INTERFACE

royalsocietypublishing.org/rsif

CC BY

Research

Cite this article: Chakraborty AK, Wang H, Ramazi P. 2026 From data-driven differential equations to mechanistic forecasting with machine learning. *J. R. Soc. Interface* **23**: 20260240.
https://doi.org/10.1098/rsif.2026.0240

Received: 4 March 2026
Accepted: 1 September 2026

Subject Category:
Life Sciences–Mathematics interface

Subject Areas:
biomathematics, environmental science

Keywords:
data-driven dynamical systems, time-varying parameters, sparse regression, mechanistic forecasting, finite-horizon forecast error bounds, machine learning

Author for correspondence:
Hao Wang
e-mail: hao8@ualberta.ca

Supplementary material is available online at https://doi.org/10.6084/m9.figshare.c.8705787.

THE ROYAL SOCIETY PUBLISHING

From data-driven differential equations to mechanistic forecasting with machine learning

Amit K. Chakraborty[1], Hao Wang[1] and Pouria Ramazi[2]

[1]Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, Alberta, Canada
[2]Department of Biological Sciences and Department of Mathematics and Statistics, University of Calgary, Calgary, Alberta, Canada

AKC, 0000-0002-7960-3984; HW, 0000-0002-4132-6109; PR, 0000-0003-4906-0090

The equations of complex dynamical systems may not be identified by expert knowledge, especially if the underlying mechanisms are unknown. Data-driven discovery methods address this challenge by inferring governing equations from time-series data using a library of functions constructed from the measured variables. However, these methods typically assume time-invariant coefficients, which limits their ability to capture evolving system dynamics. To overcome this limitation, we allow some of the parameters to vary over time, learn their temporal evolution directly from data and infer a system of equations that incorporates both constant and time-varying parameters. We then transform this framework into a forecasting model by predicting the time-varying parameters and substituting these predictions into the learned equations. The model is validated on simulated susceptible–infected–recovered and consumer–resource systems, as well as on real-world greenhouse gas concentration and cyanobacteria cell count datasets. By dynamically adapting to temporal shifts, our proposed model achieved a mean absolute error below 3% for learning a time series and below 6% for forecasting up to a month ahead. We additionally compare forecasting performance against convolutional neural network–long short-term memory and gradient boosting machine, and show that our model outperforms these methods across most datasets. Our findings demonstrate that integrating time-varying parameters into data-driven discovery of differential equations improves both modelling accuracy and forecasting performance.

1. Introduction

Mechanistic models provide the causal and interpretable governing laws of a system, but many natural systems are too complex to be captured by expert knowledge [1–3]. For example, emissions of greenhouse gases, such as methane and carbon dioxide, are influenced by temperature, humidity, wind speed and other weather variables [4,5]. In epidemiology, disease incidence is shaped by environmental factors, population density, social behaviour and policy interventions [6–9]. The relationships among the variables in such a high-dimensional system are many, time variant and unknown, making it nearly infeasible to be described by equations that are based on expert knowledge [10].

A promising alternative is data-driven discovery of the governing equation, often represented as differential equations [11–14]. Unlike mechanistic modelling that relies on predefined assumptions, data-driven approaches adapt to the intrinsic structure of the system, capturing both linear and nonlinear interactions among the system's variables [13].

© 2026 The Authors. Published by the Royal Society under the terms of the Creative Commons Attribution License http://creativecommons.org/licenses/by/4.0/, which permits unrestricted use, provided the original author and source are credited.

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

A widely used technique for the data-driven discovery of governing equations is the sparse identification of nonlinear dynamics (SINDy) [12]. SINDy and its variants have been used to uncover dynamical systems of ordinary differential equations (ODEs) and partial differential equations in fluid dynamics [12,15–19], biological systems [20–27], mechanical systems [28–30] and climate and environmental modelling [31–33]. It uses sparse regression to identify the minimal set of candidate functions needed to describe a system's dynamics [12]. Originally developed for systems with constant coefficients, SINDy has since been extended to handle time-varying parameters [14,34]. Such extensions are needed for systems with evolving dynamics, where temporal changes cannot be accurately represented by fixed parameters [8]. For example, the transmission rate and the death rate in an epidemic model are time-varying and can be estimated by calibrating the model to epidemic data [8,9,35].

However, for systems with many variables, it remains unclear how many parameters exactly need to be time-varying to capture the data accurately. Identifying only the necessary time-varying parameters can reduce computational cost, prevent overfitting and improve interpretability.

Another challenge is to use the resulting system of equations equipped with time-varying parameters in forecasting. Finding the time series of a time-varying parameter (basically a variable) over past data can be done by model fitting. However, forecasting requires running the model with an initial condition, and this requires the time series of the parameter in the future [8]. This is shaped by external forces, interactions among variables and other dynamic factors [8,9]. To address this challenge, machine learning (ML) models were employed to forecast time-varying parameters across various fields using their historical time series and known covariates [35]. In epidemiology, the disease transmission and death rates were forecasted based on policies or weather using epidemic models [8,9,36–38]. In climate science, greenhouse gas drivers were studied by forecasting time-varying parameters [39]. Therefore, once a system is modelled with time-varying parameters, they can be forecasted via an ML model using their past values and covariates whose future values are known [35,40].

The key contribution of this study is a unified mechanistic forecasting framework that connects data-driven discovery of time-varying dynamics with ML-based parameter forecasting. Unlike approaches that allow all coefficients to vary, our framework identifies a parsimonious subset of active coefficients as time-varying while retaining the remaining coefficients as constant, and uses forecasts of the selected coefficients to generate future trajectories from the learned ODEs. We implement this framework using a two-stage application of the SINDy algorithm. In the first stage, SINDy is applied to the full time series to identify the variables that actively contribute to the dynamics and their interactions under constant parameter assumptions. In the second stage, the analysis is repeated on shorter temporal intervals, allowing a subset of parameters—selected based on validation performance—to vary over time, while the remaining parameters are held fixed. This procedure yields an ODE model that preserves the underlying mechanistic structure while accommodating temporal variability in key processes. To enable forecasting, the identified time-varying parameters are predicted using a ML model driven by external covariates that influence the time-varying parameters. These predicted parameters are then substituted into the learned ODEs, allowing one to run the model and make forecasts that adapt dynamically to changing conditions.

We applied this methodology to two well-known mathematical models—the susceptible–infected–recovered (SIR) model [41] and the consumer–resource (CR) model [42]—as well as two real-world datasets: greenhouse gas concentration dynamics in Alberta's oil sands tailings ponds and cyanobacteria cell counts in lakes across Alberta. We predicted the time-varying parameters using a random forest (RF) model informed by weather-related drivers, including air temperature, relative humidity, precipitation and wind speed. To benchmark the forecasting performance of our proposed approach, we compared it against two widely used data-driven baselines: a hybrid convolutional neural network–long short-term memory (CNN–LSTM) model and a gradient boosting machine (GBM).

2. Methods

2.1. Problem formulation

Consider the evolution of a state vector $\mathbf{x}(t) \in \mathbb{R}^n$ for a natural number $n \in \mathbb{N}$, described by the dynamical system

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \tag{2.1}$$

where $\mathbf{u}(t) \in \mathbb{R}^q$, $q \in \mathbb{N}$ denotes the input variables and $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}^n$ is an unknown function. Measurements of the state variable are available at m time points $\mathbf{t} = [t_1, \ldots, t_m]$, resulting in the time-series data

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(t_1)^\top \\ \mathbf{x}(t_2)^\top \\ \vdots \\ \mathbf{x}(t_m)^\top \end{bmatrix}, \tag{2.2}$$

where m is the number of measurements and the resulting matrix has dimension $m \times n$. The corresponding time derivative data are

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}(t_1)^\top \\ \dot{\mathbf{x}}(t_2)^\top \\ \vdots \\ \dot{\mathbf{x}}(t_m)^\top \end{bmatrix}. \tag{2.3}$$

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

Similarly, the time-series data of the inputs are

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}(t_1)^\top \\ \mathbf{u}(t_2)^\top \\ \vdots \\ \mathbf{u}(t_m)^\top \end{bmatrix}.$$

The library of candidate functions to estimate $\mathbf{f}(\cdot)$ is constructed from $\mathbf{X}$ and $\mathbf{U}$ as

$$\boldsymbol{\Theta}(\mathbf{X}, \mathbf{U}) = \left[\mathbf{1},\ \mathbf{X},\ \mathbf{U},\ (\mathbf{X} \odot \mathbf{U}),\ \mathbf{X}^2,\ \mathbf{U}^2,\ \ldots,\ (\mathbf{X}^{d-1} \odot \mathbf{U}),\ (\mathbf{X}^{d-2} \odot \mathbf{U}^2),\ \ldots,\ (\mathbf{X} \odot \mathbf{U}^{d-1}),\ \mathbf{X}^d,\ \mathbf{U}^d\right], \tag{2.4}$$

where $d \in \mathbb{N}$, and the candidate library contains all monomials in $\mathbf{x}(\mathbf{t})$ and $\mathbf{u}(\mathbf{t})$ of total degree at most d. In addition to polynomial terms, the candidate function library may also include nonlinear bases such as $\sin(\cdot)$, $\cos(\cdot)$, $\tan(\cdot)$, their powers and other user-defined nonlinear functions.

Based on equation (2.1), the derivatives $\dot{\mathbf{X}}$ and the candidate function library $\boldsymbol{\Theta}(\mathbf{X}, \mathbf{U})$ can be related through

$$\dot{\mathbf{X}} = \boldsymbol{\Theta}(\mathbf{X}, \mathbf{U})\, \boldsymbol{\Xi}, \tag{2.5}$$

where

$$\boldsymbol{\Xi} = \begin{bmatrix} \boldsymbol{\xi}_1 & \boldsymbol{\xi}_2 & \cdots & \boldsymbol{\xi}_n \end{bmatrix}$$

is the matrix of unknown coefficient vectors $\boldsymbol{\xi}_k$. For a system with n variables and a candidate library $\boldsymbol{\Theta}(\cdot)$ of l terms, the coefficient matrix $\boldsymbol{\Xi}$ has dimension $l \times n$. The non-zero entries of $\boldsymbol{\xi}_k$ indicate the active candidate functions contributing to the dynamics of the kth state. Once $\boldsymbol{\xi}_k$ is identified, the nonlinear equation for the kth state is given by

$$\dot{x}_k = \boldsymbol{\Theta}(\mathbf{x}, \mathbf{u})\, \boldsymbol{\xi}_k, \tag{2.6}$$

where x_k is the kth component of $\mathbf{x}$, and $\boldsymbol{\Theta}(\mathbf{x}, \mathbf{u})$ denotes a row vector of symbolic candidate functions evaluated at $\mathbf{x}$ and $\mathbf{u}$.

Example 2.1.1. *Consider $n = 2$ state variables, i.e. $\mathbf{x} = [x_1, x_2]^\top$, and $q = 2$ input variables, i.e. $\mathbf{u} = [u_1, u_2]^\top$, and a library with maximum polynomial degree $d = 2$. If the variables are measured at time points $m = 2$, i.e. $\mathbf{t} = [t_1, t_2]$, then*

$$\mathbf{X}^2 = \left[x_1^2(\mathbf{t}),\ x_1x_2(\mathbf{t}),\ x_2^2(\mathbf{t})\right] \quad \textit{and} \quad \mathbf{U}^2 = \left[u_1^2(\mathbf{t}),\ u_1u_2(\mathbf{t}),\ u_2^2(\mathbf{t})\right],$$

and the candidate library has $l = 15$ terms:

$$\boldsymbol{\Theta}(\mathbf{X}, \mathbf{U}) = [\mathbf{1},\ x_1(\mathbf{t}),\ x_2(\mathbf{t}),\ u_1(\mathbf{t}),\ u_2(\mathbf{t}),\ x_1x_2(\mathbf{t}),\ u_1u_2(\mathbf{t}),\ x_1u_1(\mathbf{t}),\ x_1u_2(\mathbf{t}),\ x_2u_1(\mathbf{t})\ x_2u_2(\mathbf{t}), x_1^2(\mathbf{t}),\ x_2^2(\mathbf{t}),\ u_1^2(\mathbf{t}),\ u_2^2(\mathbf{t})].$$

Suppose the true dynamics are $\dot{x}_1 = 0.6x_1 + 0.2x_2u_1$ and $\dot{x}_2 = -0.4x_2 + 0.1u_2^2$. Equation (2.5) seeks $\boldsymbol{\Xi} \in \mathbb{R}^{15\times 2}$, whose columns are coefficient vectors $\boldsymbol{\xi}_k \in \mathbb{R}^{15\times 1}$. In this example,

$$\boldsymbol{\xi}_1 = [\,0,\ 0.6,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0.2,\ 0,\ \ldots,\ 0\,]^\top \quad \textit{and} \quad \boldsymbol{\xi}_2 = [\,0,\ 0,\ -0.4,\ 0,\ \ldots,\ 0,\ 0.1\,]^\top,$$

where the non-zero entries identify the active library terms.

Equation (2.5) can be solved as a sparse regression problem to determine the active coefficients $\boldsymbol{\Xi}$ of the candidate nonlinearities $\boldsymbol{\Theta}(\mathbf{X}, \mathbf{U})$, and the SINDy algorithm was introduced by Brunton *et al.* [12]. Early variants of SINDy used an L^1 regularization term for the regression, which was applied over the entire time domain to uncover systems of equations with constant coefficients [12,22]. Consequently, Li *et al.* [14] proposed a time-varying variant of SINDy, which uncovers the time-varying coefficients of a dynamical system, and Rudy *et al.* [13,34] implemented ridge regression to incorporate an L^2 regularization term into the loss function.

2.2. Discovering time-varying dynamics

The sequential threshold ridge regression (STRR), which combines ridge regression with a sparsity-enforcing thresholding technique described by Rudy *et al.* [13,34], was applied to discover time-varying coefficients $\boldsymbol{\Xi}$ in equation (2.5), following the framework of Li *et al.* [14].

First, the input variables $\mathbf{u}(t)$ in equation (2.1) were selected based on prior knowledge, and time-series data for all variables ($\mathbf{x}(t)$ and $\mathbf{u}(t)$) were collected (figure 1a). Each variable's time series was rolled over a number of time points to reduce noise. The derivative $\dot{\mathbf{X}}$ of the state variables was then numerically estimated, and the candidate library $\boldsymbol{\Theta}(\mathbf{X}, \mathbf{U})$ was constructed (equation (2.4)).

Active constant coefficients, $\boldsymbol{\xi}_k$ for the kth state of equation (2.5), were extracted over the entire training time domain (figure 1b) using STRR. The STRR method minimizes error with regularization while enforcing sparsity through an iterative two-step optimization. Ridge regression was first applied by minimizing the objective function

$$\arg\min_{\boldsymbol{\xi}_k} \|\dot{\mathbf{X}}_k - \boldsymbol{\Theta}(\mathbf{X}, \mathbf{U})\boldsymbol{\xi}_k\|_2^2 + \lambda\|\boldsymbol{\xi}_k\|_2^2, \tag{2.7}$$

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

where the term $\lambda\|\boldsymbol{\xi}_k\|_2^2$ represents the regularization component that penalizes non-zero coefficients, controlled by the hyperparameter λ, which helps in avoiding overfitting. A sparsity-enforcing thresholding step was then applied, where coefficients smaller than a predefined threshold τ are set to zero:

$$\xi_{i,k} = \begin{cases} 0 & \text{if } |\xi_{i,k}| < \tau \\ \xi_{i,k} & \text{otherwise,} \end{cases} \tag{2.8}$$

where $\xi_{i,k}$ is the ith entry of $\boldsymbol{\xi}_k$. This step removes small, less significant features. Ridge regression was then refitted on the remaining active features (those whose coefficients have not been zeroed out) $\boldsymbol{\Theta}_{\text{active}}(\mathbf{X}, \mathbf{U})$, using the same loss function

$$\arg\min_{\boldsymbol{\xi}_k} \|\dot{\mathbf{X}}_k - \boldsymbol{\Theta}_{\text{active}}(\mathbf{X}, \mathbf{U})\boldsymbol{\xi}_k\|_2^2 + \lambda\|\boldsymbol{\xi}_k\|_2^2. \tag{2.9}$$

This iterative refinement ensures that the model continues to adapt to the most relevant features. The process repeats until the coefficients converge, i.e. the change in the coefficients between two consecutive iterations is smaller than a specified tolerance ϵ:

$$\|\boldsymbol{\xi}_k^{(t)} - \boldsymbol{\xi}_k^{(t-1)}\| < \epsilon, \tag{2.10}$$

where $\boldsymbol{\xi}_k^{(t)}$ and $\boldsymbol{\xi}_k^{(t-1)}$ are the coefficients at iterations t and $t-1$, respectively.

The correlations between the active candidate terms ($\boldsymbol{\Theta}_{\text{active}}(\mathbf{X}, \mathbf{U})$) and the target variable $\dot{\mathbf{X}}_k(\mathbf{t})$ for the kth state were computed (figure 1c). The top N active candidate terms $\boldsymbol{\Theta}_{\text{top N}}$ showing the highest correlation with $\dot{\mathbf{X}}_k(\mathbf{t})$ were then identified from the candidate set ($\boldsymbol{\Theta}_{\text{active}}(\mathbf{X}, \mathbf{U})$). The coefficients of $\boldsymbol{\Theta}_{\text{top N}}$ were selected as time-varying coefficients (referred to as time-varying parameters) (figure 1c), while the remaining coefficients were kept constant. Furthermore, the bias term was designated as time-varying by default to account for baseline shifts between intervals.

To capture the time-varying dynamics of the system (equation (2.1)), a window size w was defined and kept fixed over the entire training period. STRR equations (2.7)–(2.10) were applied across each time window $[t, t+w]$ (figure 1d), with each parameter initialized using its value from the preceding interval. This sequential fitting strategy allowed the time-varying parameters to be dynamically adjusted while less impactful parameters remained fixed, capturing essential changes in the model structure and maintaining continuity between intervals. The discovered expression for the kth state can be written as

$$\dot{x}_k(t) = \xi_{k,0}(t) + \sum_{z_i(t)\in\Theta_{\text{top N}}} \xi_{k,i}(t) z_i(t) + \sum_{z_j(t)\notin\Theta_{\text{top N}}} \bar{\xi}_{k,j} z_j(t), \tag{2.11}$$

where $\boldsymbol{\xi}(\boldsymbol{t})$ are time-varying parameters and $\bar{\boldsymbol{\xi}}$ are the constant coefficients. Here, $z_i(t) \in \boldsymbol{\Theta}_{\text{top N}}$ denotes the top N active candidate terms and $z_j(t) \in \boldsymbol{\Theta}_{\text{active}}(\mathbf{X}, \mathbf{U}) \setminus \boldsymbol{\Theta}_{\text{top N}}$ represents the remaining active terms with fixed coefficients.

The time-varying parameters $\boldsymbol{\xi}(t)$ and the constant coefficients $\bar{\boldsymbol{\xi}}$ are assigned to disjoint sets of active candidate terms. The time-varying parameters are re-estimated over successive temporal windows to capture evolving dynamics, whereas the constant coefficients retain the values estimated from the full training period.

2.3. Forecasting from learned dynamics

To forecast the future dynamics of the state variables, the discovered representation in equation (2.11), which incorporates both fixed and time-varying coefficients, was leveraged. The main idea involves predicting the time-varying parameters, $\boldsymbol{\xi}(\boldsymbol{t})$, while keeping the constant coefficients, $\bar{\boldsymbol{\xi}}$, unchanged. Forecasting is thereby enabled by updating only the components of the model that evolve over time.

An ML approach was employed to model the evolution of the time-varying parameters. During model development, the time-varying parameters $\boldsymbol{\xi}(\boldsymbol{t})$ obtained from STRR were piecewise constant over each window of length w. However, during forecasting, the predicted parameters were not constrained to remain constant within a window; instead, continuous predictions were produced by the ML model based on the input features. Relevant predictors were used to train the models, with each time-varying parameter designated as a target output (figure 1e). By combining the forecasted time-varying parameters with the fixed coefficients and the active candidate terms, the right-hand side of the dynamical model (2.11) was updated, and the forecast of the kth state variable was obtained by

$$\dot{\hat{x}}_k(t) = \hat{\bar{\xi}}_{k,0}(t) + \sum_{z_i(t)\in\Theta_{\text{top N}}} \hat{\bar{\xi}}_{k,i}(t)\, z_i(t) + \sum_{z_j(t)\notin\Theta_{\text{top N}}} \bar{\xi}_{k,j}\, z_j(t), \tag{2.12}$$

where $\hat{\boldsymbol{\xi}}(\boldsymbol{t})$ denotes the forecasted time-varying parameters and $\bar{\boldsymbol{\xi}}$ are the constant coefficients. Here, $z_i(t) \in \boldsymbol{\Theta}_{\text{top N}}$ denotes the top N active candidate terms, and $z_j(t) \in \boldsymbol{\Theta}_{\text{active}}(\mathbf{X}, \mathbf{U}) \setminus \boldsymbol{\Theta}_{\text{top N}}$ represents the remaining active terms with fixed coefficients.

3. Case studies

We applied the proposed method (§2) to four datasets: two simulated from known ODE systems and two empirical. The simulated datasets were generated from (i) the SIR model and (ii) a CR model. The empirical datasets consisted of (i) carbon dioxide (CO_2)

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

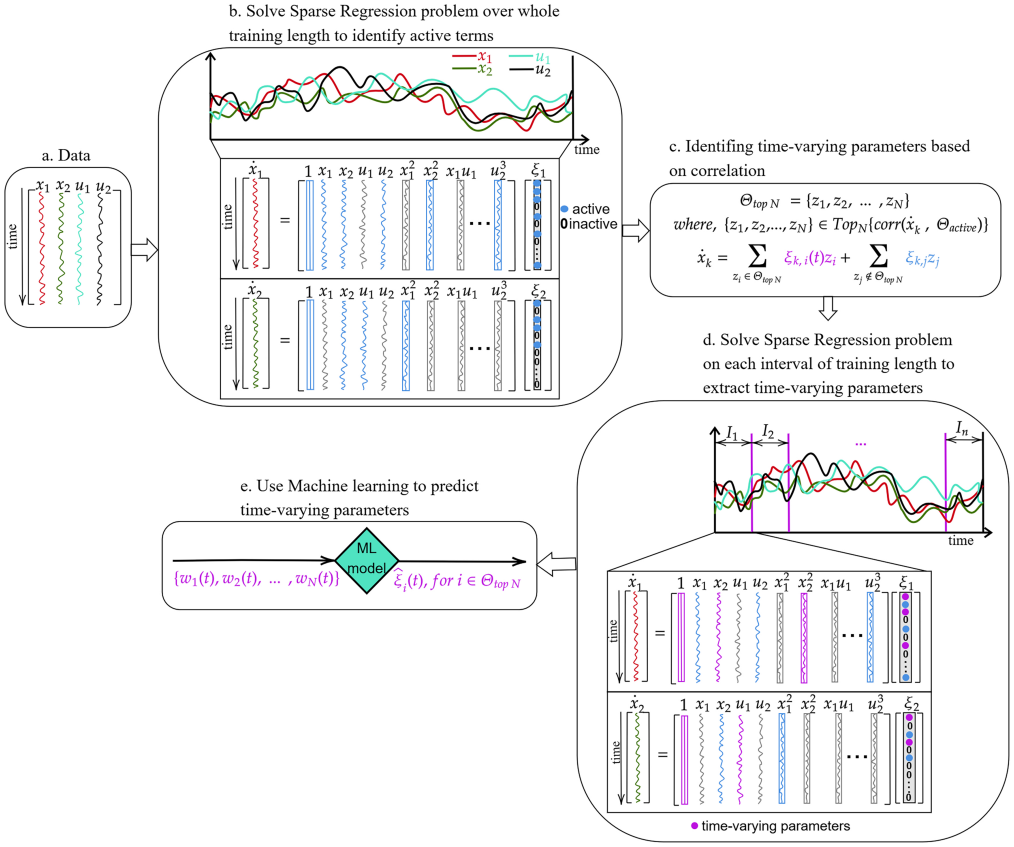


Figure 1. Schematic of data-driven discovery of governing equations and forecasting from the learned system. (a) Time-series data of variables are gathered from natural or simulated systems. (b) A sparse regression problem is solved over the whole time series to identify active terms and constant coefficients. Variables in grey represent non-active terms, while variables in blue represent active terms with constant coefficients. Variables in boxes represent generated terms. (c) Top N candidate terms are chosen based on the correlation of active candidate terms with the derivative of the state variables. The coefficients of these terms are treated as time-varying parameters. By default, the bias term is also considered time-varying. (d) Time-series data are split into intervals, and sparse regression is performed within each interval to determine the time-varying parameters of the top candidate terms. Purple represents top candidate terms and their time-varying parameters, while blue indicates terms with constant coefficients. (e) An ML model is employed to forecast the time-varying parameters. Inputs are relevant predictors, and outputs are the time-varying parameters.

and methane (CH_4) concentrations measured in Alberta's oil sands tailings ponds and (ii) cyanobacteria cell counts collected from lakes across Alberta.

3.1. Susceptible–infected–recovered model

The SIR model [41] is a foundation for most disease-spreading models. In this model, the population is categorized into three compartments: susceptible (S), infectious (I) and recovered (R). The susceptible compartment represents individuals who are not yet infected but are at risk of contracting the disease through contact with infectious individuals. The infectious compartment consists of individuals who are actively carrying the disease and can transmit it to others. Lastly, the recovered compartment includes individuals who have recovered from the disease and are assumed to have gained immunity, no longer participating in disease transmission. These compartments interact dynamically, with transitions governed by a system of ODEs that model the rates of change in each group over time. The model equations are given by

$$\begin{cases} \frac{\mathrm{d}S(t)}{\mathrm{d}t} &= \Lambda - \beta(t)S(t)I(t) - \mu S(t), \\ \frac{\mathrm{d}I(t)}{\mathrm{d}t} &= \beta(t)S(t)I(t) - \alpha I(t) - \mu I(t), \\ \frac{\mathrm{d}R(t)}{\mathrm{d}t} &= \alpha I(t) - \mu R(t), \end{cases} \tag{3.1}$$

where Λ is the recruitment rate of susceptible populations, $\beta(t)$ is the time-dependent disease transmission rate, μ is the natural death rate and α is the recovery rate from the disease. The first two equations of equation (3.1) are independent of recovered (R), and the reduced equations are given by

$$\begin{cases} \frac{\mathrm{d}S(t)}{\mathrm{d}t} &= \Lambda - \beta(t)S(t)I(t) - \mu S(t), \\ \frac{\mathrm{d}I(t)}{\mathrm{d}t} &= \beta(t)S(t)I(t) - \alpha I(t) - \mu I(t). \end{cases} \tag{3.2}$$

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

When white noise is added to the equations of the SIR model (3.2), the SIR model with additive white noise has the form [40]

$$\begin{cases} \mathrm{d}S(t) = \Lambda \mathrm{d}t - \beta(t)S(t)I(t)\mathrm{d}t - \mu S(t)\mathrm{d}t + \sigma_1 \mathrm{d}W_1(t), \\ \mathrm{d}I(t) = \beta(t)S(t)I(t)\mathrm{d}t - \alpha I(t)\mathrm{d}t - \mu I(t)\mathrm{d}t + \sigma_2 \mathrm{d}W_2(t), \end{cases} \tag{3.3}$$

where σ_i, $i = 1, 2$, are the intensities of white noise, and $W_i(t)$, $i = 1, 2$, are independent Wiener processes.

We used equation (3.3) to simulate the time series of susceptible and infected populations over 1500 days. The model parameters were set as $\Lambda = 100$, $\alpha = 0.1$ and $\mu = 0.1$, with initial conditions $S_0 = 500$ and $I_0 = 7$. The transmission rate $\beta(t)$ was defined as a combination of sinusoidal functions to capture irregular seasonal peaks by

$$\beta(t) = A + A_1 \sin(\omega_1 t) + A_2 \sin(\omega_2 t + \pi/4) + A_3 \sin(\omega_3 t + \pi/6)\sin(0.005t), \tag{3.4}$$

where $A = 0.0003$, $A_1 = 0.00\,015$, $A_2 = 0.00\,012$ and $A_3 = 0.00\,010$, and the angular frequencies are $\omega_1 = 2\pi/(365/2)$, $\omega_2 = 2\pi/365$ and $\omega_3 = 2\pi/(365/2)$. Small random fluctuations were added to $\beta(t)$ at each time step to account for irregular environmental variability, and the values were clipped to remain non-negative. Additive stochasticity was introduced through Wiener processes, $\mathrm{d}W_i \sim \mathcal{N}(0, \sqrt{\mathrm{d}t})$, $i = 1, 2$, and the Euler–Maruyama method [43] was used for numerical integration with $\mathrm{d}t = 0.01$. Simulations were repeated with noise intensities $\sigma_1 = \sigma_2 = 0, 0.1, 0.2, 0.3$ and 0.4 to examine the effect of increasing stochasticity on model predictions.

The simulated trajectories of $S(t)$ and $I(t)$ were then used to learn the dynamics of the SIR model using time-varying parameters following the proposed method in §2. To discover the governing equations, we expressed the system in the form

$$\begin{cases} \frac{\mathrm{d}S}{\mathrm{d}t} = f(S, I), \\ \frac{\mathrm{d}I}{\mathrm{d}t} = f(S, I), \end{cases} \tag{3.5}$$

where the time-varying parameters were systematically selected according to the procedure described in §2.

3.2. Consumer–resource model

The CR model [42] is a fundamental framework used to study the interactions between two populations: a resource population (e.g. prey or a nutrient source) and a consumer population (e.g. predators or organisms that depend on the resource). This model captures the dynamics of population growth and interactions, providing insights into ecological systems.

In this model, the resource population (R) grows at a rate proportional to its size, governed by the growth rate r. The growth of R is constrained by consumption from the consumer population (C), which follows a saturating functional response. This response is characterized by the consumption rate a and the half-saturation constant b. Simultaneously, the consumer population (C) grows through resource consumption but experiences a constant mortality rate m. The dynamics of the model are described by the following equations:

$$\begin{cases} \frac{\mathrm{d}R}{\mathrm{d}t} = rR - \frac{aRC}{1+bR}, \\ \frac{\mathrm{d}C}{\mathrm{d}t} = \frac{aRC}{1+bR} - mC. \end{cases} \tag{3.6}$$

To incorporate stochastic fluctuations into the model, we added noise into each of the equations of the system (3.6). The noise-induced CR model has the form

$$\begin{cases} \mathrm{d}R(t) = rR(t)\mathrm{d}t - \dfrac{aR(t)C(t)}{1+bR(t)}\mathrm{d}t + \sigma_1 \mathrm{d}W_1, \\ \mathrm{d}C(t) = \dfrac{aR(t)C(t)}{1+bR(t)}\mathrm{d}t - mC(t)\mathrm{d}t + \sigma_2 \mathrm{d}W_2, \end{cases} \tag{3.7}$$

where σ_i, $i = 1, 2$, are the intensities of white noise, and $W_i(t)$, $i = 1, 2$, are independent Wiener processes.

We simulated the CR model's dynamics using the Euler–Maruyama method, which is well suited for stochastic differential equations (SDEs). The model parameters were chosen as the resource growth rate $r = 0.1$, the consumption rate $a = 0.005$, the half-saturation constant $b = 0.001$ and the consumer mortality rate $m = 0.025$. The initial population sizes were set to $R_0 = 20$ for the resource and $C_0 = 10$ for the consumer. Additive environmental stochasticity was incorporated into both equations through Wiener processes sampled from $\mathrm{d}W_i \sim \mathcal{N}(0, \sqrt{\mathrm{d}t})$, $i = 1, 2$, and simulations were repeated across a range of noise intensities, $\sigma_1 = \sigma_2 \in \{0, 0.5, 1.0, 1.5, 2.0\}$. The system was simulated over a time horizon of 1500 units with a time step of $\mathrm{d}t = 0.01$. After each Euler–Maruyama update, any negative population values were prevented by enforcing small positive lower bounds to maintain biological realism.

We used the simulated time series of C and R to learn the dynamics of the CR model using the proposed method described in §2. During the learning phase, we assumed that the system could be represented in the general form

$$\begin{cases} \frac{\mathrm{d}R}{\mathrm{d}t} = f(C, R), \\ \frac{\mathrm{d}C}{\mathrm{d}t} = f(C, R), \end{cases} \tag{3.8}$$

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

Table 1. Variables and data sources used in gas concentration.

variable	abbreviation	unit	data sources
methane	CH_4	parts per million (ppm)	AMS-1, AMS-18
carbon dioxide	CO_2	parts per million (ppm)	AMS-1, AMS-18
ambient temperature	AT	degrees Celsius (°C)	AMS-1, AMS-18
relative humidity	RH	percentage (%)	AMS-1, AMS-18
wind speed	WS	km h^{-1}	AMS-1, AMS-18
wind direction	WD	degrees $\in [0, 360)$	AMS-1, AMS-18
global radiation	GR	W m^{-2}	AMS-1, AMS-18
precipitation	PC	millimetres (mm)	AMS-1, AMS-18
barometric pressure	BP	mB	JP104
dew point	DEW	degrees Celsius (°C)	JP104
photosynthetically active radiation at 16 m	PAR	µmol m^{-2} s^{-1}	JP104
solar radiation	SR	W m^{-2}	JP104
diluent	DIL	m^3	AER

where the time-varying parameters were systematically selected following the procedure in §2.

3.3. Gas concentration

CO_2 and CH_4 are two major greenhouse gases, and a significant amount of these gases is found in oil-sand tailings areas. Our goal is to learn a system of equations that can capture the concentration time series of CO_2 and CH_4 in Alberta's oil-sand tailings ponds based on relevant weather conditions and diluent inputs. Additionally, we leveraged the learned equation to predict future concentrations based on estimated weather conditions and diluent inputs.

We used two different air monitoring station datasets that contain CO_2 and CH_4 concentrations, and one nearby meteorological tower dataset by the Wood Buffalo Environmental Association [44], located in the regional municipality of Wood Buffalo in northeast Alberta, Canada. The air monitoring stations we used are AMS-1: Bertha Ganter–Fort McKay (latitude: 57.189428, longitude: –111.640583) and AMS-18: Stony Mountain (latitude: 55.621408, longitude: –111.172686). The meteorological tower we used is JP104-Site 1004 (latitude: 57.11901, longitude: –111.42542). The parameters used in this study and their sources are described in table 1. Air monitoring station and meteorological tower data can be accessed from [44]. The period of the collected data was from January 2019 to December 2024. Air monitoring stations recorded hourly data with some missing observations. We removed observations containing missing values and took the daily average of the recorded data.

Both stations measure CO_2 and CH_4 and some common weather variables (AT, RH, WS, WD, PC and GR) (table 1). We wanted to keep the same number of weather variables (AT, RH, WS, WD, GR, BP, DEW, PAR, SR and PC) in each dataset. Therefore, for the remaining weather variables (BP, DEW, PAR and SR), measurements from the nearby JP104 meteorological tower were used for both station datasets (table 1).

To incorporate the source of the hydrocarbons, which are responsible for producing these gases, we used the total diluent-loss data from Alberta Energy Regulator reports [45]. Since it is unknown which monitoring stations capture gases from which specific tailings pond as the source, we summed the monthly reported diluent losses across all mining companies in the region between January 2019 and December 2024. We then distributed each monthly total uniformly across the days of that month to obtain daily estimates.

To learn the system of equations, we took up to quadratic terms of the library of candidate nonlinear functions $\boldsymbol{\Theta}(\mathbf{X}, \mathbf{U})$ (equation (2.4)) to fit the data for each gas with a quadratic equation. We considered the diluent-loss data as a source of CO_2 and CH_4, and since the gas concentration varies with the weather conditions, we incorporated all the weather variables as input variables in the candidate terms. Additionally, CO_2 and CH_4 may be statistically and biogeochemically coupled through common emission sources, microbial processes and shared meteorological controls. We therefore included CO_2 as a potential predictor in the equation describing CH_4 concentration dynamics. Therefore, based on these considerations, we considered the learned system of equations to have the general form

$$\begin{cases} \frac{dCO_2}{dt} = f(DIL, \text{weather variables}), \\ \frac{dCH_4}{dt} = f(CO_2, DIL, \text{weather variables}), \end{cases} \tag{3.9}$$

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

Table 2. List of variables and their abbreviations used for cyanobacteria cell count.

variable	description
CB_{cell}	cell count of cyanobacteria sampled on the monitoring day
P	phosphorus concentration sampled on the monitoring day
N	nitrogen concentration sampled on the monitoring day
SD	Secchi depth of lake on the monitoring day
ST	stratification state of lake on the monitoring day
AT	average daily temperature from the day of monitoring to 2 weeks before the next available cyanobacteria sample
PC	average daily precipitation from the day of monitoring to 2 weeks before the next available sample
WS	average daily wind speed from the day of monitoring to 2 weeks before the next available sample
SR	average daily solar radiation from the day of monitoring to 2 weeks before the next available sample
LD	lake depth
LE	lake elevation
PL	% pastureland in watershed
CL	% cropland in watershed
FT	% forest in watershed
WL	% wetland in watershed
LWAR	lake–watershed area ratio

where the weather variables are AT, RH, WS, WD, GR, BP, DEW, PAR, SR and PC, described in table 1.

3.4. Cyanobacteria cell count

Cyanobacterial blooms are commonly found in freshwater ecosystems and can be highly toxic, posing significant risks to aquatic life, wildlife and human health through the production of harmful toxins that contaminate water sources. Our goal is the data-driven discovery of an equation for cyanobacteria cell counts and prediction from the discovered equation based on nutrient availability, meteorological conditions, lake features and watershed data.

The dataset includes observations from 78 lakes and reservoirs across Alberta, Canada, spanning the years from 1986 to 2017 [46]. These lakes and reservoirs are situated between latitudes 49.3° N and 58.8° N and longitudes 110.0079° W and 119.21667° W [47]. The lake monitoring data are collected during May to October, which include phytoplankton data gathered through the Alberta Lake Monitoring Program. The original lake monitoring dataset contains various parameters including chemical concentrations in lake water, Secchi depth measurements, lake stratification data and cyanobacteria cell counts, provided by Alberta Environment and Parks. We used the processed datasets from Heggerud *et al.* [47], which can be accessed from [46]. We averaged the data based on each year and month from different lakes. The variables used for this case study are described in table 2.

We used the method described in §2 to learn a single equation for CB cell count (CB_{cell}) based on all the covariates listed in table 2. The learned equation has the general form

$$\frac{\mathrm{d}CB_{\mathrm{cell}}}{\mathrm{d}t} = f(CB_{\mathrm{cell}}, \text{lake-monitoring variables}, \text{meteorological variables}, \text{lake features}, \text{watershed features}), \tag{3.10}$$

where the lake monitoring variables include P, N, SD and ST. Meteorological variables include AT, PC, WS and SR. Lake features include LD and LE. Watershed features include PL, CL, FT, WL and LWAR. CB_{cell} is retained on the right-hand side of the equation to indicate that the new cell count depends on the existing cell count.

3.5. Simulated weather data

Once the system was learned through the time-varying parameters, we used an ML model to forecast these parameters based on weather conditions. In this study, we considered four weather variables—temperature, humidity, wind speed and precipitation—as predictors of the time-varying parameters. For the gases and CB datasets, these weather variables were available in the dataset and directly used for forecasting. By contrast, for the SIR and CR models, we simulated the corresponding weather variables so

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

that the ML model could be applied in the same way, enabling prediction of the time-varying parameters based on these simulated environmental conditions.

We used the Ornstein–Uhlenbeck (OU) process [48] to simulate the time series of the weather variables. The OU process is represented by an SDE commonly used to model mean-reverting phenomena. The process is described by the equation

$$\mathrm{d}X(t) = -\theta_X \left(X(t) - X_{\text{mean}}\right) \mathrm{d}t + \sigma_X \, \mathrm{d}W(t), \tag{3.11}$$

where $X(t)$ represents the state variable (e.g. temperature) at time t, and X_{mean} is the long-term mean around which the process fluctuates. The parameter θ_X quantifies the rate of mean reversion, indicating how quickly deviations from the mean are corrected. The term σ_X represents the intensity of stochastic noise, and $\mathrm{d}W(t)$ is the Wiener process, which introduces randomness.

To simulate data from the OU process, we discretized the continuous-time equation using the Euler–Maruyama method. The discretized version of the SDE is given by

$$X_{i+1} = X_i - \theta_X \left(X_i - X_{\text{mean}}\right) \Delta t + \sigma_X \sqrt{\Delta t}\, \xi_i,$$

where Δt is the time increment, and the ξ_i are independent standard normal random variables.

The weather simulation model was designed with parameter values that closely mimic realistic dynamics for temperature, humidity, wind speed and precipitation over the simulated period. We used a time step of $\Delta t = 0.01$, which provides sufficient granularity to capture variations in weather conditions over a total duration of 1500 days. The initial temperature was 20°C, the initial humidity was 60%, the initial wind speed was $10\,\mathrm{m\,s^{-1}}$ and the initial precipitation was $0.2\,\mathrm{mm\,day^{-1}}$. The long-term mean values for each parameter were defined as $T_{\text{mean}} = 0$°C, $H_{\text{mean}} = 50\%$, $W_{\text{mean}} = 5\,\mathrm{m\,s^{-1}}$ and $P_{\text{mean}} = 0.1\,\mathrm{mm\,day^{-1}}$. The rates of mean reversion (θ) were chosen as $\theta_T = 0.1$, $\theta_H = 0.05$, $\theta_W = 0.2$ and $\theta_P = 0.1$. The noise intensities were set as $\sigma_T = 10$, $\sigma_H = 5$, $\sigma_W = 1$ and $\sigma_P = 0.05$, corresponding to fluctuations in temperature, humidity, wind speed and precipitation, respectively. Higher noise intensities for temperature and humidity reflect their typically greater variability compared to wind speed and precipitation. Additionally, boundaries were specified for each variable to prevent unrealistic values. Temperature was limited to the range $[-20, 35]$°C, humidity to $[0, 100]$%, wind speed to $[0, 30]\,\mathrm{m\,s^{-1}}$ and precipitation to $[0, 100]\,\mathrm{mm\,day^{-1}}$.

3.6. Machine learning models, error metrics and cross-validation

Each variable's time series was first smoothed using a 30-point rolling window to reduce high-frequency noise. To account for differences in scale across variables, all series were normalized to the range $[0, 1]$ using min–max scaling:

$$X' = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}},$$

where X' denotes the normalized variable and X_{min} and X_{max} are the minimum and maximum values of X, respectively. For all datasets, the candidate library $\boldsymbol{\Theta}(\mathbf{X}, \mathbf{U})$ in equation (2.4) was constructed using monomials of total degree up to $d = 2$.

As a case study, we employed a RF regression model [49] to predict each time-varying parameter independently from four weather-related covariates: air temperature, relative humidity, wind speed and precipitation. The RF model was implemented using the `RandomForestRegressor` class from `sklearn.ensemble` [50]. To ensure ensemble stability, we used 5000 trees (`n_estimators` = 5000). Model complexity was controlled by restricting the maximum tree depth to 5 (`max_depth=5`), requiring a minimum of 10 samples to split an internal node (`min_samples_split=10`), and enforcing at least 5 samples per leaf (`min_samples_leaf=5`). The weather variables served as predictors, and each time-varying parameter was treated as a separate target (figure 1e).

For each dataset, the time series was partitioned into training, validation and testing segments. The testing period was divided into five folds of 30 days each, except for the CB dataset, which consisted of five monthly observations. Temporal ordering was strictly preserved. For each test fold, a validation fold of equal duration was selected immediately preceding the test period (figure 2). The governing equations were fitted to the training data using equation (2.11), and forecasts were generated using equation (2.12).

Model selection was performed using an expanding-window time-series cross-validation strategy (figure 2). The window length w was varied from 7 to 28 days in increments of 7 (from 5 to 25 in increments of 5 for the CB dataset). Simultaneously, the number of time-varying parameters was varied from zero (fixed-parameter model) up to the full set of active parameters, with the bias term always included. Starting from an initial group of validation folds (e.g. folds 1–5), the validation window was progressively expanded (1–6, 1–7, …, 1–9) (figure 2). For each configuration, performance was assessed using the mean absolute error (MAE):

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left|\hat{y}_i - y_i\right|,$$

where y_i and $\hat{y}_i$ denote the observed and predicted values, respectively. The configuration minimizing the average validation MAE was selected and used to forecast the subsequent test fold.

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

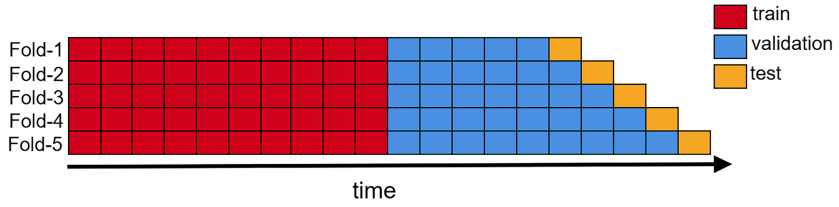


Figure 2. Illustration of the sequential, non-overlapping training, testing and validation scheme used in this study. The training period (red) covers the majority of the time series. The validation window blocks (blue), equal in length to the test window, are positioned immediately before the test period (yellow). For each test fold, only the validation window is expanded forward while the training set remains fixed, producing multiple validation folds without expanding or overlapping the training data.

To examine the influence of interval length and the number of time-varying parameters, simulations were performed for all candidate configurations on each test fold, and the corresponding MAEs were computed. Within each fold, the configuration yielding the lowest test MAE was identified as the *optimal configuration* and compared with the configuration selected through cross-validation. Because the *optimal configuration* was determined retrospectively using the observed test-fold outcomes, it represents a *post hoc* oracle benchmark rather than an unbiased estimate of out-of-sample forecasting performance. This comparison was included to quantify the potential improvement that could be achieved through a more effective adaptive configuration-selection procedure.

For all datasets, we additionally compared the proposed time-varying parameter model with a fixed-parameter baseline, in which the entire time series is represented by constant coefficients (figure 1b) [13]. In both cases, the regularization parameter was set to $\lambda = 0.01$, the sparsity threshold to $\tau = 0.001$ and the maximum number of iterations to 10 000.

To further benchmark our framework, we compared its forecasting performance against two standard ML baselines: a hybrid CNN–LSTM model [51–53] and a GBM [54].

For the CNN–LSTM model, inputs were constructed using sliding windows of length w days. For each target variable, the predictors consisted of the exogenous covariates over the same w-day interval, while the response comprised the corresponding w-day sequence of the target state, aligned in time. For the SIR and CR datasets, the covariates included air temperature (AT), wind speed (WS), precipitation (PC) and relative humidity (RH). For the gas concentration and CB datasets, all weather variables listed in tables 1 and 2 were used.

The CNN–LSTM architecture combines convolutional layers for feature extraction [53] with recurrent memory units [52], and was implemented in `TensorFlow/Keras` [55,56]. The architecture consisted of a one-dimensional causal convolutional layer, Conv1D(filters = F, kernel size = K, activation = ReLU), followed by a max-pooling layer with pool size 2, an LSTM layer with U hidden units, a dropout layer with rate d and two fully connected layers: a ReLU-activated dense layer with D units and a final linear output layer producing w outputs (with $w = 30$ in our experiments). Hyperparameters (F, K, U, D, d) and the learning rate were selected using a `KerasTuner` [57] random search to minimize the validation mean squared error. Training employed the Adam optimizer [58] with early stopping [59] (patience of 10 epochs and restoration of the best-performing weights), using a non-shuffled time-series cross-validation scheme. Within each fold, the most recent block of the training data, matching the test length, was reserved for validation.

As an additional baseline, we implemented a GBM model using the same set of predictors as in the CNN–LSTM set-up. The GBM input consisted of the exogenous covariates at each time step, and the response was the corresponding target value. The model was trained using 1000 boosting trees, a learning rate of 0.01, a maximum tree depth of 30 and a subsampling ratio of 0.9 to mitigate overfitting. Each terminal node was constrained to contain at least 10 samples. The squared error loss function was used, corresponding to a Gaussian regression framework.

3.7. Metropolis–Hastings benchmark

As an additional parameter-estimation benchmark, we implemented the Metropolis–Hastings (MH) algorithm [60,61]. This benchmark was included to compare forecasting performance using coefficients estimated through Bayesian sampling. The MH model used the same preprocessing, state and input variables, temporal folds, forecast horizons and second-order polynomial candidate library as the proposed framework. However, rather than identifying and forecasting a subset of time-varying parameters, MH estimated one fixed coefficient vector for each state equation from the complete training portion of each fold. The resulting coefficients were subsequently held constant over the corresponding test horizon.

For the kth state equation in equation (2.5), we assumed

$$\dot{\mathbf{X}}_k = \boldsymbol{\Theta}(\mathbf{X}, \mathbf{U})\boldsymbol{\xi}_k + \boldsymbol{\varepsilon}_k \quad \text{and} \quad \boldsymbol{\varepsilon}_k \sim \mathcal{N}\left(\mathbf{0}, \sigma_k^2 \mathbf{I}_m\right), \tag{3.12}$$

where $\dot{\mathbf{X}}_k \in \mathbb{R}^m$ is the numerically estimated derivative of the kth state over the complete training portion of the corresponding fold, $\boldsymbol{\Theta}(\mathbf{X}, \mathbf{U})$ is the second-order polynomial candidate library evaluated over the training data, $\boldsymbol{\xi}_k$ is the associated coefficient vector, $\boldsymbol{\varepsilon}_k \in \mathbb{R}^m$ is the residual vector and $\mathbf{I}_m$ is the $m \times m$ identity matrix. The residuals were assumed to be independent and normally distributed with a common variance σ_k^2 for each state equation.

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

The residual standard deviation was fixed before MH sampling according to

$$\sigma_k = \max\left\{0.2\ \mathrm{SD}\left(\dot{\mathbf{X}}_k\right), 10^{-5}\right\}, \tag{3.13}$$

where $\mathrm{SD}(\dot{\mathbf{X}}_k)$ denotes the sample standard deviation of the estimated derivative over the training fold. The lower bound of 10^{-5} was imposed to prevent numerical instability when the derivative variance was very small.

Under these assumptions, the likelihood was

$$p\left(\dot{\mathbf{X}}_k \mid \boldsymbol{\xi}_k, \sigma_k^2\right) \propto \exp\left[-\frac{\left\|\dot{\mathbf{X}}_k - \boldsymbol{\Theta}(\mathbf{X}, \mathbf{U})\boldsymbol{\xi}_k\right\|_2^2}{2\sigma_k^2}\right], \tag{3.14}$$

following standard Bayesian formulations for inverse problems [62,63].

Independent uniform priors were assigned to the entries of $\boldsymbol{\xi}_k$:

$$\xi_{k,j} \sim \mathrm{Uniform}(-1, 1), \qquad j = 1, \ldots, p, \tag{3.15}$$

where p is the number of terms in the candidate library. These bounds were applied to coefficients estimated from the normalized state and input variables.

Posterior samples were generated using a Gaussian random-walk proposal:

$$\boldsymbol{\xi}_k^* = \boldsymbol{\xi}_k^{(s)} + \boldsymbol{\eta} \quad \text{and} \quad \boldsymbol{\eta} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_q\right), \tag{3.16}$$

where $\boldsymbol{\xi}_k^{(s)}$ is the coefficient vector at MH iteration s, $\boldsymbol{\xi}_k^*$ is the proposed coefficient vector and $\boldsymbol{\eta}$ is the proposal perturbation. The proposal covariance matrix $\boldsymbol{\Sigma}_q$ was diagonal. The proposal standard deviation for each coefficient was set to 0.005 times its prior range. Since the prior range was $[-1, 1]$, the resulting proposal standard deviation was 0.01. Proposed values outside the prior bounds were reflected back into the admissible interval. The proposed vector was then accepted according to the standard MH acceptance probability [61].

Two independent MH chains were fitted for each state equation and temporal fold. Each chain contained 6000 iterations. The first 2000 iterations were discarded as burn-in, and every fifth sample from the remaining iterations was retained. Thus, 800 posterior samples were retained from each chain, yielding a combined total of 1600 samples for each state equation and fold.

The first chain was initialized using the least-squares coefficient estimate, truncated to the prior bounds. The second chain was initialized at a dispersed value sampled around the least-squares estimate, with a standard deviation equal to 10% of the prior range. The least-squares estimates were used only for chain initialization and were not used as the reported coefficient estimates.

Convergence and mixing were examined using the chain-specific acceptance rates and the potential scale reduction statistic $\hat{R}$ [64]. The posterior mean, posterior median, coefficient-specific $\hat{R}$ values and the corresponding 2.5 and 97.5% posterior quantiles were recorded for each coefficient.

Motivated by previous applications in which MH-estimated parameters were used in subsequent mechanistic prediction [65–67], the coefficient vector selected from the complete training fold was held constant over the test horizon. The forecast of the kth state was therefore obtained from

$$\hat{\dot{x}}_k(t) = \boldsymbol{\Theta}\left(\hat{\mathbf{x}}(t), \mathbf{u}(t)\right)\hat{\boldsymbol{\xi}}_k^{\mathrm{MH}}, \tag{3.17}$$

where $\hat{\boldsymbol{\xi}}_k^{\mathrm{MH}}$ denotes the coefficient vector used for forecasting. The posterior median was used whenever it generated a numerically stable ODE trajectory. Because a componentwise posterior median is not necessarily a joint posterior sample and can occasionally produce unstable dynamics, the algorithm searched the retained posterior samples when the median trajectory was unstable. In that case, up to 50 joint posterior draws were examined, and the first draw producing a stable trajectory was used for the point forecast. If no stable coefficient draw was identified, the corresponding fold was classified as unstable and no forecast error was reported.

The first observation of the test period was used as the initial condition, and the resulting ODE was integrated over the same test horizon used for the proposed model. The numerical integration was attempted using the LSODA, Radau and BDF solvers. For equations containing exogenous input variables, their observed test-period values were linearly interpolated during numerical integration. Solutions containing non-finite values or absolute state values greater than 20 were treated as numerically unstable.

To quantify uncertainty arising from the posterior coefficient distributions, coefficient vectors were repeatedly sampled from the retained MH samples and propagated through the ODE system. For each fold, the algorithm attempted to retain 30 numerically stable trajectories, with at most 20 attempts per required trajectory. The pointwise 2.5 and 97.5% quantiles of the retained trajectories were used to construct 95% posterior trajectory intervals. These intervals represent uncertainty propagated from the coefficient posterior and do not include an additional realization of the observation-error term in equation (3.12).

The MH analysis was repeated independently for every temporal fold, monitoring station and simulated noise level. Forecast errors were calculated using the same procedure described in §3.6. The MH benchmark should be interpreted as a fixed-parameter Bayesian calibration alternative rather than as a Bayesian implementation of the complete proposed framework. Accordingly, differences in forecasting performance reflect both the coefficient-estimation procedure and the assumption that the MH-estimated coefficients remain constant throughout the forecast horizon.

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

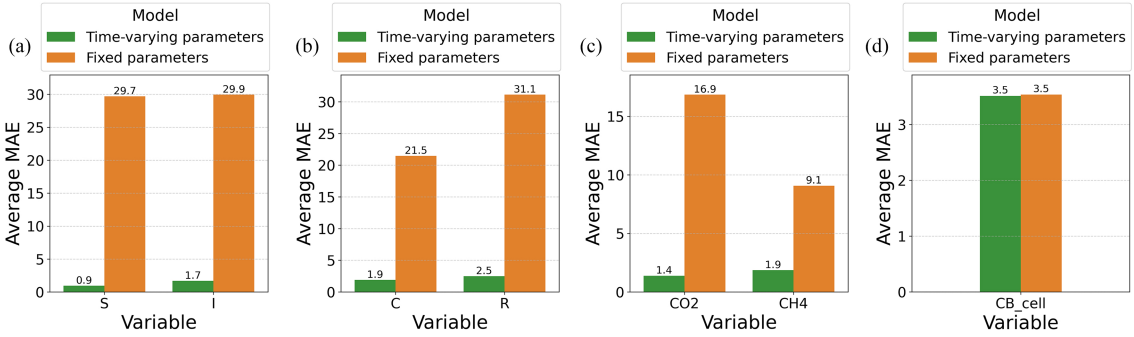


Figure 3. Average learning MAEs for the (a) SIR dataset, (b) CR dataset, (c) gases dataset and (d) CB dataset. Green bars represent the time-varying parameter model, and orange bars represent the fixed-parameter model. MAEs were computed for each variable across different noise levels or monitoring stations within each dataset and then averaged.

4. Results

4.1. Experiment results

4.1.1. Learning and reproducing system dynamics

In the SIR dataset, the time-varying parameter model demonstrated a strong capability of capturing the dynamics, achieving an average MAE of 0.9% for the susceptible (S) population and 1.7% for the infected (I) population (figure 3a). By contrast, the fixed-parameter model yielded significantly higher errors—29.7% for S and 29.9% for I (figure 3a). Across all noise levels, the time-varying model consistently maintained an average MAE below 2%, whereas the fixed-parameter model exhibited considerably poorer accuracy, with MAEs ranging from 28 to 33% (supplementary material, figure S1).

For the CR dataset, the time-varying parameter model effectively reproduced the time-series dynamics, with an average MAE of 1.9% for the consumer (C) variable and 2.5% for the resource (R) variable (figure 3b). In comparison, the fixed-parameter model performed substantially worse, producing MAEs of 21.5% for C and 31.1% for R (figure 3b). When examined across varying noise levels, the time-varying model maintained a stable MAE range of 1.6–2.7%, while the fixed-parameter model fluctuated between 11 and 38% (supplementary material, figure S1).

In the gases dataset, the time-varying parameter approach also exhibited superior performance, capturing the temporal behaviour of CO_2 and CH_4 with MAEs of 1.4 and 1.9%, respectively (figure 3c). Conversely, the fixed-parameter model achieved notably higher MAEs of 16.9% for CO_2 and 9.1% for CH_4 (figure 3c). Averaged across multiple station datasets, the time-varying model maintained an overall MAE below 3%, while the fixed-parameter counterpart ranged between 11 and 15% (supplementary material, figure S1).

In the CB dataset, both models performed comparably, with the time-varying and fixed-parameter models producing nearly identical MAEs of 3.51 and 3.53%, respectively (figure 3d).

Sample plots illustrate that the fixed-parameter model generally fails to capture the evolving trends in the time series across most variables in our case studies, with the exception of CH_4 and CB (supplementary material, figure S5). The model performs well on the CB dataset, which exhibits only a single peak, and moderately on CH_4, which has multiple peaks (supplementary material, figure S5). By contrast, the time-varying parameter model accurately captures the temporal dynamics for all variables (supplementary material, figure S5).

4.1.2. Forecasting system dynamics

For the SIR dataset, the time-varying parameter model demonstrated strong forecasting skill for both the susceptible (S) and infected (I) populations, achieving average MAEs of 10.0 and 9.9%, respectively (figure 4a). By contrast, the fixed-parameter model yielded higher errors of 13.8% for S and 14.0% for I. The MH benchmark produced MAEs of 16.5% for S and 14.8% for I, indicating that estimating a single coefficient vector from the complete training fold was less effective than forecasting the selected time-varying parameters. Across all noise levels, the time-varying model consistently maintained MAEs between 9 and 13%, whereas the fixed-parameter model exhibited errors ranging from 12 to 15% (supplementary material, figure S2).

The purely data-driven baselines performed notably worse. The CNN–LSTM model produced MAEs of 21.4% for S and 24.2% for I, while the GBM yielded corresponding errors of 21.5 and 21.4% (figure 4a). Thus, the proposed model achieved the lowest forecasting error for both SIR state variables among all five approaches.

For the CR dataset, the time-varying parameter model forecasted the consumer (C) and resource (R) variables with average MAEs of 3.9 and 5.4%, respectively (figure 4b). The fixed-parameter model achieved comparable errors of 3.9% for C and 5.1% for R. The MH benchmark also performed competitively, yielding MAEs of 3.8% for C and 5.6% for R. These results indicate that the CR dynamics over the evaluated forecast horizons could be represented reasonably well by fixed-coefficient estimates. Across

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

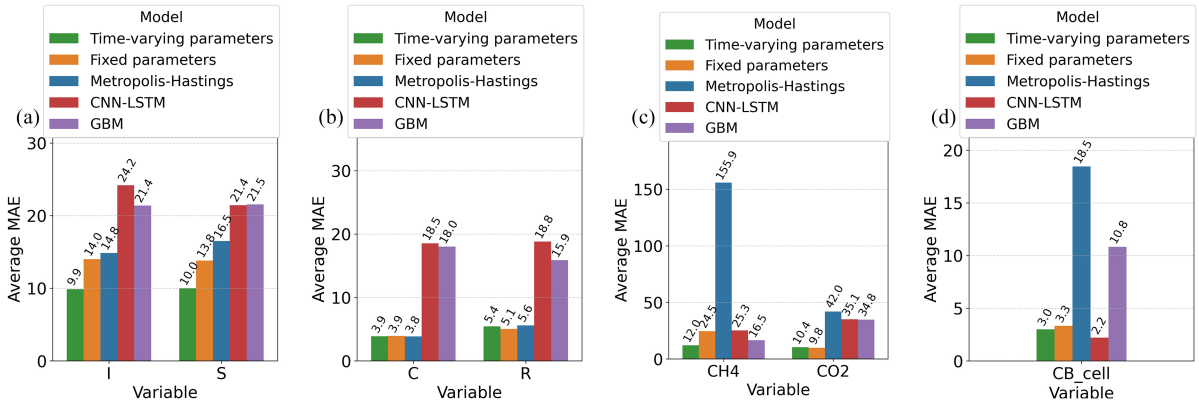


Figure 4. Average forecasting MAEs using the expanding-window cross-validation approach for the (a) SIR dataset, (b) CR dataset, (c) gases dataset and (d) CB dataset. Green bars represent the time-varying parameter model, orange bars represent the fixed-parameter model, blue bars represent the MH model, red bars represent the CNN–LSTM model and purple bars represent the GBM model. MAEs were computed for each variable across different noise levels or monitoring stations within each dataset and then averaged.

varying noise conditions, the three equation-based approaches maintained relatively low forecasting errors, generally between 2 and 7% (supplementary material, figure S2).

By contrast, the CNN–LSTM yielded MAEs of 18.5% for *C* and 18.8% for *R*, while the GBM produced errors of 18.0 and 15.9%, respectively (figure 4b). These errors were approximately three to five times greater than those of the equation-based models.

For the gases dataset, the time-varying parameter model achieved MAEs of 10.4% for CO_2 and 12.0% for CH_4 (figure 4c). The fixed-parameter model produced a slightly lower error of 9.8% for CO_2 but a substantially higher error of 24.5% for CH_4. The MH benchmark performed poorly for both gases, with MAEs of 42.0% for CO_2 and 155.9% for CH_4. The particularly large CH_4 error indicates that a single posterior coefficient vector estimated over the complete training fold was unable to adequately represent the changing gas dynamics over the forecast period.

The CNN–LSTM produced MAEs of 35.1% for CO_2 and 25.3% for CH_4, while the corresponding GBM errors were 34.8 and 16.5% (figure 4c). The time-varying parameter model therefore provided the lowest CH_4 forecasting error and remained substantially more accurate than MH and the purely data-driven models for CO_2. However, the fixed-parameter model achieved a marginally lower CO_2 error than the proposed approach. Across monitoring stations, the time-varying model remained comparatively stable, whereas the performances of the alternative models varied more substantially (supplementary material, figure S2).

For the CB dataset, the time-varying parameter and fixed-parameter models achieved comparable MAEs of 3.0 and 3.3%, respectively (figure 4d). The CNN–LSTM achieved the lowest error of 2.2%, while the GBM yielded a considerably higher MAE of 10.8%. The MH benchmark produced an MAE of 18.5%, which was the highest among the evaluated approaches. This result suggests that estimating the large set of polynomial coefficients jointly through MH was less effective for the high-dimensional CB system than the sparse equation-learning and data-driven forecasting approaches.

4.1.3. Optimal forecasting beyond cross-validation

The expanding-window cross-validation approach did not always capture the most effective set of hyperparameters, namely, the interval length and number of time-varying parameters, for the time-varying parameter model. For each dataset, there existed distinct configurations that produced superior predictive performance (supplementary material, tables S1–S4).

In the SIR dataset, the optimal configuration achieved an MAE of 2.5% for the infected (*I*) population and 2.2% for the susceptible (*S*) population (figure 5a)—an improvement of approximately 8% compared to the hyperparameters of the expanding-window cross-validation set-up. In the CR dataset, the model yielded MAEs of 1.3% for the consumer (*C*) and 2.7% for the resource (*R*) dynamics (figure 5b), reflecting an enhancement of around 2%.

Similarly, for the gases dataset, the optimal configuration produced MAEs of 5.7% for CH_4 and 2.6% for CO_2 (figure 5c), marking an improvement of roughly 5% over the expanding-window cross-validation approach. Finally, in the CB dataset, the best-performing configuration resulted in an MAE of 0.5% (figure 5d), which represents a 2.5% gain in predictive accuracy relative to the expanding-window cross-validation method.

Collectively, these results indicate that while the expanding-window cross-validation framework provides a robust baseline, fine-tuning the interval length and number of time-varying parameters can lead to noticeable improvements in predictive precision across diverse dynamical systems.

Sample forecasting plots demonstrate that the time-varying parameter model with optimal configuration consistently tracks the observed trajectories across all variables in the case studies, resulting in markedly improved forecasting performance (supplementary material, figure S6). By contrast, the fixed-parameter model moderately matched the time series of *R* and CH_4, but failed to capture the evolving dynamics of the remaining variables (supplementary material, figure S6).

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

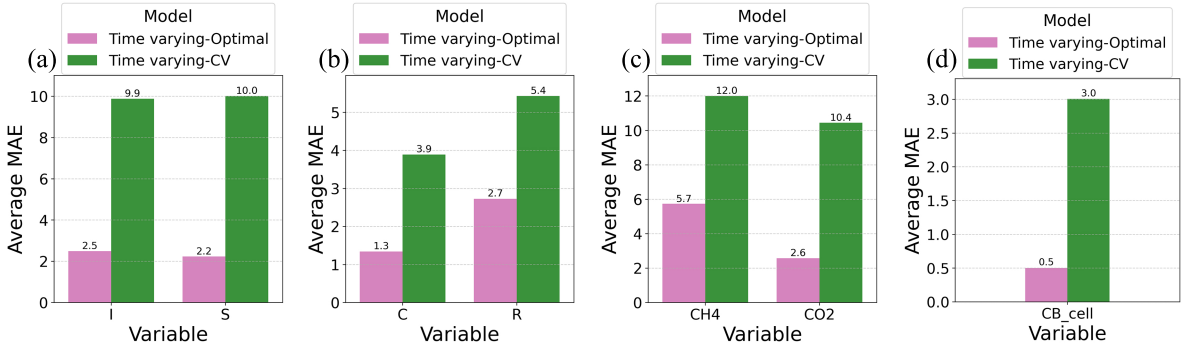


Figure 5. Average forecasting MAEs by the time-varying parameter model based on optimal configuration and cross-validation for the (a) SIR dataset, (b) CR dataset, (c) gases dataset and (d) CB dataset. Green bars represent the model's error from cross-validation (CV), and pink bars represent the model's optimal configuration forecasting error. MAEs were computed for each variable across different noise levels or monitoring stations within each dataset and then averaged.

4.2. Theoretical guarantees

In this subsection, we establish theoretical guarantees for three key components of our framework: (i) the propagation of errors from the learned equations and the forecasted time-varying parameters to the forecasted states, (ii) error bounds for the split structure of learned and forecasted equations, and (iii) conditions under which the time-varying parametrization admits a strictly tighter derived finite-horizon error bound than the best constant coefficient approximation based on the same library.

Our analysis builds on classical ODE stability theory and sparse equation discovery results [12,68,69]. We extend these foundations by deriving finite-horizon forecast error bounds for time-varying parameter models and quantifying error propagation from ML-forecasted parameters to state trajectories.

4.2.1. Finite-horizon forecast error bounds

We now quantify how errors in the learned governing equations and in the forecasted time-varying parameters propagate to errors in the forecasted trajectories. Let $\mathbf{f} : [0, H] \times \mathbb{R}^n \to \mathbb{R}^n$ denote the (unknown) right-hand side of the true system,

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t)) \quad \text{and} \quad t \in [0, H],$$

where $H > 0$ is a finite time horizon. Let $\mathbf{u} : [0, H] \to \mathbb{R}^q$ denote the exogenous-input trajectory supplied to the model. We assume that $\mathbf{f}(\cdot)$ can be uniformly approximated on a compact region by a representation constructed from our candidate library $\Theta(\cdot)$.

Assumption 4.2.1 (Lipschitz dynamics and library approximation [12,68,69]). *Let $D \subset \mathbb{R}^n$ be a compact set containing the trajectories of interest. Assume:*

(a) (Lipschitz dynamics) For each $t \in [0, H]$, the map $x \mapsto f(t, x)$ is Lipschitz on D with constant $L > 0$:

$$\|f(t, x) - f(t, y)\| \leq L\|x - y\| \quad \text{for all } x, y \in D.$$

(b) (Library approximation) There exist coefficient functions $\xi^\star : [0, H] \to \mathbb{R}^p$ and a library operator $\Theta : D \times \mathbb{R}^q \to \mathbb{R}^{n \times p}$ such that

$$\left\|f(t, x) - \Theta(x, u(t))\, \xi^\star(t)\right\| \leq \varepsilon_{\text{lib}} \quad \text{for all } (t, x) \in [0, H] \times D,$$

for some $\varepsilon_{\text{lib}} \geq 0$.

(c) (Bounded library) There exists $B_\Theta > 0$ such that $\|\Theta(x, u(t))\|_{\text{op}} \leq B_\Theta$ for all $(t, x) \in [0, H] \times D$, where $\|\cdot\|_{\text{op}}$ denotes the operator norm.

Our forecasting pipeline produces estimated time-varying coefficients $\hat{\xi}(t)$ (via STRR on sliding windows, followed by ML model forecasts), and uses the model

$$\hat{f}(t, x) := \Theta(x, u(t))\, \hat{\xi}(t).$$

We assume that the ML model forecasts approximate the ideal coefficient functions $\xi^\star(t)$ with a uniform bound on the forecast horizon. Throughout this analysis, the error bounds are conditional on the same exogenous-input trajectory $\mathbf{u}(t)$ being supplied to the true and forecast models.

Assumption 4.2.2 (Accuracy of coefficient forecasts). *There exists $\varepsilon_{\text{par}} \geq 0$ such that*

$$\sup_{t \in [0, H]} \left\|\hat{\xi}(t) - \xi^\star(t)\right\| \leq \varepsilon_{\text{par}}.$$

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

Under assumptions 4.2.1 and 4.2.2, the difference between the true vector field f and the learned-forecasted vector field $\hat{f}$ is uniformly bounded:

$$\|f(t,x) - \hat{f}(t,x)\| \le \varepsilon_{\text{lib}} + B_\Theta \varepsilon_{\text{par}} =: \delta \quad \text{for all } (t,x) \in [0,H] \times D.$$

We now quantify the impact of this bound on the state trajectories.

Theorem 4.2.1 (Conditional finite-horizon forecast error bound). *Suppose assumptions 4.2.1 and 4.2.2 hold and let* $\delta = \varepsilon_{\text{lib}} + B_\Theta \varepsilon_{\text{par}}$. *Let* $\mathbf{x}, \hat{\mathbf{x}} : [0,H] \to D$ *be absolutely continuous solutions, satisfying the respective differential equations almost everywhere, of the true system* $\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t))$ *with initial condition* $\mathbf{x}(0) = \mathbf{x}_0$, *and the learned system* $\dot{\hat{\mathbf{x}}}(t) = \hat{f}(t, \hat{\mathbf{x}}(t))$ *with the same initial condition* $\hat{\mathbf{x}}(0) = \mathbf{x}_0$. *Then, for all* $t \in [0,H]$,

$$\|x(t) - \hat{x}(t)\| \;\le\; \frac{\delta}{L}\left(e^{Lt} - 1\right) \;\le\; \frac{\delta}{L}\left(e^{LH} - 1\right).$$

Proof. Define the error $e(t) := x(t) - \hat{x}(t)$. Differentiating almost everywhere gives

$$\dot{e}(t) = f(t, x(t)) - \hat{f}(t, \hat{x}(t)) = \big(f(t, x(t)) - f(t, \hat{x}(t))\big) + \big(f(t, \hat{x}(t)) - \hat{f}(t, \hat{x}(t))\big).$$

Taking norms and applying assumption 4.2.1(a) and the definition of δ yields

$$\|\dot{e}(t)\| \le L\|e(t)\| + \delta.$$

Let $y(t) := \|e(t)\|$. Since e is absolutely continuous, y is absolutely continuous and $\dot{y}(t) \le \|\dot{e}(t)\|$ for almost every $t \in [0,H]$. Then, $y(0) = 0$ and

$$\dot{y}(t) \le Ly(t) + \delta \quad \text{for a.e. } t \in [0,H].$$

By the differential form of Grönwall's inequality [68,70],

$$y(t) \le \frac{\delta}{L}\left(e^{Lt} - 1\right) \quad \text{for all } t \in [0,H],$$

which proves the stated bound. □

Theorem 4.2.1 is a standard ODE stability result specialized to our library-based representation. It shows that the two error sources represented in this analysis enter the state forecast through the single quantity $\delta = \varepsilon_{\text{lib}} + B_\Theta \varepsilon_{\text{par}}$, combining the approximation error of the library representation and the error of the time-varying parameter forecasts.

4.2.2. Bounds for the split structure of learned and forecasted equations

We now specialize the baseline bound to the learned forecasting model in this paper. For reference, we restate the split form of the learned dynamics (equation (2.11)) and its forecasting counterpart (equation (2.12)). Let

$$\dot{\mathbf{x}}(t) = \mathbf{f}_{\text{tv}}(t, \mathbf{x}(t)) := \Theta_{\text{top}N}(x(t), u(t))\, \xi_{\text{tv}}(t) + \Theta_{\text{fix}}(x(t), u(t))\, \bar{\xi}, \tag{4.1}$$

where $\Theta_{\text{top}N}$ denotes the selected time-varying block of the active library, Θ_{fix} denotes the remaining active library block and $\bar{\xi}$ denotes the fixed coefficient vector for the non-top-n active terms. The forecast model is

$$\dot{\hat{\mathbf{x}}}(t) = \hat{\mathbf{f}}_{\text{tv}}(t, \hat{\mathbf{x}}(t)) := \Theta_{\text{top}N}(\hat{x}(t), u(t))\, \hat{\xi}_{\text{tv}}(t) + \Theta_{\text{fix}}(\hat{x}(t), u(t))\, \bar{\xi}. \tag{4.2}$$

Assumption 4.2.3 (Uniform boundedness of the top-n library block). *There exists* $B_{\text{top}N} > 0$ *such that*

$$\sup_{t \in [0,H]} \sup_{x \in D} \|\Theta_{\text{top}N}(x, u(t))\|_{\text{op}} \le B_{\text{top}N}.$$

Assumption 4.2.4 (Uniform accuracy of time-varying coefficient forecasts). *There exists* $\varepsilon_{\text{top}N} \ge 0$ *such that*

$$\sup_{t \in [0,H]} \|\hat{\xi}_{\text{tv}}(t) - \xi_{\text{tv}}(t)\| \le \varepsilon_{\text{top}N}.$$

Theorem 4.2.2 (Split-model forecast bound: only $\Theta_{\text{top}N}$ forecast errors enter). *Assume assumption 4.2.1(a) holds for* f_{tv} *in equation (4.1), and assumptions 4.2.3 and 4.2.4 hold. Let* $\mathbf{x}, \hat{\mathbf{x}} : [0,H] \to D$ *be absolutely continuous solutions, satisfying the respective differential equations almost everywhere, of equations (4.1) and (4.2) with the same initial condition* $\mathbf{x}(0) = \hat{\mathbf{x}}(0)$. *If* $x(t), \hat{x}(t) \in D$ *for all* $t \in [0,H]$, *then for all* $t \in [0,H]$,

$$\|x(t) - \hat{x}(t)\| \le \frac{B_{\text{top}N}\varepsilon_{\text{top}N}}{L}\left(e^{Lt} - 1\right) \le \frac{B_{\text{top}N}\varepsilon_{\text{top}N}}{L}\left(e^{LH} - 1\right).$$

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

Proof. For any $(t,x) \in [0,H] \times D$, under the common-input condition stated above, the fixed block cancels exactly:

$$f_{\mathrm{tv}}(t,x) - \hat{f}_{\mathrm{tv}}(t,x) = \Theta_{\mathrm{top}N}(x,u(t))\big(\xi_{\mathrm{tv}}(t) - \hat{\xi}_{\mathrm{tv}}(t)\big).$$

Hence, by assumptions 4.2.3 and 4.2.4,

$$\sup_{(t,x)\in[0,H]\times D} \|f_{\mathrm{tv}}(t,x) - \hat{f}_{\mathrm{tv}}(t,x)\| \le B_{\mathrm{top}N}\epsilon_{\mathrm{top}N}.$$

Applying the same Grönwall argument as in theorem 4.2.1, with $\delta = B_{\mathrm{top}N}\epsilon_{\mathrm{top}N}$, gives the stated bound. □

Remark 4.2.1. *Theorem 4.2.2 is specific to the split structure of equations (4.1)–(4.2): under the conditional assumption that the same exogenous-input trajectory* $\mathbf{u}(t)$ *is supplied to both models, the fixed coefficients* $\bar{\xi}$ *do not appear in the forcing term of the error bound. Consequently, improving the forecast accuracy of* $\hat{\xi}_{\mathrm{tv}}$ *for the selected* $\Theta_{\mathrm{top}N}$ *terms directly tightens the finite-horizon state forecast bound, while the fixed block affects the bound only indirectly through the Lipschitz constant L of the resulting vector field.*

4.2.3. When time-varying parameters yield tighter bounds than fixed parameters

Finally, we compare our time-varying parametrization with a fixed-parameter model. Let $\mathcal{C}_{\mathrm{const}}$ denote the class of constant coefficient vectors:

$$\mathcal{C}_{\mathrm{const}} := \{\xi : [0,H] \to \mathbb{R}^p : \ \xi(t) \equiv c\},$$

and let $\mathcal{C}_{\mathrm{tv}}^{(m)}$ denote the class of piecewise constant coefficient functions with at most m subintervals:

$$\mathcal{C}_{\mathrm{tv}}^{(m)} := \left\{ \xi : [0,H] \to \mathbb{R}^p : \begin{array}{l} \text{there exist } K \le m,\ 0 = t_0 < \cdots < t_K = H, \\ \xi(t) = c_r \text{ for } t \in I_r,\ c_r \in \mathbb{R}^p,\ r = 1,\dots,K \end{array} \right\},$$

where $I_r = [t_{r-1}, t_r)$ for $r < K$ and $I_K = [t_{K-1}, t_K]$. We quantify how well these classes can approximate the ideal coefficient trajectory $\xi^\star(t)$.

Definition 4.2.1 (Best uniform coefficient approximation errors)**.** *Let*

$$E_{\mathrm{const}} := \inf_{\xi\in\mathcal{C}_{\mathrm{const}}} \sup_{t\in[0,H]} \|\xi^\star(t) - \xi(t)\| \quad \textit{and} \quad E_{\mathrm{tv}}^{(m)} := \inf_{\xi\in\mathcal{C}_{\mathrm{tv}}^{(m)}} \sup_{t\in[0,H]} \|\xi^\star(t) - \xi(t)\|.$$

Clearly $\mathcal{C}_{\mathrm{const}} \subset \mathcal{C}_{\mathrm{tv}}^{(m)}$, so $E_{\mathrm{tv}}^{(m)} \le E_{\mathrm{const}}$ for every m. The next lemma shows that, under mild regularity, the inequality is strict for sufficiently large m whenever $\xi^\star(t)$ is truly time-varying. The result follows directly from the uniform continuity of continuous functions on the compact interval $[0,H]$ [71].

Assumption 4.2.5 (Regularity and non-stationarity of the ideal coefficients)**.** *Each component of* $\xi^\star(t)$ *is continuous on* $[0,H]$*, and* $\xi^\star$ *is not constant on* $[0,H]$ *(that is, there exist* t_1, t_2 *with* $\xi^\star(t_1) \ne \xi^\star(t_2)$*).*

Lemma 4.2.1 (Step-function approximation beats constant approximation)**.** *Suppose assumption 4.2.5 holds. Then* $E_{\mathrm{const}} > 0$*, and for every* $0 < \varepsilon < E_{\mathrm{const}}$ *there exists an integer* $m_\varepsilon \ge 1$ *such that, for all* $m \ge m_\varepsilon$*,*

$$E_{\mathrm{tv}}^{(m)} \le \varepsilon < E_{\mathrm{const}}.$$

In particular, for all sufficiently large m, we have $E_{\mathrm{tv}}^{(m)} < E_{\mathrm{const}}$*.*

Proof. Because $\xi^\star$ is continuous on the compact interval $[0,H]$, it is uniformly continuous. Hence, for any $\varepsilon > 0$, there exists $\delta > 0$ such that $\|\xi^\star(t) - \xi^\star(s)\| < \varepsilon$ whenever $|t - s| < \delta$.

Given such a δ, choose m_ε large enough that a partition of $[0,H]$ into m_ε subintervals has each interval of length strictly less than δ. Define a piecewise constant function $\tilde{\xi} \in \mathcal{C}_{\mathrm{tv}}^{(m_\varepsilon)}$ by setting $\tilde{\xi}(t)$ equal to $\xi^\star$ evaluated at a reference point in each subinterval. By construction,

$$\sup_{t\in[0,H]} \|\xi^\star(t) - \tilde{\xi}(t)\| \le \varepsilon,$$

so $E_{\mathrm{tv}}^{(m_\varepsilon)} \le \varepsilon$ and the same bound holds for all $m \ge m_\varepsilon$.

To show $E_{\mathrm{const}} > 0$, note that by assumption 4.2.5 there exist t_1, t_2 such that $\xi^\star(t_1) \ne \xi^\star(t_2)$. Let

$$V := \|\xi^\star(t_1) - \xi^\star(t_2)\| > 0.$$

For any constant vector $c \in \mathbb{R}^p$,

$$V = \|\xi^\star(t_1) - \xi^\star(t_2)\| \le \|\xi^\star(t_1) - c\| + \|\xi^\star(t_2) - c\| \le 2 \sup_{t\in[0,H]} \|\xi^\star(t) - c\|,$$

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

so $\sup_{t\in[0,H]} \|\xi^\star(t) - c\| \geq V/2$ for every c. Taking the infimum over c yields $E_{\text{const}} \geq V/2 > 0$. Choosing any $0 < \varepsilon < E_{\text{const}}$ then gives the claim. □

We now combine lemma 4.2.1 with theorem 4.2.1 to obtain a sufficient condition under which the time-varying parametrization yields a strictly smaller derived finite-horizon error bound than the best constant coefficient approximation based on the same library.

Theorem 4.2.3 (When time-varying parameters yield a tighter finite-horizon bound than fixed parameters). *Suppose assumptions 4.2.1, 4.2.2 and 4.2.5 hold. Assume that the infima defining E_{const} and $E_{\text{tv}}^{(m)}$ are attained for the values of m considered. Consider two idealized models built on the same library Θ:*

- *a fixed-parameter model with coefficient trajectory $\xi_{\text{const}}^\star \in \mathcal{C}_{\text{const}}$ achieving the error E_{const},*
- *a time-varying model with m intervals and coefficient trajectory $\xi_{\text{tv}}^\star \in \mathcal{C}_{\text{tv}}^{(m)}$ achieving the error $E_{\text{tv}}^{(m)}$.*

Let $\hat{\mathbf{x}}_{\text{const}}(t)$ denote the trajectory of the fixed-parameter model

$$\dot{\hat{\mathbf{x}}}_{\text{const}}(t) = \Theta(\hat{\mathbf{x}}_{\text{const}}(t), \mathbf{u}(t))\, \xi_{\text{const}}^\star(t) \quad \textit{and} \quad \hat{\mathbf{x}}_{\text{const}}(0) = \mathbf{x}_0.$$

Let $\hat{\mathbf{x}}_{\text{tv}}^{(m)}(t)$ denote the trajectory of the time-varying model

$$\dot{\hat{\mathbf{x}}}_{\text{tv}}^{(m)}(t) = \Theta(\hat{\mathbf{x}}_{\text{tv}}^{(m)}(t), \mathbf{u}(t))\, \xi_{\text{tv}}^\star(t) \quad \textit{and} \quad \hat{\mathbf{x}}_{\text{tv}}^{(m)}(0) = \mathbf{x}_0,$$

with $\xi_{\text{tv}}^\star \in \mathcal{C}_{\text{tv}}^{(m)}$.

Assume that the true trajectory $\mathbf{x}(t)$ and both model trajectories are absolutely continuous, satisfy their respective differential equations almost everywhere and remain in D for all $t \in [0, H]$.

For these two models, define

$$\delta_{\text{const}} := \varepsilon_{\text{lib}} + B_\Theta E_{\text{const}} \quad \textit{and} \quad \delta_{\text{tv}}^{(m)} := \varepsilon_{\text{lib}} + B_\Theta E_{\text{tv}}^{(m)}.$$

Then, for any m such that $E_{\text{tv}}^{(m)} < E_{\text{const}}$,

$$\delta_{\text{tv}}^{(m)} < \delta_{\text{const}},$$

and therefore

$$\frac{\delta_{\text{tv}}^{(m)}}{L}\left(e^{LH} - 1\right) < \frac{\delta_{\text{const}}}{L}\left(e^{LH} - 1\right).$$

Moreover,

$$\sup_{t\in[0,H]} \|x(t) - \hat{x}_{\text{tv}}^{(m)}(t)\| \leq \frac{\delta_{\text{tv}}^{(m)}}{L}\left(e^{LH} - 1\right),$$

while

$$\sup_{t\in[0,H]} \|x(t) - \hat{x}_{\text{const}}(t)\| \leq \frac{\delta_{\text{const}}}{L}\left(e^{LH} - 1\right).$$

In particular, for all sufficiently large m, the time-varying parametrization admits a strictly smaller derived finite-horizon forecast error bound than the best fixed-parameter approximation based on the same library.

Proof. Fix any m such that $E_{\text{tv}}^{(m)} < E_{\text{const}}$. For this choice of m, we have $\delta_{\text{tv}}^{(m)} < \delta_{\text{const}}$. Since the coefficient approximation errors of the fixed and time-varying models are E_{const} and $E_{\text{tv}}^{(m)}$, respectively, applying the same argument as in theorem 4.2.1 separately to the two models gives

$$\sup_{t\in[0,H]} \|x(t) - \hat{x}_{\text{const}}(t)\| \leq \frac{\delta_{\text{const}}}{L}(e^{LH} - 1)$$

and

$$\sup_{t\in[0,H]} \|x(t) - \hat{x}_{\text{tv}}^{(m)}(t)\| \leq \frac{\delta_{\text{tv}}^{(m)}}{L}(e^{LH} - 1).$$

Since $\delta_{\text{tv}}^{(m)} < \delta_{\text{const}}$, the time-varying model has the smaller derived upper bound. By lemma 4.2.1, the condition $E_{\text{tv}}^{(m)} < E_{\text{const}}$ holds for all sufficiently large m. □

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

Theorem 4.2.3 formalizes the intuition that, when the ideal coefficient trajectory is non-stationary in time, a sufficiently rich time-varying parametrization can represent the coefficient trajectory more accurately than any fixed-parameter model, and therefore admits a strictly tighter derived finite-horizon error bound.

5. Discussion

We present a data-driven framework that couples sparse equation discovery with machine-learned parameter evolution to enable adaptive learning and forecasting of dynamical systems. The results show that allowing a subset of coefficients to vary over time improves both learning fidelity and short-term predictive accuracy, particularly for non-stationary processes such as epidemic transmission and environmental gas emissions. Compared with fixed-coefficient sparse regression, the proposed model reduced forecasting MAE by 2–7% across most datasets and maintained training errors below 3%, indicating that temporal parametrization can enhance model expressiveness without sacrificing interpretability.

Traditional ML models are widely used in epidemiological and environmental prediction [72–74]. While these models often achieve strong point forecasts, their black-box structure limits mechanistic interpretation [14,32]. In our experiments, two common data-driven forecasters—CNN–LSTM and GBM—produced substantially higher forecasting MAEs and showed greater sensitivity to noise, indicating that such models alone may struggle to generalize when system mechanisms evolve, or external drivers are only partially informative. Rather than viewing this as a weakness of ML itself, this outcome highlights the importance of inductive structure: embedding ML-predicted evolving coefficients into discovered differential equations can improve both interpretability and forecasting stability [8,9]. Our contribution is not a new sparse-regression or ML algorithm, but a unified forecasting framework that connects the two. The framework selectively assigns time variation to a subset of active coefficients, retains the remaining coefficients as fixed and forecasts the evolving coefficients from external covariates before reinserting them into the discovered ODEs. This preserves the interpretability of the learned dynamics while allowing the forecast model to adapt to non-stationary processes. The accompanying finite-horizon analysis further characterizes how equation-learning and parameter-forecasting errors propagate to the predicted trajectories. This integration is especially valuable for evolving natural systems—including influenza transmission, climate-forced ecological interactions, atmospheric gas emissions and cyanobacteria dynamics—where fixed-coefficient or unconstrained ML forecasters may fail to capture transient or non-stationary responses [4,75–77].

The MH comparison further clarifies that Bayesian coefficient estimation alone does not ensure accurate forecasting when the underlying dynamics evolve over time. Its competitive performance for the CR system suggests that a single posterior coefficient vector can be adequate when the dynamics are approximately stationary over a short forecast horizon. By contrast, its weaker performance for SIR, gases and CB indicates that holding all estimated coefficients fixed can become restrictive when the system is non-stationary or the candidate library is high-dimensional. In particular, estimating many correlated polynomial coefficients jointly may increase posterior uncertainty and produce coefficient combinations whose numerical integration amplifies small estimation errors. These findings support the use of MH as a parameter-estimation benchmark, while emphasizing that the main advantage of the proposed framework arises from identifying a sparse dynamical structure and explicitly forecasting the coefficients that change over time.

The simulated SIR and CR systems provided a controlled comparison between intrinsically non-stationary and approximately stationary parameter regimes. While our framework recovered accurate dynamics for both systems, the forecasting benefit of time-varying coefficients was substantially more pronounced in SIR. This outcome is consistent with theoretical expectations: CR trajectories, driven by nearly constant interaction strengths, can be reasonably approximated by fixed coefficients over short horizons, whereas SIR evolution is strongly governed by non-stationary drivers—most critically, a temporally shifting transmission rate. Consequently, the fixed-coefficient sparse regression baseline exhibited large forecasting error for SIR, reinforcing the importance of temporal parametrization [8,35]. Similar advantages of time-varying formulations have been reported in epidemiological and ecological models under seasonal forcing and external perturbations [78,79].

In the empirical datasets, the time-varying model achieved substantially lower errors when learning and forecasting CO_2 and CH_4 dynamics. The fixed-coefficient sparse model struggled especially with forecasting CH_4, and the CNN–LSTM and GBM baselines struggled with both, reflecting the influence of seasonal cycles and emission fluctuations on atmospheric gas dynamics. Because environmental systems shift continuously, capturing their behaviour requires time-varying representations [80]. In the CB dataset, both models performed similarly during learning, suggesting that some systems can be approximated with fixed parameters; however, the time-varying model achieved a modest improvement over the fixed-parameter model during forecasting. Collectively, these results support a constructive conclusion: equation learners that incorporate time-varying parameters yield more noise-robust and more accurate finite-horizon forecasts than fixed-coefficient sparse regression or standalone ML forecasters.

In noise-free settings, both sparse models achieved low forecasting MAE, confirming that fixed-coefficient representations can be adequate when dynamics are nearly stationary and data are clean. However, performance diverged under noise and transient dynamical regimes. The time-varying formulation remained stable across both SIR and CR learning intervals, while the fixed-coefficient model exhibited high variance in CR folds, producing substantially larger learning error. Interestingly, the fixed-coefficient model exhibited high learning MAE at certain CR noise levels but lower short-horizon prediction error, as its coefficients converged to long-run averages that smoothed local fluctuations. While such averaging can appear competitive over small forecast windows, it obscures derivative-level dynamics and limits regime-specific generalization. By contrast, learning coefficients on short intervals improves dynamical expressiveness and reduces cross-fold error, but forecasting accuracy can still degrade if the prediction horizon is not well-aligned with the timescale of coefficient evolution.

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

Forecast accuracy in our framework is conditioned on the predictive skill of the time-varying parameters, which in this study were forecast using a single RF model. Although RF effectively captures nonlinear feature–parameter relationships, forecasting models with evolving dynamics may benefit from predictors with complementary inductive biases or stronger temporal adaptation [81,82]. Neural networks, Bayesian networks and hybrid learners have shown promise in modelling complex, noisy and transient feature–response interactions [8,9,83]. Future work should therefore explore multi-baseline or ensemble-based parameter predictors to improve generalization and reduce propagation drift under noise, particularly for rapidly evolving or seasonally forced systems [40,84].

Although expanding-window cross-validation is standard for hyperparameter selection, it may not identify the best interval length or the most effective number of time-varying parameters, as it optimizes for average fold performance rather than segment-specific predictive skill. Our fold-specific evaluation revealed configurations that consistently outperformed the cross-validation-selected model—most notably in the SIR system, where optimal interval alignment reduced forecasting MAE by approximately 8% for both S and I, with additional 2–5% gains observed in the CR, gases and CB datasets. We further observed that cross-validation error increased more sharply in the presence of noise, a known limitation of averaging-based validation that can obscure locally optimal configurations [85,86]. These findings motivate future work on forecast-aware validation objectives, noise-weighted model selection and adaptive interval tuning for transient or externally forced dynamics.

Candidate terms were constructed using domain knowledge to ensure a well-posed and interpretable library. For simulated systems (SIR and CR), where the governing interactions are known, state variables were used directly as candidate functions. For empirical datasets (gases and CB), the library was expanded to include measured source variables and weather covariates anticipated to modulate dynamics. Although this design supports strong performance, further reductions in library size are often possible without degrading forecast skill. Causal selection strategies—such as restricting predictors to the Markov blanket in a Bayesian network [47]—can substantially lower dimensionality while preserving short-horizon accuracy. Because the cost of learning and predicting evolving coefficients scales with the number of active terms, library reduction also bounds the number of time-varying parameters, improving computational efficiency and robustness. Our framework readily accommodates such dataset-specific library adaptation, enabling future integration of automated causal or correlation-based term selection to further enhance scalability and generalization.

Although the framework achieved strong learning and forecasting accuracy, the analysis also identifies several opportunities for improvement. First, expanding-window cross-validation, while standard, may miss locally optimal choices of interval length and time-varying parameter counts because it prioritizes average fold performance rather than horizon-specific forecast skill. Second, autoregressive ML forecasters exhibit error accumulation over longer horizons, especially under noise or rapid regime shifts, a challenge that can be mitigated through dynamical constraints and noise-aware validation objectives. Third, parameter forecasting depends on exogenous drivers (here, four weather covariates); limited variability or measurement error in these inputs can reduce coefficient-forecast reliability, motivating future work on adaptive or expanded driver sets. Fourth, the selection of time-varying parameters is based on the highest correlation (with the bias term always treated as time-varying), which may introduce selection bias and can fail when correlations are transient, noisy or do not reflect true causal influence. Finally, the candidate library was initialized using prior system knowledge, which ensured interpretability but may exclude latent or unobserved drivers; future extensions should investigate automated causal or correlation-based library reduction and selection to enhance scalability and generalization. Collectively, these considerations highlight areas for careful model design and future methodological improvements.

Ethics. This work did not require ethical approval from a human subject or animal welfare committee.

Data accessibility. Air monitoring station and meteorological tower data were obtained from the Wood Buffalo Environmental Association (WBEA) network [44]. Diluent loss data were sourced from publicly accessible Alberta Energy Regulator (AER) reports [45]. The cyanobacteria dataset was obtained from Heggerud *et al.* [46]. All data and code used in this study are publicly available through a Zenodo repository [87].

Supplementary material is available online [89].

Declaration of AI use. We polished spelling, grammar and general style of this paper using ChatGPT [88].

Authors' contributions. A.K.C.: conceptualization, data curation, formal analysis, investigation, methodology, software, visualization, writing—original draft, writing—review and editing; H.W.: conceptualization, data curation, funding acquisition, methodology, project administration, resources, software, supervision, validation, writing—review and editing; P.R.: methodology, supervision, validation, writing—review and editing.

All authors gave final approval for publication and agreed to be held accountable for the work performed therein.

Conflict of interest declaration. We declare we have no competing interests.

Funding. H.W.'s research was partially supported by the Natural Sciences and Engineering Research Council of Canada (Individual Discovery Grant RGPIN-2025-05734) and the Canada Research Chairs programme (Tier 1 Canada Research Chair Award). P.R. also acknowledges support from an NSERC Discovery Grant.

Acknowledgements. We would like to thank ILMEE members for providing valuable feedback. Computational resources were provided by the Digital Research Alliance of Canada (formerly Compute Canada).

References

1. Baker RE, Pena JM, Jayamohan J, Jérusalem A. 2018 Mechanistic models versus machine learning, a fight worth fighting for the biological community? *Biol. Lett.* **14**, 20170660. (doi:10.1098/rsbl.2017.0660)

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

2. Lan Y, Shin SY, Nguyen LK. 2025 From shallow to deep: the evolution of machine learning and mechanistic model integration in cancer research. *Curr. Opin. Syst. Biol.* **40**, 100541. (doi:10.1016/j.coisb.2025.100541)
3. Trappe MI, Chisholm RA. 2023 A density functional theory for ecology across scales. *Nat. Commun.* **14**, 1089. (doi:10.1038/s41467-023-36628-4)
4. Small CC, Cho S, Hashisho Z, Ulrich AC. 2015 Emissions from oil sands tailings ponds: review of tailings pond parameters and emission estimates. *J. Pet. Sci. Eng.* **127**, 490–501. (doi:10.1016/j.petrol.2014.11.020)
5. Natchimuthu S, Panneer Selvam B, Bastviken D. 2014 Influence of weather variables on methane and carbon dioxide flux from a shallow pond. *Biogeochemistry* **119**, 403–413. (doi:10.1007/s10533-014-9976-z)
6. Lowen AC, Mubareka S, Steel J, Palese P. 2007 Influenza virus transmission is dependent on relative humidity and temperature. *PLOS Pathog.* **3**, e151. (doi:10.1371/journal.ppat.0030151)
7. Shaman J, Kohn M. 2009 Absolute humidity modulates influenza survival, transmission, and seasonality. *Proc. Natl Acad. Sci. USA* **106**, 3243–3248. (doi:10.1073/pnas.0806852106)
8. Chakraborty AK, Wang H, Ramazi P. 2024 From policy to prediction: assessing forecasting accuracy in an integrated framework with machine learning and disease models. *J. Comput. Biol.* **31**, 1104–1117. (doi:10.1089/cmb.2023.0377)
9. Wang X, Wang H, Ramazi P, Nah K, Lewis M. 2022 From policy to prediction: forecasting COVID-19 dynamics under imperfect vaccination. *Bull. Math. Biol.* **84**, 90. (doi:10.1007/s11538-022-01047-x)
10. Banga JR, Villaverde AF. 2025 Mechanistic dynamic modelling of biological systems: the road ahead. *Curr. Opin. Syst. Biol.* **42**, 100553. (doi:10.1016/j.coisb.2025.100553)
11. Song W *et al*. 2024 Towards data-driven discovery of governing equations in geosciences. *Commun. Earth Environ.* **5**, 589. (doi:10.1038/s43247-024-01760-6)
12. Brunton SL, Proctor JL, Kutz JN. 2016 Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **113**, 3932–3937. (doi:10.1073/pnas.1517384113)
13. Rudy SH, Brunton SL, Proctor JL, Kutz JN. 2017 Data-driven discovery of partial differential equations. *Sci. Adv.* **3**, e1602614. (doi:10.1126/sciadv.1602614)
14. Li S, Kaiser E, Laima S, Li H, Brunton SL, Kutz JN. 2019 Discovering time-varying aerodynamics of a prototype bridge by sparse identification of nonlinear dynamical systems. *Phys. Rev. E* **100**, 022220. (doi:10.1103/physreve.100.022220)
15. Fukami K, Murata T, Zhang K, Fukagata K. 2021 Sparse identification of nonlinear dynamics with low-dimensionalized flow representations. *J. Fluid Mech.* **926**, A10. (doi:10.1017/jfm.2021.697)
16. Callaham JL, Brunton SL, Loiseau JC. 2022 On the role of nonlinear correlations in reduced-order modelling. *J. Fluid Mech.* **938**, A1. (doi:10.1017/jfm.2021.994)
17. Callaham JL, Rigas G, Loiseau JC, Brunton SL. 2022 An empirical mean-field model of symmetry-breaking in a turbulent wake. *Sci. Adv.* **8**, eabm4786. (doi:10.1126/sciadv.abm4786)
18. Loiseau JC. 2020 Data-driven modeling of the chaotic thermal convection in an annular thermosyphon. *Theor. Comput. Fluid Dyn.* **34**, 339–365. (doi:10.1007/s00162-020-00536-w)
19. Loiseau JC, Brunton SL. 2018 Constrained sparse galerkin regression. *J. Fluid Mech.* **838**, 42–67. (doi:10.1017/jfm.2017.823)
20. Prokop B, Gelens L. 2024 From biological data to oscillator models using SINDy. *iScience* **27**, 109316. (doi:10.1016/j.isci.2024.109316)
21. Mangan NM, Brunton SL, Proctor JL, Kutz JN. 2016 Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Trans. Mol. Biol. Multi-Scale Commun.* **2**, 52–63. (doi:10.1109/tmbmc.2016.2633265)
22. Kaiser E, Kutz JN, Brunton SL. 2018 Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. R. Soc. A* **474**, 20180335. (doi:10.1098/rspa.2018.0335)
23. Wu X, McDermott M, MacLean AL. 2025 Data-driven model discovery and model selection for noisy biological systems. *PLoS Comput. Biol.* 21, e1012762. (doi:10.1371/journal.pcbi.1012762)
24. Prokop B, Gelens L. 2023 Data-driven discovery of oscillator models using SINDy: towards the application on experimental data in biology. *bioRxiv*. (doi:10.1101/2023.08.25.554817)
25. Sandoz A, Ducret V, Gottwald GA, Vilmart G, Perron K. 2023 SINDy for delay-differential equations: application to model bacterial zinc response. *Proc. R. Soc. A* **479**, 20220556. (doi:10.1098/rspa.2022.0556)
26. Prabhu S, Rangarajan S, Kothare M. 2023 Data-driven discovery of sparse dynamical model of cardiovascular system for model predictive control. *Comput. Biol. Med.* **166**, 107513. (doi:10.1016/j.compbiomed.2023.107513)
27. Brunton SL, Proctor JL, Kutz JN. 2016 Sparse identification of nonlinear dynamics with control (SINDYc). *IFAC-PapersOnLine* **49**, 710–715. (doi:10.1016/j.ifacol.2016.10.249)
28. Thiele G, Fey A, Sommer D, Krüger J. 2020 System identification of a hysteresis-controlled pump system using SINDy. In*24th Int. Conf. on System Theory, Control and Computing (ICSTCC)*, pp. 457–464. New York, NY: IEEE. (doi:10.1109/ICSTCC50638.2020.9259776)
29. Wulff P, Gräbner N, von Wagner U. 2024 Minimal model identification of drum brake squeal via SINDy. *Arch. Appl. Mech.* **94**, 3101–3117. (doi:10.1007/s00419-024-02659-6)
30. Lee H, Ren R, Qian Y, Rosen J. 2025 Energy reduction for wearable pneumatic valve system with SINDy and time-variant model predictive control. *IEEE/ASME Trans. Mechatron.* **30**, 862–872. (doi:10.1109/tmech.2024.3458092)
31. Guo L, Yang X, Zheng Z, Riemer N, Tessum CW. 2024 Uncertainty quantification in reduced-order gas-phase atmospheric chemistry modeling using ensemble SINDy. (doi:10.48550/arXiv.2407.09757)
32. Rubio-Herrero J, Marrero CO, Fan WTL. 2022 Modeling atmospheric data and identifying dynamics temporal data-driven modeling of air pollutants. *J. Cleaner Prod.* **333**, 129863. (doi:10.1016/j.jclepro.2021.129863)
33. Yang X, Guo L, Zheng Z, Riemer N, Tessum CW. 2024 Atmospheric chemistry surrogate modeling with sparse identification of nonlinear dynamics. *J. Geophys. Res. Mach. Learn. Comput.* **1**, e2024JH000132. (doi:10.1029/2024jh000132)
34. Rudy S, Alla A, Brunton SL, Kutz JN. 2019 Data-driven identification of parametric partial differential equations. *SIAM J. Appl. Dyn. Syst.* **18**, 643–660. (doi:10.1137/18m1191944)
35. Wang X, Wang H. 2024 Discrete inverse method for extracting disease transmission rates from accessible infection data. *SIAM J. Appl. Math.* **84**, S336–S361. (doi:10.1137/22m1498796)
36. Bousquet A, Conrad WH, Sadat SO, Vardanyan N, Hong Y. 2022 Deep learning forecasting using time-varying parameters of the SIRD model for Covid-19. *Sci. Rep.* **12**, 3030. (doi:10.1038/s41598-022-06992-0)
37. Long J, Khaliq AQM, Furati KM. 2021 Identification and prediction of time-varying parameters of COVID-19 model: a data-driven deep learning approach. *Int. J. Comput. Math.* **98**, 1617–1632. (doi:10.1080/00207160.2021.1929942)
38. Ji J, Ahmed S, Wang H. 2025 A hybrid approach to study and forecast climate-sensitive norovirus infections in the USA. *J. Theor. Biol.* **598**, 112007. (doi:10.1016/j.jtbi.2024.112007)

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

39. Pastpipatkul P, Subsai P. 2024 Analysis of time-varying coefficients and forecasting effects between greenhouse gas emissions and its determinants in thailand. In *Applications of optimal transport to economics and related topics* (eds V Kreinovich, W Yamaka, S Leurcharusmee), pp. 591–604. Cham, Switzerland: Springer.
40. Chakraborty AK, Gao S, Miry R, Ramazi P, Greiner R, Lewis MA, Wang H. 2024 An early warning indicator trained on stochastic disease-spreading models with different noises. *J. R. Soc. Interface* **21**, 20240199. (doi:10.1098/rsif.2024.0199)
41. Kermack WO, McKendrick AG. 1927 A contribution to the mathematical theory of epidemics. *Proc. R. Soc. A* **115**, 700–721. (doi:10.1098/rspa.1927.0118)
42. Rosenzweig ML, MacArthur RH. 1963 Graphical representation and stability conditions of predator-prey interactions. *Am. Nat.* **97**, 209–223. (doi:10.1086/282272)
43. Kloeden PE, Platen E. 1992 *Numerical solution of stochastic differential equations.* Berlin, Germany: Springer.
44. Wood Buffalo Environmental Association. *Network map & station data*. http://wbea.org/data/network-map-station-data/ (accessed 6 November 2024).
45. Alberta Energy Regulator. *Alberta mineable oil sands plant statistics*. http://www.aer.ca/data-and-performance-reports/statistical-reports/st39 (accessed 23 January 2025).
46. Heggerud CM. 2025 Cheggerud/CB-prediction: Nov_10_2023. Zenodo. (doi:10.5281/zenodo.10109225)
47. Heggerud CM, Xu J, Wang H, Lewis MA, Zurawell RW, Loewen CJ, Vinebrooke RD, Ramazi P. 2024 Predicting imminent cyanobacterial blooms in lakes using incomplete timely data. *Water Resour. Res.* **60**, e2023WR035540. (doi:10.1029/2023wr035540)
48. Uhlenbeck GE, Ornstein LS. 1930 On the theory of the Brownian motion. *Phys. Rev.* **36**, 823–841. (doi:10.1103/physrev.36.823)
49. Breiman L. 2001 Random forests. *Mach. Learn.* **45**, 5–32. (doi:10.1023/a:1010933404324)
50. Pedregosa F *et al*. 2011 Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830.
51. Shi X, Chen Z, Wang H, Yeung D-Y, Wong W-K, Woo W-C. 2015 Convolutional LSTM network: a machine learning approach for precipitation nowcasting. 2015 In *Proc. 29th Int. Conf. on Neural Information and Processing Systems*. New York, NY: Curran Associates, Inc.
52. Hochreiter S, Schmidhuber J. 1997 Long short-term memory. *Neural Comput.* **9**, 1735–1780. (doi:10.1162/neco.1997.9.8.1735)
53. Bai S, Kolter JZ, Koltun V. 2018 An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. (doi:10.48550/arXiv.1803.01271)
54. Friedman JH. 2001 Greedy function approximation: a gradient boosting machine. *Ann. Statist.* **29**, 1189–1232. (doi:10.1214/aos/1013203451)
55. Abadi M *et al*. 2016 TensorFlow: a system for large-scale machine learning. In *Proc. 12th USENIX Symp. on Opererating Systems Design and Implementation*, pp. 265–283. Savannah, GA: USENIX Association.
56. Chollet F. 2015 Keras. GitHub repository. http://keras.io.
57. O'Malley T *et al*. 2019 Keras tuner. http://keras.io/keras_tuner.
58. Kingma DP, Ba J. 2014 Adam: a method for stochastic optimization. (doi:10.48550/arXiv.1412.6980)
59. Prechelt L. 1998 Early stopping—but when? In *Neural networks: tricks of the trade* (eds GB Orr, K-R Müller), pp. 55–69. Berlin, Germany: Springer.
60. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. 1953 Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087–1092. (doi:10.1063/1.1699114)
61. Hastings WK. 1970 Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97–109. (doi:10.1093/biomet/57.1.97)
62. Kaipio J, Somersalo E. 2005 *Statistical and computational inverse problems*. New York, NY: Springer.
63. Stuart AM. 2010 Inverse problems: a Bayesian perspective. *Acta Numer.* **19**, 451–559. (doi:10.1017/S0962492910000061)
64. Gelman A, Rubin DB. 1992 Inference from iterative simulation using multiple sequences. *Statist. Sci.* **7**, 457–472. (doi:10.1214/ss/1177011136)
65. Liu F *et al*. 2023 Data assimilation method for improving the global spatiotemporal predictions of epidemic dynamics yielded by an ensemble Kalman filter and Metropolis–Hastings sampling. *Nonlinear Dyn.* **111**, 15 573–15 589. (doi:10.1007/s11071-023-08632-2)
66. Liu F, Li X, Zhu G. 2020 Using the contact network model and Metropolis-Hastings sampling to reconstruct the COVID-19 spread on the 'diamond princess'. *Sci. Bull.* **65**, 1297–1305. (doi:10.1016/j.scib.2020.04.043)
67. Liu F, Zhao Z, Ma C, Nie X, Wu A, Li X. 2022 Return to normal pre-COVID-19 life is delayed by inequitable vaccine allocation and SARS-CoV-2 variants. *Epidemiol. Infect.* **150**, e46. (doi:10.1017/S0950268822000139)
68. Teschl G. 2012 *Ordinary differential equations and dynamical systems*. Providence, RI: American Mathematical Society.
69. Zhang L, Schaeffer H. 2019 On the convergence of the SINDy algorithm. *Multiscale Model. Simul.* **17**, 948–972. (doi:10.1137/18M1189828)
70. Gronwall TH. 1919 Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. *Ann. Math.* **20**, 292–296. (doi:10.2307/1967124)
71. Rudin W. 1987 *Real and complex analysis*, 3rd edn. New York, NY: McGraw-Hill.
72. Hamrani A, Akbarzadeh A, Madramootoo CA. 2020 Machine learning for predicting greenhouse gas emissions from agricultural soils. *Sci. Total Environ.* **741**, 140338. (doi:10.1016/j.scitotenv.2020.140338)
73. Guo P *et al*. 2017 Developing a dengue forecast model using machine learning: a case study in China. *PLoS Negl. Trop. Dis.* **11**, e0005973. (doi:10.1371/journal.pntd.0005973)
74. Saha E, Wang O, Chakraborty AK, Garcia PV, Milne R, Wang H. 2025 Dispersion based recurrent neural network model for methane monitoring in Albertan tailings ponds. *J. Environ. Manage.* **395**, 127748. (doi:10.1016/j.jenvman.2025.127748)
75. Lowen AC, Steel J. 2014 Roles of humidity and temperature in shaping influenza seasonality. *J. Virol.* **88**, 7692–7695. (doi:10.1128/jvi.03544-13)
76. Laws AN. 2017 Climate change effects on predator–prey interactions. *Curr. Opin. Insect Sci.* **23**, 28–34. (doi:10.1016/j.cois.2017.06.010)
77. Tseng SH, Feng PH, Duong THT. 2025 Apply data science and feature selection techniques to predict carbon dioxide emissions in Taiwan. *Stoch. Environ. Res. Risk Assess.* **39**, 5765–5787. (doi:10.1007/s00477-025-03099-6)
78. Spannaus A, Papamarkou T, Erwin S, Christian JB. 2022 Inferring the spread of COVID-19: the role of time-varying reporting rate in epidemiological modelling. *Sci. Rep.* **12**, 10761. (doi:10.1038/s41598-022-14979-0)
79. Song X, Bryan BA, Almeida AC, Paul KI, Zhao G, Ren Y. 2013 Time-dependent sensitivity of a process-based ecological model. *Ecol. Modell.* **265**, 114–123. (doi:10.1016/j.ecolmodel.2013.06.013)
80. Ornik M, Topcu U. 2021 Learning and planning for time-varying MDPs using maximum likelihood estimation. *J. Mach. Learn Res.* **22**, 1–40.
81. Ahmed NK, Atiya AF, Gayar NE, El-Shishiny H. 2010 An empirical comparison of machine learning models for time series forecasting. *Econom. Rev.* **29**, 594–621. (doi:10.1080/07474938.2010.481556)

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026

82. Muter BM, Mohammed AJ. 2024 The future of AI: assessing the strengths and limitations of deep learning and machine learning. *Wisdom J. Stud. Res.* **4**, 348–374. (doi:10.55165/wjf-sar.v4i06.407)
83. Coorssen L, Kalantari H, Afsharnia P, Ramazi P. 2026 Applied Bayesian networks rely on expert knowledge and scarce data sharing. *Mathematics* **14**, 2669. (doi:10.3390/math14152669)
84. Chakraborty AK, Miry R, Greiner R, Lewis MA, Wang H, Guan T, Ramazi P. 2025 Deep learning for disease outbreak prediction: a parallel LSTM-CNN model. *J. R. Soc. Interface* **22**, 20250046. (doi:10.1098/rsif.2025.0046)
85. Bates S, Hastie T, Tibshirani R. 2024 Cross-validation: what does it estimate and how well does it do it? *J. Am. Statist. Assoc.* **119**, 1434–1445. (doi:10.1080/01621459.2023.2197686)
86. Bergmeir C, Hyndman RJ, Koo B. 2018 A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Comput. Stat. Data Anal.* **120**, 70–83. (doi:10.1016/j.csda.2017.11.003)
87. Chakraborty AK. 2026 Data and codes: turning mechanistic models into forecasters by using machine learning. Zenodo. (doi:10.5281/zenodo.18473544)
88. OpenAI. 2023 ChatGPT. Large language model used for proofreading assistance.
89. Chakraborty AK,Wang H, Ramazi P. 2026 Supplementary material from: From data-driven differential equations to mechanistic forecasting with machine learning. Figshare. (doi:10.6084/m9.figshare.c.8705787)

Downloaded from royalsocietypublishing.org/rsif/article-pdf/doi/10.1098/rsif.2026.0240/6175310/rsif.2026.0240.pdf
by guest
on 04 November 2026