*Mathematical and Computational Applications*

MDPI

*Article*

# Dynamic Time Warping as Elementary Effects Metric for Morris-Based Global Sensitivity Analysis of High-Dimension Dynamical Models

**Dhan Lord B. Fortela [1,2,*], Ashley P. Mikolajczyk [1], Rafael Hernandez [1,2], Emmanuel Revellame [1,2], Wayne Sharp [2,3], William Holmes [1,2], Daniel Gang [2,3] and Mark E. Zappi [1,2]**

[1] Department of Chemical Engineering, University of Louisiana at Lafayette, Lafayette, LA 70504, USA; ashley.mikolajczyk@louisiana.edu (A.P.M.); rafael.hernandez@louisiana.edu (R.H.); emmanuel.revellame@louisiana.edu (E.R.); william.holmes@louisiana.edu (W.H.); mark.zappi@louisiana.edu (M.E.Z.)
[2] Energy Institute of Louisiana, University of Louisiana at Lafayette, Lafayette, LA 70504, USA; wayne.sharp@louisiana.edu (W.S.); daniel.gang@louisiana.edu (D.G.)
[3] Department of Civil Engineering, University of Louisiana at Lafayette, Lafayette, LA 70504, USA
* Correspondence: dhanlord.fortela@louisiana.edu

**Abstract:** This work focused on demonstrating the use of dynamic time warping (DTW) as a metric for the elementary effects computation in Morris-based global sensitivity analysis (GSA) of model parameters in multivariate dynamical systems. One of the challenges of GSA on multivariate time-dependent dynamics is the modeling of parameter perturbation effects propagated to all model outputs while capturing time-dependent patterns. The study establishes and demonstrates the use of DTW as a metric of elementary effects across the time domain and the multivariate output domain, which are all aggregated together via the DTW cost function into a single metric value. Unlike the commonly studied coefficient-based functional approximation and covariance decomposition methods, this new DTW-based Morris GSA algorithm implements curve alignment via dynamic programing for cost computation in every parameter perturbation trajectory, which captures the essence of "elementary effect" in the original Morris formulation. This new algorithm eliminates approximations and assumptions about the model outputs while achieving the objective of capturing perturbations across time and the array of model outputs. The technique was demonstrated using an ordinary differential equation (ODE) system of mixed-order adsorption kinetics, Monod-type microbial kinetics, and the Lorenz attractor for chaotic solutions. DTW as a Morris-based GSA metric enables the modeling of parameter sensitivity effects on the entire array of model output variables evolving in the time domain, resulting in parameter rankings attributed to the entire model dynamics.

**Keywords:** multivariate dynamical systems; system identification; functional data analysis



**Citation:** Fortela, D.L.B.; Mikolajczyk, A.P.; Hernandez, R.; Revellame, E.; Sharp, W.; Holmes, W.; Gang, D.; Zappi, M.E. Dynamic Time Warping as Elementary Effects Metric for Morris-Based Global Sensitivity Analysis of High-Dimension Dynamical Models. *Math. Comput. Appl.* **2024**, *29*, 111. https://doi.org/10.3390/mca29060111

Academic Editor: Maria Amélia Ramos Loja

Received: 13 October 2024
Revised: 21 November 2024
Accepted: 26 November 2024
Published: 27 November 2024



## 1. Motivations

Sensitivity analysis (SA) is important in the development and application of dynamical models due to several reasons [1,2]: (i) understanding model behavior—SA helps in understanding how changes in model parameters affect the overall behavior of the system; (ii) identifying critical parameters—sensitivity analysis identifies which parameters have the most significant impact on model outputs, allowing modelers to focus on accurately estimating these critical parameters, improving the model's reliability; (iii) model validation and confidence building—SA aides in building confidence in the model by demonstrating the robustness of the model and highlighting areas where more precise data is needed; (iv) guiding data collection—SA informs data collection efforts by determining which parameters require more accurate measurements, resulting in more efficient use of resources and time during data collection; (v) exploring uncertainty—SA allows modelers to explore the effects of uncertainty in parameter values on the models' predictions.

A widely accepted definition of SA, written in one of the foundational works on the topic by Saltelli, Tarantola, et al. [3], is as follows: "Sensitivity analysis (SA) is the study of how uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input". Given a model $y = g(p)$, SA explores the relationship between the model's $k$ input variables, which are the model parameters $p = [p_1, p_2, \ldots, p_k]$, and $n$ output variables, $y = [y_1, y_2, \ldots, y_n]$, where $g$ is the model that maps the model's inputs to the outputs [3,4]. There are two broad categories of SA: global SA (GSA) and local SA. Local SA involves adjusting model parameters near certain reference points to examine how minor changes in the inputs affect the model's prediction performance. Although widely used because of its low computational demand, local SA is limited in its capability to accurately account for the entire model's behavior, especially if there are several nonlinear terms in the model [5,6]. This is solved by performing GSA that allows for the sampling of model parameters from their entire range of possible values [4,7]. There are two main steps in GSA: (1) sampling of model parameters from the space of allowed values and this results in the perturbation of the model parameters; and (2) computation of the effect of model parameters perturbation. Despite progress in refining various aspects of the GSA workflow for various applications, there are new challenges in GSA when applied to more advanced applications, such as SA of high-dimensional dynamical models [2], and this current work aimed to contribute a solution.

Performing SA on high-dimensional dynamical models has been a challenge due to the functional nature of data, i.e., information is a sequence of data [2]. Handling such functional data during SA has been the focus of numerous studies that can be grouped into two main categories: (1) elementary effects of the coefficients of basis functions [8–10], and (2) variance decomposition [10,11]. However, it is computationally prohibitive to apply these prior SA techniques separately on each time-dependent output in high-dimensional dynamical models [12]. This work provides a solution to this challenge by establishing the theory and demonstrating the use of dynamic time warping (DTW) as a metric of elementary effects in Morris-based GSA. Originating in time series analysis, the DTW algorithm measures the similarity between two sequential data by determining their optimal alignment or matching via dynamic programming [13]. The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity, which is the alignment path with the minimal cost [14]. This makes DTW a fitting technique to be the generalized metric of the Morris GSA for dynamical models that exhibit a range of linear and non-linear effects of the parameters. We propose DTW as an elementary effects metric for Morris-based GSA of high-dimension dynamical models.

## 2. The Proposed DTW-Based Metric of Elementary Effects for Dynamical Systems

Our proposed SA scheme, which we call DTW-based Morris GSA, can be visualized using Figure 1 and Algorithm 1. Due to determination of the minimal cost of alignment, DTW determines not only the path to align the paired sequences, but also the optimal cost value to align the sequences [13]. After alignment via dynamic programming, the difference between the two sequences (or perturbation in the context of SA) is computed using a distance measure, like Euclidean distance, and aggregated by taking the mean of the distances we denote as $C_{DTW}$ as shown in Equation (1), where $a_m$ is the length of the $m^{\text{th}}$ alignment path and $M$ is the total number of paired alignment paths. This makes the optimal cost of alignment $C_{DTW}$ as a measure of perturbation between the two sequences, which is equivalent to the "elementary effect" in the original Morris GSA [15]. This leads us to the following proposition:

$$C_{DTW} = \frac{1}{M} \sum_{m=1}^{M} a_m \tag{1}$$

**Proposition 1.** *We take the dynamic time warping (DTW) cost of alignment between pairs of sequential model outputs as the "elementary effect" to be used in the Morris GSA. By adding conditioning on the multiple outputs, e.g., via normalization along each output dimension, the DTW cost of all model outputs may then be aggregated together, e.g., via summation, averaging, etc., to compute a single metric of model perturbation between two parameter settings; hence, a single metric of perturbation across all model outputs is computed per parameter trajectory in the Morris GSA.*

This proposed SA scheme implements a functional computation of the elementary effect because DTW takes into account the entire sequential data in every model output pair to compute the cost of alignment. In contrast to the prior algorithms implementing basis functions approximation [8] and variance decomposition [11], DTW-based Morris GSA does not approximate the model outputs, nor make assumptions on the sequential data (normality of residuals, independence, etc.) in computing SA index. This makes our proposed algorithm to be the first GSA on dynamical models that does not approximate the model perturbations. We show in this paper the workflow on how to implement a DTW-based Morris GSA.

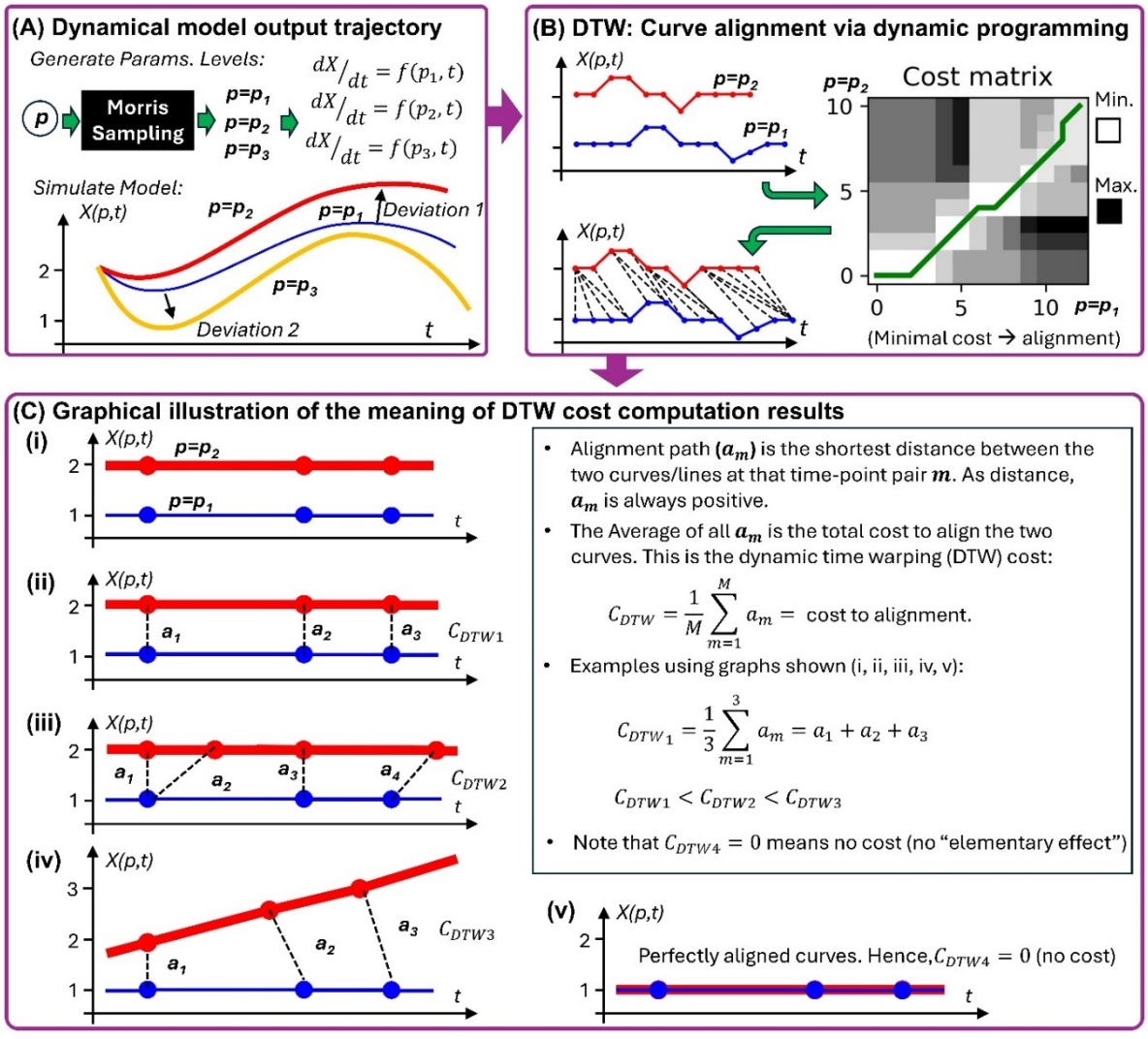



**Figure 1.** Schematic on how DTW is implemented for each pair of model parameter perturbations. (**A**) The resulting data sequences (time curves) of model output between two pairs of parameter settings constitute one trajectory of perturbation (deviation). (**B**) Each pair of perturbed curves undergoes DTW alignment computation, achieved by applying dynamic programming. (**C**) The DTW cost of alignment, $C_{DTW}$, represents the deviation between the curves.

**Algorithm 1: DTW-based Morris GSA**

This is the core DTW-based Morris GSA. Note: The algorithmic innovations in this current work from the base Morris algorithm are in Step 3.
Steps:
1. Initialize the grid jump ($\Delta$), and the number of trajectories ($r$) of each parameter.
2. Simulate model for each trajectory and collect all model output vectors:
   2.1. Randomly generate a starting point in the input parameter space.
   2.2. Simulate model across random trajectories of input parameter:
      2.2.1. Perturb the parameter by $\Delta$, keeping others constant.
      2.2.2. Run the model with the original and perturbed inputs and collect sequential data of model outputs.
3. Calculate the elementary effect of each parameter per trajectory step via DTW:
   3.1. Normalize model output vectors according to each model output's high and low values.
   3.2. Calculate the elementary effect for each parameter per trajectory step via DTW:
      3.2.1. Compute DTW cost ($C_{DTW}$) then the elementary effect in each pair of original and perturbed model outputs.
      3.2.2. Compute the average of the elementary effects of each parameter across all model output dimensions.
4. Calculate the mean ($\mu$) of the elementary effects for each parameter across all trajectories. This number is the SA index for each model parameter subject to GSA.
5. Rank the parameters based on the mean $\mu$ to identify the most influential model parameters.

## 3. Notations and Formulations for DTW-Based Morris GSA

We first present the original Morris formulation to have a reference for the new approach in our proposed DTW-based Morris GSA. Then we show the modifications of the formulations to implement the DTW cost as the new elementary effect metric.

### *3.1. Original Morris GSA Formulation*

The Morris GSA involves the computation of the SA index using the fundamental measure of "elementary effect" as a representation of model output perturbations relative to the corresponding parameter perturbations. Let us denote the $n$ model outputs (high-dimensional) of the system of differential equations to be $\boldsymbol{X} = [X_1, X_2,\ X_3, \ldots, X_n]$, and the $k$ model parameters to be $\boldsymbol{P} = [p_1, p_2,\ p_3, \ldots, p_k]$. So, the system of ordinary differential equations (ODEs) for the dynamical system evolving in time $t$ is represented as follows with the function $f$:

$$\frac{d\boldsymbol{X}}{dt} = f(X, P, t) \tag{2}$$

Note that we are generalizing the ODE model by assuming $f$ to be composed of coupled ODEs where the $X_n$'s may exhibit relations among each other via the mathematical definition of $f$. To compute the "elementary effect", each parameter in $\boldsymbol{P}$ is varied across $r$ levels in the space of the parameters according to the Morris sampling approach [7,15]. The region of Morris sensitivity sampling is a $k$-dimensional $r$-level grid. Each parameter $p_i$ being perturbed by an amount $\Delta$ has its own elementary effect $EE_i$ (Equation (3)) on the model outputs [7], but the original Morris formulation is valid only at a particular $j$th time-point $t_j$ in the time-sequence.

$$[EE_i(\boldsymbol{P})]_{t_j} = \frac{\boldsymbol{X}(p_1, \ldots, p_{i-1}, p_i + \Delta,\ p_{i+1}, \ldots, p_k) - \boldsymbol{X}(\boldsymbol{P})}{\Delta} \tag{3}$$

$$\mu_i = \frac{\sum_r \left| [EE_i(\boldsymbol{P})]_{t_j} \right|}{r} \tag{4}$$

The Morris sensitivity index for $i$th parameter $p_i$ is then computed and we denote the SA index as $\mu_i$ (Equation (4)). High level of $\mu_i$ indicates high SA index, which means very sensitive model outputs relative to parameter $p_i$ perturbations. Note that several prior works were conducted to extend this original Morris formulation to try to capture the elementary effects across the entire sequence of $t$, but these works implemented approxi-

mations of the sequential data of model outputs, as pointed out in the Introduction Section. This paper proposes the DTW-based Morris GSA that eliminates the approximations while achieving the goal of capturing the perturbation patterns across $t$.

### *3.2. DTW-Based Morris GSA Formulation*

Our proposed DTW-based Morris GSA builds on the formulations of the original Morris GSA, as shown above, by replacing the computation of elementary effect with the DTW alignment cost, which captures the perturbations in the model outputs across the time sequence $t$. This means that the model outputs $\boldsymbol{X}$ are now functions of both $\boldsymbol{P}$ and $t$, i.e., $\boldsymbol{X}(\boldsymbol{P}, t)$, when computing the elementary effect. We simplify the notation by replacing the numerator expression representing the perturbation in the model outputs $[\boldsymbol{X}(p_1, \ldots, p_{i-1}, p_i + \Delta,\ p_{i+1}, \ldots, p_k, t) - \boldsymbol{X}(\boldsymbol{P}, t)]$ with the DTW cost when the parameter $p_i$ is perturbed by $\Delta$, which we simply denote as $C_{DTWi}$ (see Figure 1 also)**.** That is, we define the new elementary effect $[EE_i(\boldsymbol{P})]_{DTW}$ as measured by the cost of DTW alignment as shown in Equation (5). The SA index, $\mu_{DTWi}$, is computed in the same manner as the original Morris approach but using the $[EE_i(\boldsymbol{P})]_{DTW}$ as shown in Equation (6). For correct aggregation of the costs across all model outputs into a single $C_{DTWi}$ value, model outputs are normalized within each dimension in $\boldsymbol{X}$ (see Algorithm 1) prior to computing $C_{DTWi}$; hence, prior to computing $[EE_i(\boldsymbol{P})]_{DTW}$.

$$[EE_i(\boldsymbol{P})]_{DTW} = \frac{\boldsymbol{X}(p_1, \ldots, p_{i-1}, p_i + \Delta,\ p_{i+1}, \ldots, p_k, t) - \boldsymbol{X}(\boldsymbol{P}, t)}{\Delta} \stackrel{\text{def}}{=} \frac{C_{DTWi}}{\Delta} \tag{5}$$

$$\mu_{DTWi} = \frac{\sum_r |[EE_i(\boldsymbol{P})]_{DTW}|}{r} \tag{6}$$

### *3.3. Assertion: The Original Morris GSA Is a Special Case of DTW-Based Morris GSA*

This new formulation of SA index computation from the DTW-based elementary effect is a generalization of the original Morris elementary effect when the DTW-based Morris is applied on a single time point $t_j$. This naturally follows from the definition of the DTW-based elementary effect $[EE_i(\boldsymbol{P})]_{DTW}$ as discussed above. This means the original Morris GSA is a special case of one time-point implementation of our proposed DTW-based Morris GSA.

## 4. Methodology

### *4.1. Pseudo-Code of Implementing DTW-Based Morris GSA*

The implementation of DTW-based Morris GSA in this study was conducted by setting different levels of a random number generator index, which varies the sampling trajectories for the model parameters. Algorithm 2 summarizes the workflow for this implementation.

**Algorithm 2: Implement DTW-based Morris GSA at Varying Randomization**

This is the implementation-level algorithm for the DTW-based Morris GSA. Note: RNG seed randomization affects step 2.1 and 2.2 in Algorithm 1.
Steps:
1. Generate a set of seed index for the random number generator (RNG)
2. Collect instances of parameter sensitivity ranks via Algorithm 1
   2.1. Implement Algorithm 1 for each seed index.
   2.2. Append each parameter ranking results until all RNG seeds are implemented.
3. Apply descriptive statistics and aggregation on the all-parameter sensitivity ranks.
4. Apply a set of rank-aggregation techniques to computationally determine the overall ranks of the parameters

### *4.2. Python Code Implementation: DTW-Morris GSA Python Module*

The popular programming language Python was used to implement the computations for this DTW-based Morris GSA. The code scripts are maintained in the open-access online

GitHub repository of the paper [16]. For ease of use and readability of the codes and workflow in this paper, the Python codes were also organized in Jupyter Notebook files, which are also available in the same repository. See the Data Availability Section for more information.

#### 4.2.1. Model Parameter Sampling via Morris

The generation of the model parameters was achieved by using an existing Python code implementation of the Morris sampling written by a prior study [17]. To maintain good integration of our proposed DTW-based Morris GSA, this prior code implementation of Morris sampling was adopted into the Python module we created for this work.

#### 4.2.2. ODE Integration

The ODE integration method implemented was the 'RK45': Explicit Runge-Kutta method of order 5(4) [18]. This was carried out using the Python package 'Scipy' [19] via the function 'solve_ivp()' under the 'integrate' method class: scipy.integrate.solve_ivp(). After testing other integrating methods in the Scipy module, it was found that the RK45 simulated the ODEs in the same fidelity as the more computationally demanding methods in the module (e.g., 'Radau', 'BDF', 'DOP853'). The effect of step size was evaluated in preliminary computations (at 50 runs, 100 runs, and 500 runs) and it was found that the best step size was achieved when the integrating function is run in its default setting of "inferring the best step size", based on the error tolerances, relative error (or "rtol" in the module code), and absolute error (or "atol" in the module code) when running the RK45. It must be noted, however, that the another argument in the "scipy.integrate.solve_ivp()" function the "t_eval" is different from the step size. The "t_eval" specifies the time-points to "store" the computed solution values, but the step size of evaluation is separate and is based on the specified "atol" and "rtol", as explained above. The defaults values of rtol and atoll are: rtol = $1 \times 10^{-3}$ and atoll = $1 \times 10^{-6}$ [19].

#### 4.2.3. GSA Index Computation

The computation of the GSA index was achieved via the proposed DTW-based Morris (Algorithms 1 and 2) implemented in a Python module developed for this study. This module we call 'DTWMorrisGSA' is available as an open source code written in Python hosted in the online repository of the paper [16]. This module is also being packaged for deployment in the PyPi repository (currently the module is in the TestPyPi repository) for easier installation by interested users.

### *4.3. Example Time-Series Dynamical Models Tested*

The set of examples chosen for the testing of the proposed algorithm showcases a progression of the complexity of the dynamical models. The first example involves a single model output, the second example involves three model outputs, and the third example involves a set of solutions to chaotic systems. This progression allows the evaluation of the advances the proposed algorithm brings and also allows for the elucidation of potential limitations.

#### 4.3.1. Example 1: Mixed-Order Adsorption Kinetics—Single-Output Dynamics

The first example tested is the mixed-order adsorption kinetics model involving a single differential equation of the adsorption capacity ($q$) [20,21]. Following is the set of ODE system (Equation (7)) for this adsorption dynamics. There are three parameters in the model and all are subjected to GSA via the proposed algorithm: $k_1$ = first-order rate constant, $k_2$ = second-order rate constant, and $q_e$ = adsorption capacity at equilibrium [22].

$$\frac{dq}{dt} = k_1(q_e - q) + k_2(q_e - q)^2 \tag{7}$$

4.3.2. Example 2: Microbial Growth Kinetics—Multiple-Output Dynamics

The second example tested is a Monod-type microbial kinetics that models the growth of microbial cells ($C$), the depletion of the substrate ($S$) during biogrowth, and the production of extracellular product ($P$) during biogrowth. That is, $\boldsymbol{X} = [C\ S\ P]^T$ having three dimensions with each variable expressed in its ODE form with respect to time $t$. The specific model used is for the conversion of glucose to ethanol by *Saccharomyces cerevisiae* [23]. Following is the set of differential-algebraic ODE system for this biogrowth dynamics. There are six parameters in the model and all are subjected to GSA via the proposed algorithm: $k_d$ = cell natural death rate constant, $Y_{s/c}$ = substrate-per-cell yield coefficient, $Y_{p/c}$ = product-per-cell yield coefficient, $K_s$ = substrate saturation constant, $k_{sm}$ = cell maintenance utilization coefficient, and $\mu_{max}$ = maximum specific growth rate. The model is shown in Equations (8)–(14).

$$\frac{dC}{dt} = r_g - r_d \tag{8}$$

$$\frac{dS}{dt} = -r_g Y_{s/c} - r_{sm} \tag{9}$$

$$\frac{dP}{dt} = r_g Y_{p/c} \tag{10}$$

where:

$$r_g = k_{obs} C \frac{S}{K_s + S} \tag{11}$$

$$k_{obs} = \mu_{max} \left(1 - \frac{P}{93}\right)^{0.52} \tag{12}$$

$$r_d = k_d C \tag{13}$$

$$r_{sm} = k_{sm} C \tag{14}$$

4.3.3. Example 3: Lorenz Attractor—A Set of Chaotic Solutions

The third example tested is the Lorenz attractor ODE system [24] that is commonly used as a benchmark ODE for similar works on the analysis of dynamical models due to its characteristic chaotic behavior that occurs with certain combinations of model parameter values [25]. The Lorenz system is a foundational dynamical model in the areas of chaos theory and weather modeling such as atmospheric convection [26]. The Lorenz ODE system consists of three variables we denote as $X_1$ = rate of convection, $X_2$ = rate of horizontal temperature variation, and $X_3$ = rate of vertical temperature variation. Following is the set of differential-algebraic ODE system for this chaos dynamics. That is, $X = [X_1\ X_2\ X_3]^T$ having three dimensions with each variable expressed in its ODE form with respect to time $t$. There are three parameters in the model and all are subjected to GSA via the proposed algorithm: $\alpha$, $\beta$, and $\gamma$, which are parameters respectively proportional to the Prantl number, Rayleigh number, and certain physical dimensions of the atmospheric layer itself [25]. The model is shown in Equations (15)–(17).

$$\frac{dX_1}{dt} = \alpha(X_2 - X_1) \tag{15}$$

$$\frac{dX_2}{dt} = X_1(\beta - X_3) - X_2 \tag{16}$$

$$\frac{dX_3}{dt} = X_1 X_2 - \gamma X_3 \tag{17}$$

*4.4. Analysis and Aggregation of Model Parameter Sensitivities*

Existing data analysis and data aggregation techniques are used to evaluate the performance of the proposed DTW-based Morris GSA, and to make conclusions about the sensitivities of the model parameters in Example 1, Example 2, and Example 3. For each

run of the DTW-based Morris algorithm, the parameter sensitivity values are used to rank the parameters, with Rank 1 assigned to the most sensitive (highest value). With the seed index of the random number generator (RNG) varied in the study of each model, a set of parameter ranking is created for every RNG index. After running the set total of varied RNG index (Algorithm 2), a set of parameter rank list is consequently created. Graphical analysis on the parameter rank list is conducted to evaluate sensitivity trends. Finally, a set of rank-aggregation techniques is implemented on each model parameter rank list to computationally determine the overall ranks of the parameters. The techniques include: Borda's rank aggregation method (Borda), and the Cross Entropy Monte Carlo rank aggregation method (CEMC), which are all implemented using an existing R-package 'TopKLists' [27]. These methods were selected based on their long-established performance and their suitability according to the data structure of the sensitivity indices from the DTW-based Morris GSA. Furthermore, the Borda method is a popular type of non-optimization-based method and the CEMC is a type of optimization-based method [28].

## 5. Results and Discussion

The results show the successful implementation of our proposed DTW-based Morris GSA for high-dimensional dynamical models.

### *5.1. Example 1: Mixed-Order Adsorption Kinetics—Single-Output Dynamics*

The results are shown in Figure 2. The sample adsorption kinetics curves in Figure 2A follow the typical patterns of adsorption of various adsorbate-adsorbent pairs. The rank-aggregation results (Figure 2B,C) on the parameter SA index rankings show consistent overall rankings of the parameters with $k_1$ as Top1 indicating it is the most sensitive model parameter. Parameter $k_2$ is Top2 and $q_e$ as the least sensitive parameter being Top3.

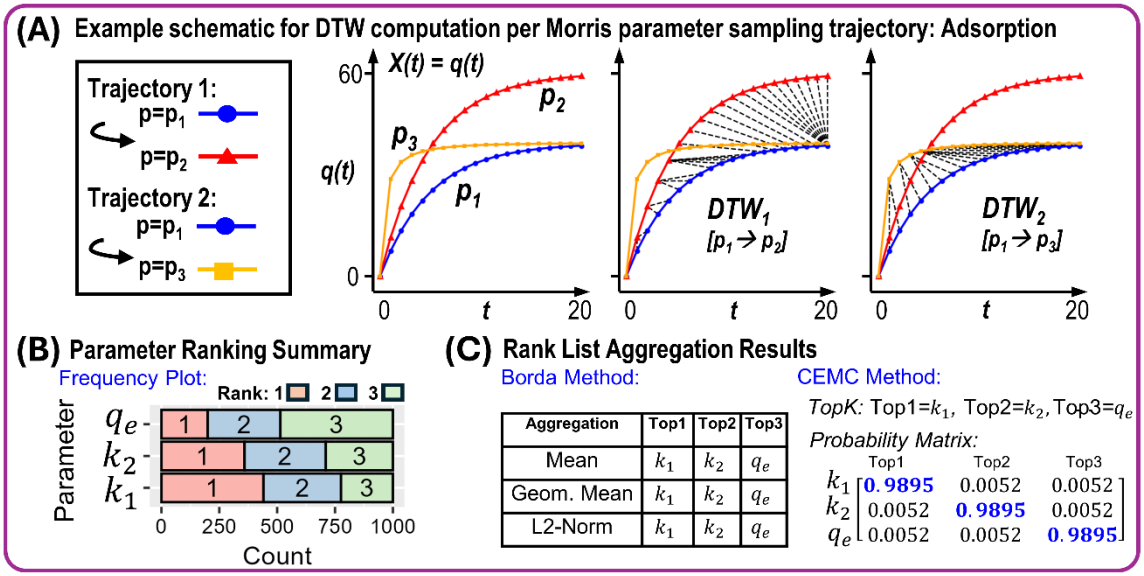



**Figure 2.** Results in implementing the DTW-based Morris GSA on a single-output dynamics using the example of mixed-order adsorption kinetics: (**A**) example natural curves from the simulated adsorption kinetics showing sample two trajectories among several trajectories during runs; (**B**) descriptive summary of the model parameter ranking based in SA index; (**C**) Borda method and CEMC method rank-aggregation results. First, the Morris sampling approach was applied to create the set of model parameter values that represent the perturbation in the parameter values creating the sampling trajectory such that $p_1$ = [0.2, 0.0, 37.65862069] to $p_2$ = [0.2, 0.0, 57.65862069] perturbation creates the Trajectory 1 and $p_1$ = [0.2, 0.0, 37.65862069] to $p_3$ = [0.2, 0.05655172, 37.65862069] (where $p_i$ = [$k_1$, $k_2$, $q_e$]) perturbation creates the Trajectory 2 as shown in (**A**). Then, the model is simulated

using these model parameter values and the DTW alignment cost is computed for each pair of curves in a trajectory, and the graphics in (**A**) show the $DTW_1$ as alignment cost for Trajectory 1 and $DTW_2$ as alignment cost for Trajectory 2. The DTW alignment cost for each trajectory was then used to compute the elementary effects, as shown in the DTW-based Morris GSA equations above, and the resulting GSA index values were used to rank the model parameters, with Rank 1 assigned to the parameter with highest GSA index. Finally, the parameter rankings were analyzed graphically as shown in (**B**) and aggregated using rank-aggregation methods Borda and CEMC as shown in (**C**).

### *5.2. Example 2: Microbial Growth Kinetics—Multiple-Output Dynamics*

The results are shown in Figure 3. The sample microbial kinetics curves in Figure 3A follow the typical patterns of cell growth ($C$), substrate consumption ($S$), and product accumulation ($P$). Evident in Figure 3B is the effect of normalizing the model outputs along each output dimension. That is, $\hat{C}$ consists of curves of $C$ normalized based on the lower-bound $C_{min}$ and upper-bound $C_{max}$ values of $C$: $\hat{C} = C/(C_{max} - C_{min})$. The same was performed on the other model outputs: $\hat{S} = S/(S_{max} - S_{min})$ and $\hat{P} = P/(P_{max} - P_{min})$. This normalization is a critical step to make sure the succeeding aggregation of the $C_{DTWi}$ across all model outputs is correct. This normalization step has been coded in our Python script implementing DTW-based Morris GSA.

The rank-aggregation results (Figure 3C–E) on the parameter SA index rankings show consistent overall rankings of the four most sensitive parameters via the Borda method (Figure 3D): $\mu_{max}$ as Top1, $k_{sm}$ as Top2, $K_s$ as Top3, and $Y_{S/C}$ as Top4. On the other hand, the lower-rank parameters $Y_{P/C}$, and $k_d$ changed overall ranking depending on the rank-aggregation metric—mean, geometric mean, and L2-norm (Figure 3D). The CEMC method results in almost the same overall parameter ranking aggregation as Borda except for the parameter $k_{sm}$ as Top1, and $\mu_{max}$ as Top2 (Figure 3E).

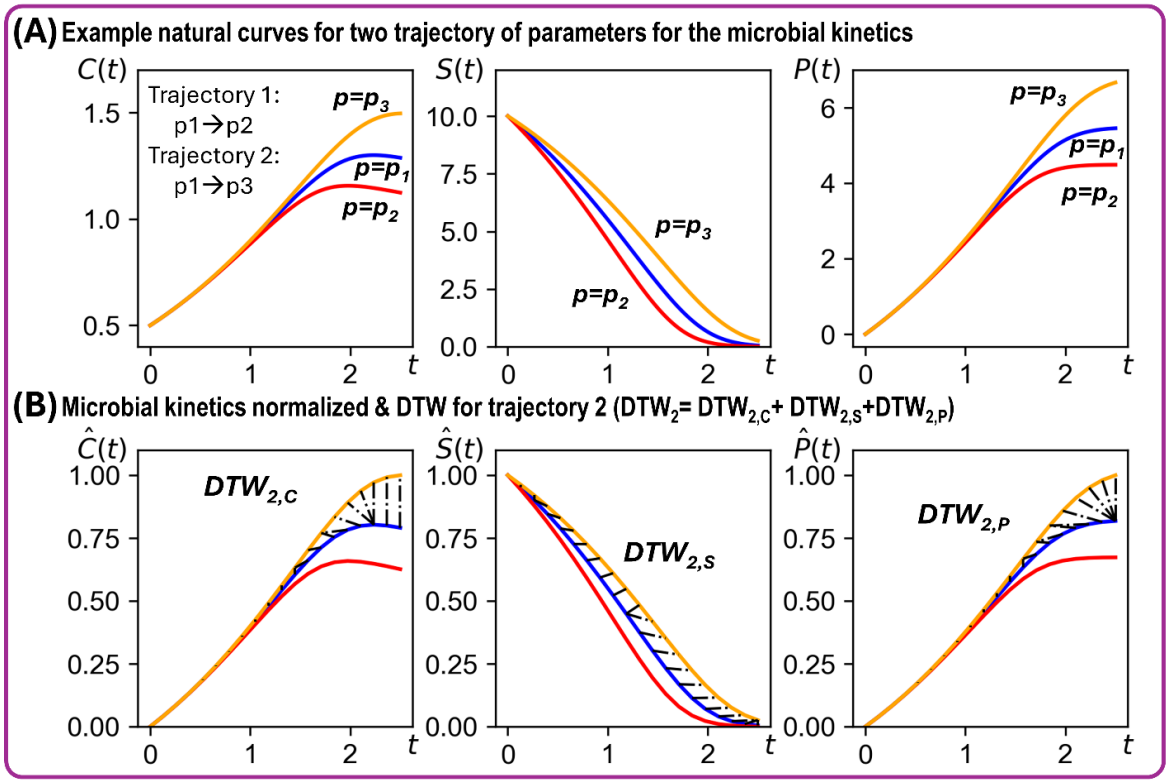
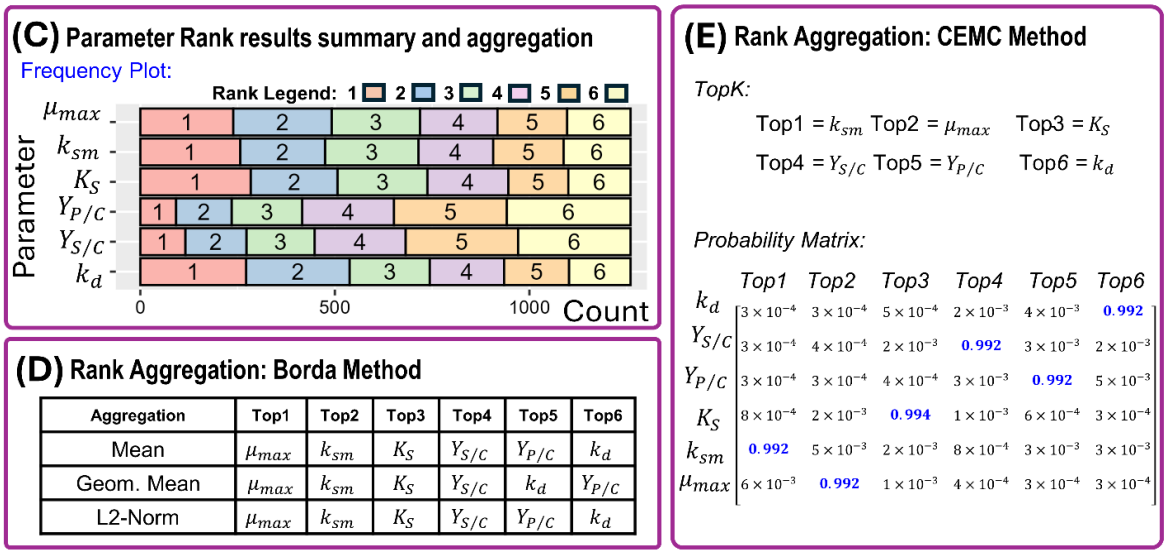



**Figure 3.** *Cont.*




| Aggregation | Top1 | Top2 | Top3 | Top4 | Top5 | Top6 |
|---|---|---|---|---|---|---|
| Mean | $\mu_{max}$ | $k_{sm}$ | $K_S$ | $Y_{S/C}$ | $Y_{P/C}$ | $k_d$ |
| Geom. Mean | $\mu_{max}$ | $k_{sm}$ | $K_S$ | $Y_{S/C}$ | $k_d$ | $Y_{P/C}$ |
| L2-Norm | $\mu_{max}$ | $k_{sm}$ | $K_S$ | $Y_{S/C}$ | $Y_{P/C}$ | $k_d$ |

**Figure 3.** Results in implementing the DTW-based Morris GSA on a multiple-output dynamics using the example of microbial growth kinetics. (**A**) Example natural curves from the simulated Monod-type microbial kinetics with color lines blue for $p_1$, red for $p_2$, and yellow for $p_3$. (**B**) Example of the normalized model outputs annotated with the optimally aligned path for each trajectory. (**C**) Descriptive summary of the model parameter ranking based in SA index. (**D**) Borda method rank-aggregation results. (**E**) CEMC rank-aggregation results with highlights in blue color for the highest probability value per TopK rank.

### *5.3. Example 3: Lorenz Attractor: A Set of Chaotic Solutions*

The results are shown in Figure 4. The sample Lorenz attractor curves shown in Figure 4A,B follow the typical patterns of Lorenz chaotic solutions. An example of the normalized model output $\hat{X}_3$ is shown Figure 4C. The rank-aggregation results (Figure 4D,E) on the parameter SA index rankings show consistent overall rankings of the parameters, with $\alpha$ as Top1 indicating it is the most sensitive model parameter. Parameter $\beta$ is Top2 and $\gamma$ as the least sensitive parameter being Top3. Being a complex dynamical model due to its chaotic nature, the Lorenz Attractor was conveniently subjected to the DTW-based Morris GSA. After a careful search of the literature, we concluded that this is the first time that the parameters sensitivities were computed for the Lorenz Attractor dynamical model, which is an indication of the potential of our proposed approach.

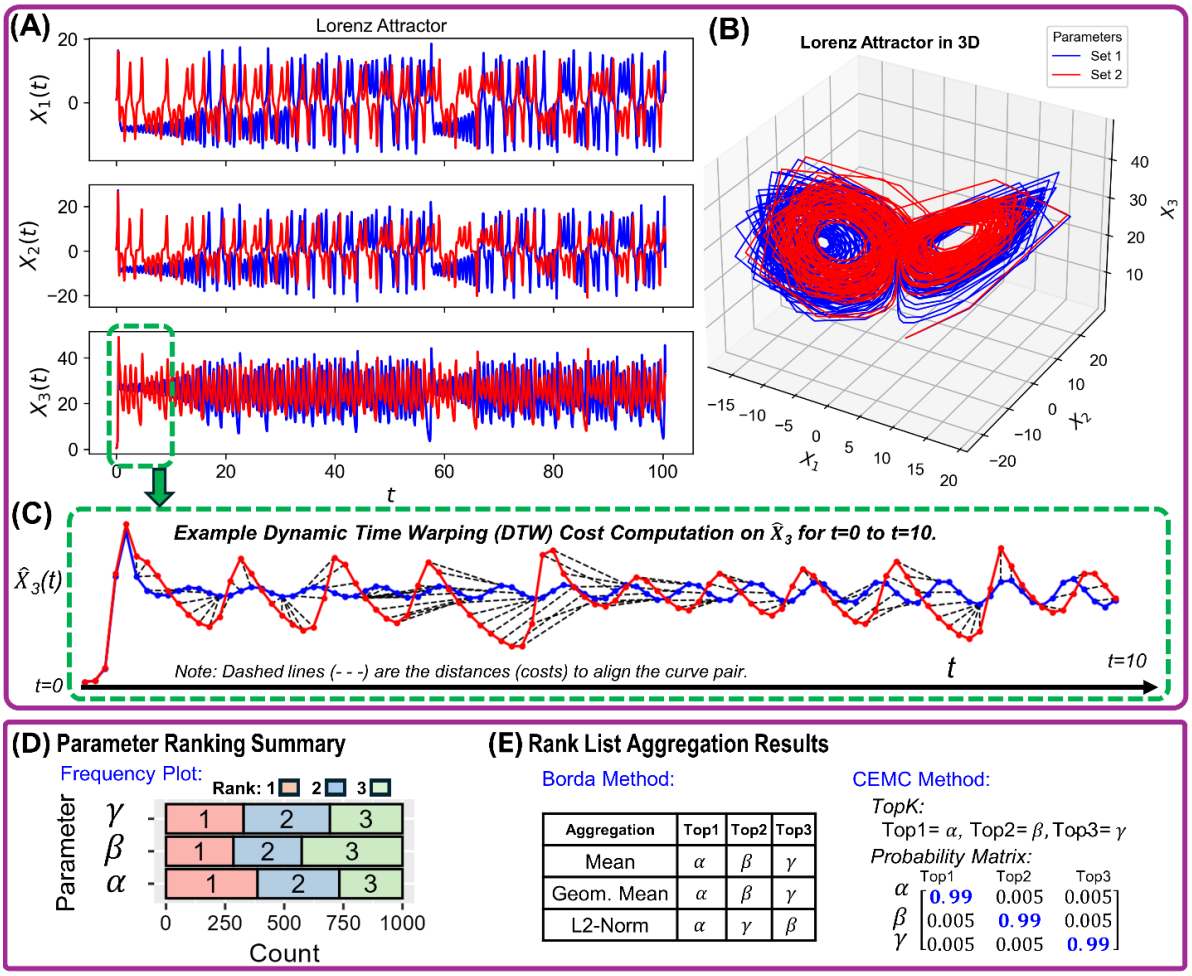



**Figure 4.** Results in implementing the DTW-based Morris GSA on a set of chaotic solutions using the example for the Lorenz Attractor. (**A**) Example natural curves from the simulated Lorenz Attractor. (**B**) Three-dimensional rendering of the model outputs. (**C**) Normalized curves prior to computation of DTW alignment cost. (**D**) Descriptive summary of the model parameter ranking based in SA index. (**E**) Borda method and CEMC method rank-aggregation results.

## 6. Conclusions

DTW-based Morris GSA can successfully determine parameter sensitivities of high-dimensional dynamical models. The results verify the capabilities of our proposed DTW-based Morris GSA. The findings also open the possibilities of extending the technique to more complex dynamical systems and to other areas that may benefit from the computational capabilities of the technique. Here are potential future research directions to further evaluate the capabilities and limitations of our proposed method: (1) Analysis of deep neural networks (DNNs), which are crucial components of many popular advanced deep learning (DL) models, to develop explanations of predictions and the behavior of DL models during training and prediction [29]. A popular topic in this area currently is the examination on how large language models (LLMs) learn through their DNN components [30]. Since LLMs learn using sequences of data, the DTW-based Morris GSA can be applied to examine the model parameters inside DNNs as sequential data are assimilated by LLMs. (2) Sensitivity analysis of discrete chaotic maps, such as adaptive symmetric Hénon maps [31]. A Hénon map provides a way to conduct more detailed exploration of the chaotic dynamics and there are open problems in this research area [32]. (3) Analysis of chaos-based communication systems that use chaotic systems as an efficient, deterministic high-entropy source that can mask information signals for safe transmission through public

channels [33]. Finally, the immediate next directions from this current work will also involve the further refinement of the Python module 'DTWMorrisGSA' [16] so it will allow easy integration with existing modules in open-source repositories (e.g., PyPi, Conda, etc.), ready for use by various users.

**Author Contributions:** Conceptualization, D.L.B.F., A.P.M. and W.S.; methodology, D.L.B.F. and A.P.M.; software, D.L.B.F.; validation, D.L.B.F. and A.P.M.; formal analysis, D.L.B.F., A.P.M., R.H., E.R., W.S., W.H., D.G. and M.E.Z.; resources, D.L.B.F., R.H. and M.E.Z.; data curation, D.L.B.F.; writing—D.L.B.F., A.P.M. and W.S.; writing—review and editing, R.H., E.R., W.H., D.G. and M.E.Z.; visualization, D.L.B.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The Python package and Jupyter Notebook files used to implement the DTW-based Morris GSA developed in this paper is provided as an open-source material through the GitHub repository of the project [16] (accessed on 21 November 2024): https://github.com/dhanfort/DTW_based_Morris_GSA.git.

**Acknowledgments:** This work was conducted with the support of the Energy Institute of Louisiana (EIL) at the University of Louisiana at Lafayette. We are greatly appreciative of EIL's administrative staff, Sheila Holmes, for making things happen when research projects face challenges during implementation.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Murray-Smith, D.J. Sensitivity Analysis for Model Evaluation. In *Testing and Validation of Computer Simulation Models: Principles, Methods and Applications*; Murray-Smith, D.J., Ed.; Springer International Publishing: Cham, Switzerland, 2015; pp. 49–60.
2. Zhang, K.; Lu, Z.; Cheng, K.; Wang, L.; Guo, Y. Global sensitivity analysis for multivariate output model and dynamic models. *Reliab. Eng. Syst. Saf.* **2020**, *204*, 107195. [CrossRef]
3. Saltelli, A.; Tarantola, S.; Campolongo, F.; Ratto, M. *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*; Wiley: New York, NY, USA, 2004.
4. Sobol′, I.M. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Math. Comput. Simul.* **2001**, *55*, 271–280. [CrossRef]
5. Rakovec, O.; Hill, M.C.; Clark, M.P.; Weerts, A.H.; Teuling, A.J.; Uijlenhoet, R. Distributed Evaluation of Local Sensitivity Analysis (DELSA), with application to hydrologic models. *Water Resour. Res.* **2014**, *50*, 409–426. [CrossRef]
6. Saltelli, A.; Annoni, P. How to avoid a perfunctory sensitivity analysis. *Environ. Model. Softw.* **2010**, *25*, 1508–1517. [CrossRef]
7. Campolongo, F.; Cariboni, J.; Saltelli, A. An effective screening design for sensitivity analysis of large models. *Environ. Model. Softw.* **2007**, *22*, 1509–1518. [CrossRef]
8. Campbell, K.; McKay, M.D.; Williams, B.J. Sensitivity analysis when model outputs are functions. *Reliab. Eng. Syst. Saf.* **2006**, *91*, 1468–1472. [CrossRef]
9. Fortela, D.L.B.; Farmer, K.; Zappi, A.; Sharp, W.W.; Revellame, E.; Gang, D.; Zappi, M. A Methodology for Global Sensitivity Analysis of Activated Sludge Models: Case Study with Activated Sludge Model No. 3 (ASM3). *Water Environ. Res.* **2019**, *91*, 865–876. [CrossRef]
10. Cosenza, A.; Mannina, G.; Vanrolleghem, P.A.; Neumann, M.B. Global sensitivity analysis in wastewater applications: A comprehensive comparison of different methods. *Environ. Model. Softw.* **2013**, *49*, 40–52. [CrossRef]
11. Gamboa, F.; Janon, A.; Klein, T.; Lagnoux, A. Sensitivity indices for multivariate outputs. *Comptes Rendus Math.* **2013**, *351*, 307–310. [CrossRef]
12. Li, L.; Papaioannou, I.; Straub, D. Efficient global sensitivity analysis method for dynamic models in high dimensions. *Int. J. Numer. Methods Eng.* **2024**, *125*, e7494. [CrossRef]
13. Gold, O.; Sharir, M. Dynamic Time Warping and Geometric Edit Distance: Breaking the Quadratic Barrier. *ACM Trans. Algorithms* **2018**, *14*, 1–17. [CrossRef]
14. Jeong, Y.-S.; Jeong, M.K.; Omitaomu, O.A. Weighted dynamic time warping for time series classification. *Pattern Recognit.* **2011**, *44*, 2231–2240. [CrossRef]
15. Morris, M.D. Factorial Sampling Plans for Preliminary Computational Experiments. *Technometrics* **1991**, *33*, 161–174. [CrossRef]
16. Fortela, D.L. DTW_Based_Morris_GSA: GitHub Repository of DTW-Based Morris GSA for Dynamical Systems. Available online: https://github.com/dhanfort/DTW_based_Morris_GSA.git (accessed on 21 November 2024).
17. Cuntz, M.; Mai, J.; Zink, M.; Thober, S.; Kumar, R.; Schäfer, D.; Schrön, M.; Craven, J.; Rakovec, O.; Spieler, D.; et al. Computationally inexpensive identification of noninformative model parameters by sequential screening. *Water Resour. Res.* **2015**, *51*, 6417–6441. [CrossRef]

18. Dormand, J.R.; Prince, P.J. A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.* **1980**, *6*, 19–26. [CrossRef]
19. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef]
20. Wang, J.; Guo, X. Adsorption kinetic models: Physical meanings, applications, and solving methods. *J. Hazard. Mater.* **2020**, *390*, 122156. [CrossRef]
21. Mikolajczyk, A.P.; Fortela, D.L.B.; Berry, J.C.; Chirdon, W.M.; Hernandez, R.A.; Gang, D.D.; Zappi, M.E. Evaluating the Suitability of Linear and Nonlinear Regression Approaches for the Langmuir Adsorption Model as Applied toward Biomass-Based Adsorbents: Testing Residuals and Assessing Model Validity. *Langmuir* **2024**, *40*, 20428–20442. [CrossRef]
22. Guo, X.; Wang, J. A general kinetic model for adsorption: Theoretical analysis and modeling. *J. Mol. Liq.* **2019**, *288*, 111100. [CrossRef]
23. Fogler, H.S. Chapter 9: Reaction Mechanisms, Pathways, Bioreactions, and Bioreactors. In *Elements of Chemical Reaction Engineering*, 6th ed.; Pearson: Harlow, UK, 2021.
24. Lorenz, E.N. Deterministic Nonperiodic Flow. *J. Atmos. Sci.* **1963**, *20*, 130–141. [CrossRef]
25. Sparrow, C. The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors. In *Applied Mathematical Sciences*; Springer: New York, NY, USA, 1982.
26. Shen, B.-W.; Pielke, R.; Zeng, X.; Cui, J.; Faghih-Naini, S.; Paxson, W.; Kesarkar, A.; Zeng, X.; Atlas, R. The Dual Nature of Chaos and Order in the Atmosphere. *Atmosphere* **2022**, *13*, 1892. [CrossRef]
27. TopKLists: Inference, Aggregation and Visualization for Top-K Ranked Lists. CRAN (Comprehensive R Archive Network). 2022. Available online: https://cran.r-project.org/web/packages/TopKLists/index.html (accessed on 21 November 2024).
28. Li, X.; Wang, X.; Xiao, G. A comparative study of rank aggregation methods for partial and top ranked lists in genomic applications. *Brief. Bioinform.* **2019**, *20*, 178–189. [CrossRef] [PubMed]
29. Zhou, B.; Bau, D.; Oliva, A.; Torralba, A. Interpreting Deep Visual Representations via Network Dissection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 2131–2145. [CrossRef] [PubMed]
30. Mitchell, M.; Krakauer, D.C. The debate over understanding in AI's large language models. *Proc. Natl. Acad. Sci. USA* **2023**, *120*, e2215907120. [CrossRef] [PubMed]
31. Tutueva, A.V.; Moysis, L.; Rybin, V.G.; Kopets, E.E.; Volos, C.; Butusov, D.N. Fast synchronization of symmetric Hénon maps using adaptive symmetry control. *Chaos Solitons Fractals* **2022**, *155*, 111732. [CrossRef]
32. de Hénon, J.X. Hénon Maps: A List of Open Problems. *Arnold Math. J.* **2024**, *10*. [CrossRef]
33. Grzybowski, J.M.V.; Eisencraft, M.; Macau, E.E.N. Chaos-Based Communication Systems: Current Trends and Challenges. In *Applications of Chaos and Nonlinear Dynamics in Engineering—Vol. 1*; Banerjee, S., Mitra, M., Rondoni, L., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 203–230.