X}}$, SSE($\hat{\mathbf{X}}$), $\mathcal{M}$, and $\varepsilon \mathcal{D}$ at each iteration of the method, dropping the subscript $\tau$ to ease notation. The identified model is highlighted in italics.

| $\tau$ | Model | SSE($\hat{\mathbf{X}}$, $\hat{\mathbf{X}}$) | $\mathcal{M}$ | $\varepsilon \mathcal{D}$ |
|-------|-------|-----------------|----------|--------|
| 0     | $\begin{cases} \dot{x} = -0.021 + 1.043x + 0.034y - 0.017x^2 - 0.981xy - 0.014y^2 \\ \dot{y} = -0.021 - 0.033x - 1.045y + 0.013x^2 + 0.982xy + 0.018y^2 \end{cases}$ | 0.065 | – | – |
| 1     | $\begin{cases} \dot{x} = 0.111 + 0.981x - 0.006y - 0.981xy - 0.0023y^2 \\ \dot{y} = -0.003 + 0.14x - 1.014y + 0.981xy + 0.009y^2 \end{cases}$ | 0.125 | 0.065 | 0.651 |
| 2     | $\begin{cases} \dot{x} = 0.015 + 0.981x - 0.015y - 0.981xy \\ \dot{y} = -0.015 + 0.014x - 0.981y + 0.981xy \end{cases}$ | 0.153 | 0.095 | 2.990 |
| 3     | $\begin{cases} \dot{x} = -0.002 + 0.991x - 0.989xy \\ \dot{y} = -0.002 - 0.988y + 0.989xy \end{cases}$ | 0.205 | 0.114 | 3.648 |
| 4     | $\begin{cases} \dot{x} = 0.990x - 0.989xy \\ \dot{y} = -0.989y + 0.989xy \end{cases}$ | 0.206 | 0.161 | 3.252 |
| 5     | $\dot{x} = -0.167x$ \quad $\dot{y} = -0.006y$ | 1115.975 | 0.188 | 2.501 |

$\Theta(X) = \begin{bmatrix} 1 & x & y & x^2 & xy & y^2 \end{bmatrix}$

is the $200 \times 6$ matrix of potential candidate functions. For each chosen maximum degree, we also vary the tuning parameter $\alpha \in \{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1\}$ of the ridge regression, totaling $N = 20$ experiments for the SINDy-SA method. For the original SINDy approach, we additionally vary the threshold $\lambda \in \{10^{-1}, 10^0, 10^1\}$, ending up with $N = 60$ experiments. In all SINDy-SA experiments, we use $\varepsilon = 100.0$ and perform the SA at $t = t_f$.

To detail the steps of Algorithm 1, consider the experiment in which the maximum degree of the polynomial is $P_2$ and the tuning parameter is $\alpha = 10^{-3}$. With these settings, the SINDy-SA method leads to the sequence of $\hat{\mathbf{X}}$, SSE($\hat{\mathbf{X}}$, $\hat{\mathbf{X}}$), $\mathcal{M}$, and $\varepsilon \mathcal{D}$ displayed in Table 2 for each iteration. Figure 2 graphically relates the SSE($\hat{\mathbf{X}}$, $\hat{\mathbf{X}}$) with the sum $\mathcal{M} + \varepsilon \mathcal{D}$, where the bar graphs represent the SSE, the points denote the mean $\mathcal{M}$, and the vertical lines describe the margin of error $\varepsilon \mathcal{D}$. Figure 3 displays the heatmap of the total scores $\mathcal{P}_{k,i}$ calculated for all candidate functions in each iteration of the method. The corresponding combined sensitivity index $\mathcal{S}_{k,i}$ are indicated inside each heatmap cell. The total score $\mathcal{P}_{k,i}$ depicts four possible situations:

(i) $\mathcal{P}_{k,i} = -1$: the term is considered to be more important in relation to QoI by SA and thus should be kept in the governing equations;
(ii) $\mathcal{P}_{k,i} = 0$: the term was eliminated in some previous iteration;
(iii) $\mathcal{P}_{k,i} = 1, \ldots, 4$: the term has medium importance and can be deleted or maintained; and
(iv) $\mathcal{P}_{k,i} = 5$: the term is considered to be less important in relation to QoI by SA and thus should be eliminated from the governing equations.

According to Table 2 and Figs. 2 and 3, the method performed six iterations, in which eliminating two can-
Fig. 3 Heatmap of the total scores $P_{k,i}$ for all candidate functions of the dynamical system in each iteration of the SINDy-SA method. The corresponding combined sensitivity index $S_{k,i}$ are indicated inside each heatmap cell. The darkest color indicates terms to be eliminated in the current iteration; the white color indicates terms eliminated in previous iterations; the lightest gray indicates more important terms.

Candidate functions from the mathematical model. SSE increased as terms are eliminated, remaining relatively small until $\tau = 4$. In this iteration, the elimination of the nonlinear terms leads to a significant increase in the SSE of the iteration $\tau = 5$ that satisfies the stopping criterion (8). Thus, the identified model is the one associated with iteration $\tau = 4$ (step 7 of Algorithm 1), which corresponds to the true model although with parameters slightly different from 1.0. Those estimates are improved using the Levenberg-Marquardt optimization algorithm through which model (10) is recovered. It is important to note that, throughout the algorithm, each eliminated candidate function is associated with a state variable of the dynamical system. This behavior is due to the fact that we performed the SA of the parameters only at the final time of the experimental time frame for this illustrative example. If SA is performed at multiple time instants, each iteration can eliminate varying numbers of candidate functions. For a generic application, in which the true model is not known a priori, we recommend that SA be performed at multiple time instants.

Running the SINDy-SA method for all 20 experiments resulted in four different mathematical models. For comparison, running the original SINDy method for all 60 experiments resulted in nine different mathematical models. In order to select the best model among the resulting set of models, we numerically simulate each dynamical system, given the initial conditions and the discretized time domain defined in Eq. (10), and compare with the measurement data calculating $\text{SSE}(X, \hat{X})$. Using $m = 200$ data, we calculate AIC, $\text{AIC}_c$, and BIC, and their corresponding weights, as shown in Table 3. Of note, Figure SM-2 in the Supplementary Material illustrates the Pareto curve relating the SSE to the model complexity, measured in terms of the number of parameters $d$.

According to Table 3, the first model is the best according to all the three considered information criteria for the SINDy-SA method, with probability equal to one. This is due to the fact that the number of parameters in this model is relatively small, and the SSE($X, \hat{X}$) is significantly lower compared to the other identified models. Likewise, the model selected by the information criteria for the original SINDy approach also structurally corresponds to the true model (10). Figure 4 shows a comparison between the observed data and the numerical solution of the selected model for both SINDy-SA and SINDy methods, since they led to the same best identified (true) model. As we will see in the following, this is not always the case. Moreover, the results obtained for this application demonstrate that the model recalibration step can be essential for the accurate identification of the true mathematical model.
Table 3  Model selection results for the prey–predator application. Subscripts $w_i$ indicate the model selection criterion weights. Models selected as the best ones are indicated by italics

| Method | Model | $d$ | SSE$(X, \hat{X})$ | AIC | AICc | BIC | AICc $w_i$ | BIC $w_i$ |
|--------|-------|----|------------------|-----|------|-----|----------|----------|
| SINDy-SA | 1     | 4  | $2.910 \times 10^{-24}$ | $-11890.428$ | $-11890.223$ | $-11877.235$ | 1.000 | 1.000 | 1.000 |
|         | 2     | 18 | 0.950            | $-1034.015$ | $-1030.236$ | $-974.646$ | 0.000 | 0.000 | 0.000 |
|         | 3     | 22 | 0.380            | $-1208.986$ | $-1203.269$ | $-1136.423$ | 0.000 | 0.000 | 0.000 |
|         | 4     | 28 | 650.070          | 291.753    | 301.250    | 384.106    | 0.000 | 0.000 | 0.000 |
| SINDy   | 1     | 4  | $2.910 \times 10^{-24}$ | $-11890.428$ | $-11890.223$ | $-11877.235$ | 1.000 | 1.000 | 1.000 |
|         | 2     | 8  | 1339.752         | 396.384    | 397.138    | 422.771    | 0.000 | 0.000 | 0.000 |
|         | 3     | 6  | 94.667           | $-137.591$ | $-137.156$ | $-117.801$ | 0.000 | 0.000 | 0.000 |
|         | 4     | 4  | 760.018          | 275.005    | 275.210    | 288.198    | 0.000 | 0.000 | 0.000 |
|         | 5     | 1  | 821.749          | 284.624    | 284.644    | 287.922    | 0.000 | 0.000 | 0.000 |
|         | 6     | 6  | 1062.334         | 345.981    | 346.417    | 365.771    | 0.000 | 0.000 | 0.000 |
|         | 7     | 8  | 1171.715         | 369.581    | 370.335    | 395.968    | 0.000 | 0.000 | 0.000 |
|         | 8     | 4  | 1489.375         | 409.559    | 409.764    | 422.752    | 0.000 | 0.000 | 0.000 |
|         | 9     | 3  | 1407.354         | 396.230    | 396.352    | 406.125    | 0.000 | 0.000 | 0.000 |

Fig. 4  Comparison between the observed data $X$ and the numerical solution $\hat{X}$ of the best predator–prey model (shown in the bottom right) identified by the proposed framework using either the SINDy-SA method or the original SINDy method. The observed data are simulated from the prey–predator model (10), which corresponds to the best identified dynamical system.

3.2 Tumor growth model

In our next application, we consider the logistic growth model, previously calibrated using 14 tumor volume (in mm$^3$) data points from the breast cancer growth (GI-101A xenografts) in athymic mice, distributed over 114 days [24]. By denoting $x(t)$ as the tumor volume at time $t$, the evolution of the tumor volume over time, $0 < t \leq t_f = 300$ days, is then given by:

\[
\begin{align*}
\frac{dx}{dt} &= c_0 x + c_1 x^2, \\
x(0) &= x_0,
\end{align*}
\]

where $c_0 = 0.028$ /day, $c_1 = -3.305 \times 10^{-6}$ /day $\times$ mm$^{-3}$, and $x_0 = 245.185$ mm$^3$. The absolute value of $c_1$ is obtained by dividing the growth rate $c_0$ by the carrying capacity 8472 mm$^3$. The derivatives $\dot{X}$ are approximated from the numerical solution obtained by simulating model (11) at $m = 300$ evenly distributed time points. As in the previous application, we define $N = 20$ experiments for the SINDy-SA method by building $\Theta(X)$ using polynomials of degree at most equal to $\Theta_i$, $i \in \{2, 3, 4, 5\}$, and varying $\alpha \in \{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1\}$. We also vary the threshold $\lambda \in \{10^{-7}, 10^{-6}, 10^{-5}\}$, which yields
Mathematical models identified by the SINDy-SA and SINDy methods after running all experiments using simulated data from the logistic model (11)

| Method  | Model | Equation                  |
|---------|-------|---------------------------|
| SINDy-SA | 1     | \( \dot{x} = 0.028x - 3.305 \times 10^{-6}x^2 \) |
|         | 2     | \( \dot{x} = 0.028x - 3.305 \times 10^{-6}x^2 - 2.393 \times 10^{-16}x^3 \) |
| SINDy   | 1     | \( \dot{x} = 1.729 \times 10^{-5} + 0.028x - 3.305 \times 10^{-6}x^2 \) |
|         | 2     | \( \dot{x} = 33.862 - 5.169 \times 10^{-4}x \) |
|         | 3     | \( \dot{x} = 0.013x \) |

Model selection results for the tumor growth application. Subscripts \( w_i \) indicate the model selection criterion weights. Models selected as the best ones are indicated by italics.

| Method  | Model | \( d \) | \( \text{SSE}(\hat{X}, \hat{\dot{X}}) \) | \( \text{AIC} \) | \( \text{AIC}_c \) | \( \text{BIC} \) | \( \text{AIC} \, w_i \) | \( \text{AIC}_c \, w_i \) | \( \text{BIC} \, w_i \) |
|---------|-------|--------|-----------------|---------|---------|---------|---------------|---------------|---------------|
| SINDy-SA | 1     | 2      | 8.103 \times 10^{-17} | -12822.639 | -12822.598 | -12815.231 | 1.000 | 1.000 | 1.000 |
|         |       | 3      | 1.842 \times 10^{-5}  | -4975.767  | -4975.686  | -4964.656  | 0.000 | 0.000 | 0.000 |
|         | 2     | 2      | 1.157 \times 10^{-4}  | -4424.609  | -4424.527  | -4413.497  | 1.000 | 1.000 | 1.000 |
| SINDy   | 1     | 3      | 153697672.305         | 3948.015   | 3948.055   | 3955.422   | 0.000 | 0.000 | 0.000 |
|         | 2     | 2      | 2089401837.561        | 4728.908   | 4728.922   | 4732.612   | 0.000 | 0.000 | 0.000 |

\( N = 60 \) experiments for the original SINDy approach. We use \( \epsilon = 10.0 \) and we perform the SA at \( t = t_f \).

The execution of the SINDy-SA method for all 20 experiments resulted in only two different mathematical models. For comparison, the execution of the original SINDy method for all 60 experiments resulted in three different mathematical models. Table 4 presents the sets of models discovered by each system identification method. The model comparison of the identified models is performed, and the results are displayed in Table 5. For completeness, Figure SM-3 in the Supplementary Material illustrates the corresponding Pareto curves.

All three-model selection criteria agreed in choosing the best models with probability equal to one, as shown in Table 5. While the SINDy-SA method identified the correct model, notice that the SINDy approach could not capture the correct structure of the model since it wrongly incorporates a source term. Figure 5 illustrates a comparison between the observed data and the numerical solution of the selected best model for each sparse identification method as well as the corresponding mathematical description. The results are quite similar visually, but the best model identified by the SINDy method led to an SSE 13 orders of magnitude higher than that identified by the SINDy-SA method. Of note, despite the SINDy-SA method having identified two mathematical models considering different experimental settings, the information criteria correctly selected the most parsimonious model that best fits the simulated data. In comparison, the original SINDy approach fails to solve the system identification problem in this example, mainly due to the difference between the orders of magnitude of the two terms in the true model. Therefore, the results obtained for this application show a limitation of using a threshold to eliminate candidate functions from the governing equations and highlight the importance of performing the SA of model parameters as in our proposed approach, allowing us to correctly discover the governing equations that generated the data.

3.3 Pendulum motion model

In this subsection, we apply the pendulum equation to generate the simulated data that is used in the dynamical system identification problem. The pendulum motion over time \( t \), driven by a damped harmonic oscillation, is described in terms of the angle \( x(t) \) and the angular velocity \( y(t) \) through the following system of differen-
SINDy-SA framework: enhancing nonlinear system 2601

(a) SINDy-SA method

\[ \dot{x} = 0.028x - 3.305 \times 10^{-6}x^2 \]

(b) SINDy method

\[ \dot{x} = 1.729 \times 10^{-5} + 0.028x - 3.305 \times 10^{-6}x^2 \]

Fig. 5 Comparison between the observed data \( \hat{X} \) and the numerical solution \( X \) of the best tumor growth models selected by the used information criteria, which are shown inside the graphs.

Although the dynamics are quite similar, the model in (b) includes a source term (shown in gray) that is not present in the true model.

In this example, we use the parameters \( c_0 = -0.25 \) and \( c_1 = -5.0 \), and we set \((x_0, y_0) = (\pi - 0.1, 0.0)\). System (12) is integrated up to time \( t_f = 30.0 \), and we collect the numerical solutions at \( m = 301 \) evenly distributed time points, which are used to approximate the derivative \( \dot{X} \). Due to the specific dynamics observed in this problem, we build the library \( \Theta(X) \) using both polynomials, varying their maximum degree \( P_i, i \in \{1, 2, 3\} \), and trigonometric functions (sine and cosine), varying their frequency \( F_j, j \in \{1, 2, 3\} \). If, for example, the maximum degree of the polynomial is \( P_1 \) and the frequency of the trigonometric functions is \( F_1 \), the library \( \Theta(X) \) is given by:

\[ \Theta(X) = \begin{bmatrix} 1 & x & y & \sin(x) & \cos(x) & \sin(y) & \cos(y) \end{bmatrix} \]

which contains seven candidate functions. As in the previous examples, for each chosen maximum degree and frequency, we also vary \( \alpha \in \{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1\} \), which means that we have \( N = 45 \).
experiments for the SINDy-SA method. These experiments are also combined with possible variations of the threshold \( \lambda \) for the original SINDy approach, leading to \( N = 135 \) experiments by setting \( \lambda \in \{10^{-3}, 10^{-2}, 10^{-1}\} \). In all SINDy-SA experiments, we use \( \varepsilon = 10^3 \) and we perform the SA at three time instants over the experimental time frame, \( t = 10.0, 20.0, 30.0 \), to keep track of the dynamics of the pendulum motion.

The SINDy-SA and SINDy methods identified four and 84 different mathematical models, respectively. Table 6 details the model selection results and indicates the best model among the candidate set. To complement these results, Figure SM-4 in the Supplementary Material illustrates the corresponding Pareto curves relating the SSE with the model complexity.

As shown in Table 6, the first pendulum motion model was selected as the best among the four models identified by the SINDy-SA method for the three information criteria with weights of 100%. The best identified model is indeed the true model, which was also correctly identified by the original SINDy approach. In this particular application, it is worth mentioning the crucial role the model recalibration step plays. Without this step, the model selection criteria would have selected the fourth model identified by the SINDy approach, which is a much more complex model with three and five additional terms in the governing equations for the evolution of angle \( x(t) \) and angular velocity \( y(t) \), respectively. Figure 6 displays a comparison between the experimental data and the numerical solution of the best model for both system identification methods, with SSE of the order of \( 10^{-7} \).

### 3.4 Compartmental model

Our fourth application involves using the epidemiological SIR compartmental model to generate the simulated data as input to both SINDy-SA and SINDy methods. This model describes the evolution of a fixed population of \( N \) individuals over time \( t \) divided into three compartments: susceptible \( S(t) \), infected \( I(t) \), and recovered \( R(t) \). For \( t \leq t_f \), it is represented by:

\[
\begin{align*}
\frac{dS}{dt} &= c_0 SI, \\
\frac{dI}{dt} &= c_1 SI + c_2 I, \\
\frac{dR}{dt} &= c_3 I,
\end{align*}
\]

\( S(0) = S_0, \quad I(0) = I_0, \quad R(0) = R_0, \quad N = S(t) + I(t) + R(t). \)

Table 6 Model selection results for the pendulum motion application. Subscripts \( w_i \) indicate the model selection criterion weights. Models selected as the best ones are indicated by italics.

| Method | Model | \( d \) | SSE(\( X, \hat{X} \)) | AIC | AICc | BIC | AIC \( w_i \) | AICc \( w_i \) | BIC \( w_i \) |
|--------|-------|--------|-----------------|-----|------|-----|-------------|-------------|-------------|
| SINDy-SA | 1 | 3 | \( 6.554 \times 10^{-7} \) | \(-5997.491\) | \(-5997.410\) | \(-5986.370\) | 1.000 | 1.000 | 1.000 |
| 2 | 17 | 0.057 | \(-2546.523\) | \(-2544.360\) | \(-2483.502\) | 0.000 | 0.000 | 0.000 |
| 3 | 35 | 0.115 | \(-2297.928\) | \(-2288.418\) | \(-2168.179\) | 0.000 | 0.000 | 0.000 |
| 4 | 43 | 0.014 | \(-2926.289\) | \(-2911.565\) | \(-2766.883\) | 0.000 | 0.000 | 0.000 |
| SINDy | 1 | 10 | 0.003 | \(-3419.983\) | \(-3419.224\) | \(-3382.912\) | 0.000 | 0.000 | 0.000 |
| 2 | 6 | 0.204 | \(-2184.844\) | \(-2184.558\) | \(-2162.601\) | 0.000 | 0.000 | 0.000 |
| 3 | 3 | \( 6.554 \times 10^{-7} \) | \(-5997.491\) | \(-5997.410\) | \(-5986.370\) | 1.000 | 1.000 | 1.000 |
| 4 | 11 | 0.248 | \(-2115.911\) | \(-2114.997\) | \(-2075.132\) | 0.000 | 0.000 | 0.000 |
| 5 | 12 | 0.331 | \(-2026.392\) | \(-2025.309\) | \(-1981.907\) | 0.000 | 0.000 | 0.000 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 80 | 43 | 0.015 | \(-2889.626\) | \(-2874.902\) | \(-2730.220\) | 0.000 | 0.000 | 0.000 |
| 81 | 25 | 0.005 | \(-3281.282\) | \(-3276.554\) | \(-3188.604\) | 0.000 | 0.000 | 0.000 |
| 82 | 10 | \( 6.391 \times 10^{-4} \) | \(-3911.838\) | \(-3911.079\) | \(-3874.767\) | 0.000 | 0.000 | 0.000 |
| 83 | 41 | 0.001 | \(-3695.703\) | \(-3682.406\) | \(-3543.712\) | 0.000 | 0.000 | 0.000 |
| 84 | 29 | 0.010 | \(-3044.384\) | \(-3037.964\) | \(-2936.878\) | 0.000 | 0.000 | 0.000 |
By construction, $c_i$ are the transition coefficients that satisfy $c_0 = -c_1$ and $c_2 = -c_3$, so that $R(t)$ is known from the infected individuals $I(t)$. In this way, we can simplify the sparse identification problem of this model by removing the equation associated with the recovered individuals. To generate the training data, we collect the numerical solution of $m = 400$ evenly distributed time points using $t_f = 40.0$, the infection coefficient $c_0 = -4 \times 10^{-4}$ and the recovery rate $c_2 = -0.04$. We set $S_0 = 997$ and $I_0 = 3$ so that $N = 1000$ individuals ($R_0 = 0$). From the collected solution, we approximate the derivative $\dot{X}$ and build the SINDy-SA experiments by setting $\Theta(X)$ consisting of polynomials, varying their maximum degree $P_i$, $i \in \{2, 3, 4, 5\}$, and $\alpha \in \{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1\}$. Those $N = 20$ experiments are solved using $\epsilon = 100.0$ and performing the SA at $t = 10.0, 20.0, 30.0, 40.0$ to capture the overall solution behavior over the considered time frame. For the original SINDy method, we consider $N = 60$ experiments by additionally varying $\lambda \in \{10^{-5}, 10^{-4}, 10^{-3}\}$.

Table 7 shows the model selection results for the 10 and 14 models identified by the SINDy-SA and original SINDy methods, respectively. The corresponding Pareto curves relating the SSE to the number of model parameters are illustrated in Figure SM-5 in the Supplementary Material. Of note, although both approaches identified many dynamical systems, the information criteria were able to correctly select the true model that originated the experimental data for the SINDy-SA method. However, the model selected for the original SINDy approach does not correspond to the true model since it incorporates a source term in the time evolution of the infected population $I(t)$ that is not present in the true model. In both cases, there was great evidence supporting the choice of the best models. Figure 7 shows a comparison between the experimental data and the numerical solution of the selected best models, whose mathematical description is also shown inside the graphs.

As in the tumor growth model application, the original SINDy approach also fails to solve the system identification problem. Therefore, the results obtained for this application show that the SA of model parameters in our proposed SINDy-SA framework circumvents the difficulty of choosing the threshold and allows the correct identification of the true dynamical system.

3.5 Remarks on limited and noise-contaminated data scenarios

It is worth commenting on the robustness of the SINDy-SA approach with respect to scenarios with noisy or sparse data. For all applications and keeping the other setups fixed, we firstly carry out experiments reducing the number of evenly distributed time points ($m$) up to the value for which the SINDy-SA framework is able to
Table 7 Model selection results for the SIR application. Subscripts $w_i$ indicate the model selection criterion weights. Models selected as the best ones are indicated by italics.

| Method  | $Model$ | $d$ | $\text{SSE}(X, \hat{X})$ | AIC | AIC$_c$ | BIC | AIC $w_j$ | AIC$_c$ $w_j$ | BIC $w_j$ |
|---------|---------|----|--------------------------|-----|--------|------|-----------|-------------|-----------|
| SINDy-SA           | 1       | 3  | 0.861                    | −2450.414 | −2450.353 | −2438.440 | 1.000 | 1.000 | 1.000 |
|                    | 2       | 8  | 29.585                   | −1025.674 | −1025.305 | −993.742  | 0.000 | 0.000 | 0.000 |
|                    | 3       | 8  | 28.116                   | −1046.044 | −1045.676 | −1014.113 | 0.000 | 0.000 | 0.000 |
|                    | 4       | 16 | 2.799                    | −1952.874 | −1951.453 | −1889.010 | 0.000 | 0.000 | 0.000 |
|                    | 5       | 19 | 2.201                    | −2043.051 | −2041.051 | −1967.214 | 0.000 | 0.000 | 0.000 |
|                    | 6       | 29 | 2.318                    | −2002.284 | −1997.582 | −1886.532 | 0.000 | 0.000 | 0.000 |
|                    | 7       | 35 | 24577261.088             | 4480.347  | 4487.270  | 4620.048  | 0.000 | 0.000 | 0.000 |
|                    | 8       | 41 | 23109275.001             | 4467.712  | 4477.332  | 4631.362  | 0.000 | 0.000 | 0.000 |
|                    | 9       | 37 | 23261656.824             | 4462.341  | 4470.109  | 4610.025  | 0.000 | 0.000 | 0.000 |
|                    | 10      | 40 | 23261940.744             | 4468.346  | 4477.482  | 4628.004  | 0.000 | 0.000 | 0.000 |
| SINDy             | 1       | 8  | 0.018                    | −3987.467 | −3987.098 | −3955.535 | 0.000 | 0.000 | 0.000 |
|                    | 2       | 5  | $1.144 \times 10^{-7}$   | −8780.034 | −8779.881 | −8760.076 | 9.049 $\times 10^{-105}$ | 8.821 $\times 10^{-105}$ | 1.230 $\times 10^{-105}$ |
|                    | 3       | 5  | 52532578.252             | 4724.192  | 4724.344  | 4744.149  | 0.000 | 0.000 | 0.000 |
|                    | 4       | 3  | 21898540.949             | 4370.186  | 4370.247  | 4382.161  | 0.000 | 0.000 | 0.000 |
|                    | 5       | 6  | $1.540 \times 10^{-4}$   | −5895.933 | −5895.719 | −5871.984 | 0.000 | 0.000 | 0.000 |
|                    | 6       | 4  | $3.717 \times 10^{-6}$   | −7389.671 | −7389.569 | −7373.705 | 0.000 | 0.000 | 0.000 |
|                    | 7       | 1  | 254775423.290            | 5347.771  | 5347.781  | 5351.763  | 0.000 | 0.000 | 0.000 |
|                    | 8       | 4  | $3.470 \times 10^{-8}$   | −9259.171 | −9259.070 | −9243.205 | 1.000 | 1.000 | 1.000 |
|                    | 9       | 5  | $1.049 \times 10^{-6}$   | −7893.628 | −7893.475 | −7873.670 | 2.992 $\times 10^{-297}$ | 2.916 $\times 10^{-297}$ | 4.066 $\times 10^{-298}$ |
|                    | 10      | 5  | 35834684.742             | 4571.185  | 4571.337  | 4591.142  | 0.000 | 0.000 | 0.000 |
|                    | 11      | 3  | 219088760.235            | 5291.409  | 5291.470  | 5303.384  | 0.000 | 0.000 | 0.000 |
|                    | 12      | 5  | 4855374087.697           | 6534.755  | 6534.907  | 6554.712  | 0.000 | 0.000 | 0.000 |
|                    | 13      | 2  | 254775423.002            | 5349.771  | 5349.802  | 5357.754  | 0.000 | 0.000 | 0.000 |
|                    | 14      | 2  | 208797458.581            | 5270.164  | 5270.195  | 5278.147  | 0.000 | 0.000 | 0.000 |
SINDy-SA framework: enhancing nonlinear system

Fig. 7 Comparison between the observed data $X$ and the numerical solution $\hat{X}$ of the selected best SIR models, shown inside the graphs. Although the dynamics are quite similar, the model in (b) includes one term (shown in gray) that is not present in the true model.

Table 8 Best models identified by the SINDy-SA method for the applications considered in this work, whose structures correspond to the true models. Results were obtained using the lowest number of data points ($m$) and the highest noise intensities.

| Application       | $m$ | Noise                  | Best identified model                                      |
|-------------------|-----|------------------------|-----------------------------------------------------------|
| Prey-predator     | 70  | Multiplicative Log-normal(0, 0.06) |
| Tumor growth      | 7   | Multiplicative Log-normal(0, 1.0)   | $\dot{x} = 0.926x - 0.920xy$ \$
| Pendulum motion   | 70  | Additive $\mathcal{N}(0, 0.01)$ | $\dot{x} = 0.034x - 4.586 \times 10^{-6}x^2$ |
| Compartmental     | 22  | Multiplicative Log-normal(0, 0.001) | $\dot{S} = -4.004 \times 10^{-4}SI$ \$

$$I = 3.999 \times 10^{-4}SI - 0.040I$$
to correctly identify the true mathematical model. We then perturb the prey–predator, tumor growth, and SIR compartmental data points with a multiplicative noise following log-normal distributions, whereas we contaminate the data of the pendulum motion model with a normal additive noise. We gradually increase the noise level up to the value from which correct identification is prevented. Table 8 summarizes the lowest values for $m$ and the highest noise intensities for all applications, and the corresponding best identified models, while Fig. 8 compares their simulations against the noisy and limited data. Although the parameter values of the identified models are slightly different from the parameters of the true models, the SINDy-SA framework was quite robust to handle noisy and limited data.

4 Final remarks

In this work, we built on the original SINDy method by integrating it with a global SA technique of model parameters. The SINDy method has recently been used to identify nonlinear dynamical system structures from noisy measurement data. However, this method depends on choosing a threshold under which regression coefficients are eliminated in the process of identifying the model structure. The SA technique circumvents the need to define a threshold value by ranking terms according to their importance in relation to the rate of change in time of each state variable (QoI) and eliminating those less influential.

In our proposed SINDy-SA approach, we employed the EE method to analyze the sensitivity of the parameters, and the defined QoI is associated with the formulation of a sparse regression problem. We chose the EE method since it allows ranking the global influence of the model parameters on a QoI and requires a relatively small number of model evaluations compared to variance-based methods. We remark that the proposed methodology is not dependent on the EE method and other global SA techniques can also be used. For solving the sparse regression problem, we used the ridge regression method, also implemented in the PySINDy.
SINDy-SA framework: enhancing nonlinear system

module, due to its robustness and accuracy in identifying nonlinear dynamical systems compared with other classical regression methods. Our SINDy-SA method is carried out through an iterative process, whose convergence is dictated by statistics of the dynamic change of previously computed errors.

We integrated the proposed SINDy-SA method in a general framework, in which we formulated different experimental settings yielding a set of resulting identified models, whose parameter estimates were then improved through model recalibration. Among all possible identified models, the best and most parsimonious model was selected using model selection techniques based on information theory. The general framework can also employ other system identification methods, including the original SINDy approach, as performed here for comparison. Both SINDy-SA and SINDy frameworks were executed in a variety of applications, in which we generated simulated data from true models in order to analyze the ability of the methods to correctly discover the structure of the mathematical model, as well as the parameter values. The applications presented include models with different behaviors encompassing a prey–predator model, a logistic model calibrated from tumor growth data, a pendulum motion model, and a SIR compartmental model. While the SINDy framework failed to identify the logistic and the SIR models, the SINDy-SA framework was able to correctly identify the true model for all applications, emphasizing a relevant advantage of performing a SA of model parameters to solve the system identification problem. These facts demonstrate the potential of our SINDy-SA method to be used in real applications, in which the true model is not known a priori, in the search for interpretable and predictive mathematical models.

In the applications presented, we defined the library of candidate functions composed of the terms present in the true model and some additional terms that we expected to be eliminated by the system identification methods. This was assumed in order to investigate the ability of the SINDy-SA and SINDy methods to discover known mathematical models. However, in a realistic scenario, the true model is generally not known. In this context, an important question arises, associated with the behavior of the methods when some term of the true model is not part of the library of candidate functions. In such case, both SINDy-SA and the original SINDy approaches would try to find the best possible combination of candidate functions that best fits the experimental data. This emphasizes the crucial role the modeler has in system identification problems. The modeler must define the set of possible functions based on the available data and prior knowledge about the expected behavior of the system, as well as assess whether the identified mathematical model is physically interpretable and sufficiently accurate.

The time and state variables to perform a global sensitivity analysis can play a crucial role in model identification. For problems with multiple dependent outputs, Saltelli et al. [30] suggest composing multiple outputs into a single scalar and performing a sensitivity analysis. However, in some situations, it is more appropriate to quantify the sensitivity of the parameters at each output [19]. Although this analysis was not carried out in the present work, applying multi-output global sensitivity analysis methods in our framework is the target of future investigations.

The model recalibration step of our proposed framework is critical for correctly identifying the true dynamical system. Although we have successfully employed the Levenberg-Marquardt method, other parameter inference techniques can be used alternatively, such as Bayesian calibration procedures, through which parameter uncertainties can be quantified. This is particularly important when dealing with limited and noisy data. In this context, data preprocessing techniques, such as the Gaussian process regression [18], could also provide robustness on system identification, a topic for further investigation.

Overall, our SINDy-SA framework can leverage new developments together with other recent SINDy works to deal with, for example, implicit dynamical systems [16], partial differential equations [29], and control [3].

Funding The authors would like to thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES). Gustavo T. Naozuka is supported by a PhD scholarship from Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ), grant number 200.325/2020. Heber L. Rocha is supported by a postdoctoral fellowship from U01-CA232137 (NCI), 1720625 (NSF), 1818187 (NSF), the Breast Cancer Research Foundation, and the Jayne Koskinas Ted Giovanis Foundation for Health and Policy.
Availability of Data and Material  The data used in the current study are generated by running the source code publicly available at https://github.com/tmglncc/SINDy-SA.

Declarations

Conflict of interest  The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript.

Code Availability  The source code used to generate the results during the current study is publicly available at https://github.com/tmglncc/SINDy-SA.

References

1. Boninsegna, L., Nüske, F., Clementi, C.: Sparse learning of stochastic dynamical equations. J. Chem. Phys. 148(24), 241723 (2018). https://doi.org/10.1063/1.5018409
2. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. 113(15), 3932–3937 (2016). https://doi.org/10.1073/pnas.1517384113
3. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Sparse identification of nonlinear dynamics with control (SINDyC). IFAC-PapersOnLine 49(18), 710–715 (2016). https://doi.org/10.1016/j.ifacol.2016.10.249. 10th IFAC Symposium on Nonlinear Control Systems NOLCOS 2016
4. Burnham, K.P., Anderson, D.R.: Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach, 2nd edn. Springer, New York (2002)
5. Campolongo, F., Tarantola, S., Saltelli, A.: Tackling quantitatively large dimensionality problems. Comput. Phys. Commun. 117(1–2), 75–85 (1999). https://doi.org/10.1016/S0010-4655(98)00165-9
6. Cortiella, A., Park, K.C., Doostan, A.: Sparse identification of nonlinear dynamical systems via reweighted ℓ1-regularized least squares. Comput. Methods Appl. Mech. Eng. 376, 113620 (2021). https://doi.org/10.1016/j.cma.2020.113620
7. França, T., Braga, A.M.B., Ayala, H.V.H.: Feature engineering to cope with noisy data in sparse identification. Expert Syst. Appl. 188, 115995 (2022). https://doi.org/10.1016/j.eswa.2021.115995
8. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer series in statistics. Springer (2009). https://books.google.com.br/books?id=eBSgoAEACAAJ
9. Herman, J., Usher, W.: SALib: an open-source Python library for sensitivity analysis. J. Open Source Softw. (2017). https://doi.org/10.21105/joss.00097
10. Hirsh, S.M.: Dimensionality reduction and sparsity promotion for complex dynamical systems. PhD thesis, University of Washington (2020)
11. Hoffmann, M., Fröhner, C., Noé, F.: Reactive SINDy: discovering governing reactions from concentration data. J. Chem. Phys. 150(2), 025101 (2019). https://doi.org/10.1063/1.5066099
12. Horrocks, J., Bauch, C.T.: Algorithmic discovery of dynamic models from infectious disease data. Sci. Rep. 10(1), 7061 (2020). https://doi.org/10.1038/s41598-020-63877-w
13. Horrocks, J.H.: Sparse identification of epidemiological models from empirical data. Master’s thesis, University of Waterloo, Waterloo, Canada (2018)
14. James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning: With Applications in R. Springer Publishing Company, Incorporated (2014)
15. Jiang, Y.X., Xiong, X., Zhang, S., Wang, J.X., Li, J.C., Du, L.: Modeling and prediction of the transmission dynamics of COVID-19 based on the SINDy-LM method. Nonlinear Dyn. 105(3), 2775–2794 (2021). https://doi.org/10.1007/s11071-021-06707-6
16. Kaheman, K., Kutz, J.N., Brunton, S.L.: SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. Proc. R. Soc. A Math. Phys. Eng. Sci. 476(2242), 20200279 (2020). https://doi.org/10.1098/rspa.2020.0279
17. Liang, J., Zhang, X., Wang, K., Tang, M., Tian, M.: Discovering dynamic models of COVID-19 transmission. Transbound. Emerg. Dis. (2021). https://doi.org/10.1111/tbed.14263
18. Libotte, G.B., Anjos, L.D., Almeida, R.C.C., Malta, S.M.C., Silva, R.S.: Framework for enhancing the estimation of model parameters for data with a high level of uncertainty. Nonlinear Dyn. 7, 1–18 (2022). https://doi.org/10.1007/s11071-021-07069-9
19. Liu, F., Wei, P., Tang, C., Wang, P., Yue, Z.: Global sensitivity analysis for multivariate outputs based on multiple response gaussian process model. Reliab. Eng. Syst. Safe. 189, 287–298 (2019). https://doi.org/10.1016/j.ress.2019.04.039
20. Maddu, S., Cheeseman, B.L., Müller, C.L., Sbalzarini, I.F.: Learning physically consistent differential equation models from data using group sparsity. Phys. Rev. E 103, 042310 (2021). https://doi.org/10.1103/PhysRevE.103.042310
21. Mangan, N.M., Brunton, S.L., Proctor, J.L., Kutz, J.N.: Inferring biological networks by sparse identification of nonlinear dynamics. IEEE Trans. Molec. Biol. Multi-Scale Commun. 2(1), 52–63 (2016). https://doi.org/10.1109/TMBMC.2016.2633265
22. Mangan, N.M., Kutz, J.N., Brunton, S.L., Proctor, J.L.: Model selection for dynamical systems via sparse regression and information criteria. Proc. R. Soc. A Math. Phys. Eng. Sci. 473(2204), 20170009 (2017). https://doi.org/10.1098/rspa.2017.0009
23. Matos Silva Lopes, D., Cunha, Jr. A.: On the physical consistency of evolution laws obtained with sparse regression. In: Second International Nonlinear Dynamics Conference, Rome, Italy, (2021). https://hal.archives-ouvertes.fr/hal-03130833
24. Naozuka, G., Paixão, E., Silva, J.V., Menezes, M., Almeida, R.: Model comparison and uncertainty quantification in tumor growth. Trends Comput. Appl. Math. 22(3), 495–514 (2021)
25. Niven, R.K., Mohammad-Djafari, A., Cordier, L., Abel, M., Quade, M.: Bayesian identification of dynam-
SINDy-SA framework: enhancing nonlinear system

cal systems. Proceedings (2019). https://doi.org/10.3390/proceedings201903303
26. Niven, R.K., Ali, M.D., Cordier, L., Abel, M., Quade, M.: Dynamical system identification by Bayesian inference. In: 22nd Australasian Fluid Mechanics Conference AFMC2020, Brisbane, Australia (2020). https://doi.org/10.14264/692fcb8
27. Petzold, L.: Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. SIAM J. Sci. Stat. Comput. 4(1), 136–148 (1983). https://doi.org/10.1137/0904010
28. Quade, M., Abel, M., Nathan Kutz, J., Brunton, S.L.: Sparse identification of nonlinear dynamics for rapid model recovery. Chaos 28(6), 063116 (2018). https://doi.org/10.1063/1.5027470
29. Rudy, S.H., Brunton, S.L., Proctor, J.L., Kutz, J.N.: Data-driven discovery of partial differential equations. Sci. Adv. 3(4), e1602614 (2017). https://doi.org/10.1126/sciadv.1602614
30. Saltelli, A., Tarantola, S., Campolongo, F.: Sensitivity analysis as an ingredient of modeling. Stat. Sci. (2000). https://doi.org/10.1214/ss/1009213004
31. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S.: Global Sensitivity Analysis: The Primer. Wiley (2008). https://doi.org/10.1002/9780470725184
32. de Silva, B.M., Champion, K., Quade, M., Loiseau, J.C., Kutz, J.N., Brunton, S.L.: PySINDy: a Python package for the sparse identification of nonlinear dynamical systems from data. J. Open Source Softw. 5(49), 2104 (2020). https://doi.org/10.21105/joss.02104
33. Subramanian, H.: Combining scientific computing and machine learning techniques to model longitudinal outcomes in clinical trials. Master’s thesis, Linköping University, Department of Computer and Information Science (2021)
34. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, I., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 10 Contributors: SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods 17, 261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2
35. Wang, Z., Huan, X., Garikipati, K.: Variational system identification of the partial differential equations governing the physics of pattern-formation: Inference under varying fidelity and noise. Comput. Methods Appl. Mech. Eng. 356, 44–74 (2019). https://doi.org/10.1016/j.cma.2019.07.007
36. Wang, Z., Zhang, X., Teichert, G.H., Carrasco-Teja, M., Garikipati, K.: System inference for the spatio-temporal evolution of infectious diseases: Michigan in the time of COVID-19. Comput. Mech. 66(5), 1153–1176 (2020). https://doi.org/10.1007/s00466-020-01894-2
37. Yang, Y., Aziz Bhouri, M., Perdikaris, P.: Bayesian differential programming for robust systems identification under uncertainty. Proc. R. Soc. A Math. Phys. Eng. Sci. 476(2243), 20200290 (2020). https://doi.org/10.1098/rspa.2020.0290

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.