Dimensionless embedding for nonlinear time series analysis

Yoshito Hirata and Kazuyuki Aihara

Phys. Rev. E **96**, 032219 — Published 19 September 2017

DOI: 10.1103/PhysRevE.96.032219

Dimensionless embedding for nonlinear time series analysis

Yoshito Hirata* and Kazuyuki Aihara

Institute of Industrial Science, The University of Tokyo,

4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan

*yoshito@sat.t.u-tokyo.ac.jp.

Abstract

Recently, infinite-dimensional delay coordinates (InDDeCs, pronounced as “index”) have been proposed for predicting high-dimensional dynamics instead of conventional delay coordinates. Although InDDeCs can realize faster computation and more accurate short-term prediction, it is still not well-known whether InDDeCs can be used in other applications of nonlinear time series analysis in which reconstruction is needed for the underlying dynamics from a scalar time series generated from a dynamical system. Here, we give theoretical support for justifying the use of InDDeCs, and provide numerical examples to show that InDDeCs can be used for various applications for obtaining the recurrence plots, correlation dimensions, and maximal Lyapunov exponents, as well as testing directional couplings and extracting slow-driving forces. We demonstrate performance of the InDDeCs using the weather data. Thus, InDDeCs can eventually realize “dimensionless embedding” while we enjoy faster and more reliable

computations.

I. INTRODUCTION

Reconstruction of the underlying dynamics is the first step to analyse its behaviour based on the limited observations, namely a time series. A very popular approach used over the last 35 years has been to reconstruct states for the underlying dynamics by using delay coordinates [1,2]. Delay coordinates collectively represent a vector constructed by arraying successive observations of a time series. Given a time series generated by a continuous-time dynamical system, we need to choose two parameters; namely, the time delay and the embedding dimension. The common rules of thumb for choosing the time delay and the embedding dimension for the last 25 years have been to apply, for example, the first minimum of mutual information [3] and the false nearest neighbour method [4], respectively.

Two years ago, we proposed an alternative approach by using infinite-dimensional delay coordinates [5] (InDDeCs) for reconstructing the underlying dynamics by extending weighted delay coordinates [6-9]. In InDDeCs, we virtually consider an infinite-dimensional vector whose components decay exponentially when they become older. We can access these virtual coordinates by recycling the previous distances to calculate the current distances. In Ref. [5], we demonstrated that InDDeCs have three advantages compared with the conventional delay coordinates: (i) the ability to take into account high-dimensional dynamics; (ii) faster computation; and (iii) more accurate short-term prediction. However, it is not currently well-known whether InDDeCs can be used for the other applications of nonlinear time series analysis.

Therefore, we provide herein the theoretical justifications for why InDDeCs may be used to reconstruct the underlying dynamics instead of the conventional delay coordinates, as well as providing numerical examples for other applications. We demonstrate our method using the weather data at Akita, Japan. In other words, InDDeCs realize an "embedding" without considering the dimensions explicitly. Our assumption here is that we need to obtain a distance matrix for a given time series in applications; If we only need to obtain distances for neighbouring points, we may use another approach, such as the *k-d* tree [10].

II. RESULTS

A. Theorems

To state our theoretical results more rigorously, we formally introduce our current mathematical setup. We consider a dynamical system $f : M \to M$ of a diffeomorphism on an

m-dimensional manifold M and its observation function $g: M \to R$. Then, delay coordinates can be written as $G_d(x) = (g(x), g(f^{-1}(x)), g(f^{-2}(x)), \ldots, g(f^{-d+1}(x)))$. If $d \geq 2m+1$, it is a generic property [1] that x and $G_d(x)$ are one-to-one on M. This theorem by Takens [1] has been extended by using the box-counting dimension [2] and for a forced system [11]. On the other hand, InDDeCs can be written as $H_\lambda(x) = (g(x), \lambda g(f^{-1}(x)), \lambda^2 g(f^{-2}(x)), \ldots)$ where we need to enforce $\lambda \in (0,1)$ so that the following L_1 metric between $H_\lambda(x)$ and $H_\lambda(y)$ converges:

$$\|H_\lambda(x) - H_\lambda(y)\|_{L_1} = \sum_{c=0}^{\infty} \lambda^c \left| g(f^{-c}(x)) - g(f^{-c}(y)) \right|. \qquad (1)$$

It is easy to see that the L_1 metric $\|H_\lambda(f(x)) - H_\lambda(f(y))\|_{L_1}$ for a step ahead can be calculated by using $\|H_\lambda(x) - H_\lambda(y)\|_{L_1}$ by

$$\|H_\lambda(f(x)) - H_\lambda(f(y))\|_{L_1} = \sum_{c=0}^{\infty} \lambda^c \left| g(f^{-c+1}(x)) - g(f^{-c+1}(y)) \right|$$

$$= \left| g(f(x)) - g(f(y)) \right| + \lambda \|H_\lambda(x) - H_\lambda(y)\|_{L_1}. \qquad (2)$$

In reality, as shown below, we can speed up the calculations for a distance matrix and obtain a recurrence plot [12,13] by using Eq. (2). We may introduce a time delay for defining $H_\lambda(x)$ as the common practice, as Ref. [3] does for the conventional delay coordinates.

Then, the following two theorems hold:

Theorem 1 (One-to-one). If x and $G_d(x)$ are one-to-one, then $G_d(x)$ and $H_\lambda(x)$ are one-to-one, and thus x and $H_\lambda(x)$ are one-to-one.

Theorem 2 (equivalence of metrics). Let $\underline{L}$ be the infimum for the local minimum Lyapunov exponent for the system. If $\lambda < e^{\underline{L}}$, then the L_1 metric for $H_\lambda(x)$ is bounded from above and below by the L_1 metric for $G_d(x)$. Namely, under the condition, the L_1 metric for $H_\lambda(x)$ is equivalent to the L_1 metric for $G_d(x)$.

Refer to Appendix A for the proofs of these theorems. Due to Theorems 1 and 2, it is reasonable to assume theoretically that one can calculate the dynamical invariants for the system, such as the correlation dimension [14] and maximal Lyapunov exponent [15,16] by using InDDeCs.

B. Obtaining a distance matrix

To evaluate the correlation dimension and maximal Lyapunov exponent, as well as obtaining a recurrence plot, the calculation for a distance matrix is necessary, within which we can find a distance between two states corresponding to any pair of time points. Thus, we implemented the calculation as discussed in Appendix B.1, and in the following applications, we replace the calculation of a distance matrix using the conventional delay coordinates by using InDDeCs.

By using InDDeCs, we can obtain a recurrence plot [12,13], which represents a two-dimensional plot originally proposed for visualizing time series data. Both axes correspond to the same time axes of the time series. For every pair of time points, we evaluate whether the corresponding states are close to each other. If and only if they are sufficiently close, we plot a point at the corresponding place in the two-dimensional plot.

C. Recurrence plots

We compared recurrence plots obtained from a scalar time series generated from the Rössler model [17] using InDDeCs and the conventional three-dimensional delay coordinates (Compare Figs. 1a, 1d, and 1f with Figs. 1b, 1e, and 1h). We found that these recurrence plots look very similar to each other (Fig. 2): Namely, if λ is small and close to 0, the recurrence plot obtained by the InDDeCs looks similar to that obtained by low-dimensional delay coordinates, while if λ is large and close to 1, the recurrence plot obtained by the InDDeCs looks similar to that obtained by high-dimensional delay coordinates (Figs. 1 and 2). However, InDDeCs have some advantages in the computational times required (Fig. 3). For example, when $\lambda = 0.5$, InDDeCs needed 3.31 ± 0.16 seconds on average to calculate the distance matrix when a given time series is at the length 10000, while the conventional three-dimensional delay coordinates using the L_1 metric needed 3.62 ± 0.12 seconds on average even if we used the decomposition of Eq. (4) shown in Ref. [5]. When we did not use such decomposition and applied the conventional method for calculating the L_2 metric, we required 248.66 ± 1.88 seconds on average.

For a stricter comparison, we used the k-d tree [10] to obtain recurrence plots for the same data (see Figs. 1-3). When we used the k-d tree, we use the first 400 time points to estimate 1% point of distances. Then, we used the estimated 1% point of distances, to obtain pairs of times where the distance is less than the estimated 1% point. We found that the computational time for the k-d tree were slower for shorter time series, became comparable with the InDDeCs when the length was more than or equal to 5000 (Fig. 3). In addition, the accuracies evaluated were also similar. Taking into the fact that the InDDeCs obtain the whole distance matrix, while the k-d tree extracts only pairs of neighbours, the results of the InDDeCs are more informative and useful than those of the k-d tree if we use the similar computational resources. In addition, we note that if we also want to obtain the distances between neighbours so that we can plot 1% of places exactly, the method using the k-d tree needs more time.

D. Correlation dimensions and maximal Lyapunov exponents

When we evaluated the correlation dimensions [14] (Appendix B.2) as well as the maximal Lyapunov exponents [15,16] (Appendix B.3), the values obtained seemed to depend on which λ we used (Fig. 4). However, the means of the estimated values coincided well with the values provided in the literature [14,18,19] (see Fig. 4 and Tables I and II). To imply deterministic chaos or strangeness, we may combine the correlation dimension and the maximum Lyapunov exponent with surrogate data [20,21]. As demonstrated in Figs. 5 and 6, the correlation dimesion and the maximal Lyapunov exponent for the autoregressive linear model ($x(t+1) = -0.8x(t) + \xi(t)$, where $\xi(t)$ follows the Gaussian distribution with mean 0 and standard deviation 1) were 95% confidence intervals of the iteratively adjusted Fourier transform surrogates [21], while those values for the Hénon map [22] were not. Thus, it may be worth expanding the use of InDDeCs for these applications.

E. Applications to forced systems

InDDeCs have their strongest potential when they are applied along with Stark's embedding theorem [11] for forced systems. Stark's embedding theorem can be used for detecting directional couplings [23,24] and extracting slow driving forces [25,26]. Although there are some rules of thumb for choosing the embedding parameters for Takens' embedding theorem [1], such as the first minimum of mutual information [3] and the false nearest neighbour method [4], there are no known practical rules of thumb for Stark's theorem [11]. However, if we use InDDeCs, we do not have to worry much if the embedding dimension is higher than twice the sum of the dimensions for a driving force and its forced system.

First, we show some examples for detecting directional couplings (see Fig. 7). We implemented the method using the joint distribution of distances [24] with InDDeCs (see Appendix B.4 for the detail). Namely, we leverage the characteristic whereby the distances for the reconstructed space for A become small when those for B are small, if A drives B [23,24]. When we try to identify directional couplings using toy models composed by coupled logistic maps (the first to the third examples of Ref. [24]), the method detected directional couplings appropriately, whose results did not depend much on the parameter λ. Namely, when two logistic maps were mutually coupled (Appendix B.5), directional couplings tended to be identified correctly (Figs. 7a and 7b); when two logistic maps were driven by another logistic map (Appendix B.6), the directional couplings identified did not depend much on the coupling parameters that decided the driving strengths (Figs. 7c and 7d); and even when there was a driving force affecting two logistic maps which were mutually coupled (Appendix B.7), the directional couplings between the mutually coupled logistic maps were identified correctly (Figs. 7e and 7f). Thus, the method using the joint distribution of distances seems to work well with InDDeCs.

Second, we present some examples for extracting slow-driving forces using the methods of Refs. [25,26] (see Appendix B.8 for the detail). We used the Hénon map [22] driven by the Lorenz model [27] and the Rössler model [17] (see Appendix B.9). We found that the driving forces modelled by the Lorenz model and the Rössler model were identified correctly when the parameter λ was in the appropriate range, i.e., between 0.17 and 0.88 (see Fig. 8). Therefore, by using InDDeCs, slow driving forces appear to have been identified correctly.

An important problem we will encounter when we analyse a real dataset is how many driving forces we should choose. When we predicted 25 steps ahead by taking into account the reconstructed slow driving forces with the radial basis function model [28], we found that the prediction errors decreased up to the second driving force (Fig. 9). But when we included the third driving force, the prediction errors did not decrease to a great extent (Fig. 9). Thus, we can validate the reconstructed driving forces with time series prediction.

Lastly, we show a real world example for detecting directional couplings and slow-driving forces in the weather at Akita, Japan (see Appendix B.10). The slow-driving forces reconstructed from the temperature, the solar irradiance, the precipitation, and the wind speed and validated by 12 hours ahead time series prediction (Fig. 10) were shown in Figs. 11a-11d. Here, we used the method of Ref. [29] to ensure the continuity for the reconstructed driving forces. Thus, we saw abnormal behaviour, especially for the driving force reconstructed from the precipitation at the beginning of the time series. Thus, we compared the reconstructed driving forces between January 2011 and May 2015 and found that these reconstructed driving forces showed strong correlations between most pairs of possible combinations (Table III). Thus, their drivers seem to reflect similar behaviour. When we identified directional couplings by means of the original method using the joint distribution of distances with the conventional delay coordinates, directional couplings failed to be identified, especially if a pair of combinations contained the precipitation or solar irradiance (see Fig. 11e), both of which were intermittent and contained many continuous zeros (Fig. 12). For example, observe that the directional coupling from the wind speed to precipitation was not identified. But, if we applied the method using the joint distribution of distances with InDDeCs, we did not have many problems related to these intermittent non-zero observations and we succeeded in identifying directional couplings, even for the pairs related to the precipitation and solar irradiance (see Fig. 11f, especially the coupling direction from the wind speed to precipitation), implying that each weather variable demonstrates aspects of the same underlying dynamics. This detectability could be the strongest point among the applications of InDDeCs.

We classified the topologies of temporal networks into two types using the k-mean algorithm [30] (Fig. 13). In the cluster 0, the solar irradiance drives the temperature, and the temperature and the wind speed drive the precipitation. In the cluster 1, the wind speed drives the precipitation. Note that the topology of the cluster 1 is a sub-network for the topology of the cluster 0. We also found that the cluster 0 is likely to appear in the summer while the cluster 1 is likely to appear in the winter (Fig. 14). To validate the inferred network topology, we attempted to predict the precipitation 1 hour ahead by taking into account the temperature and the wind speed. We found that by considering the temperature as well as the wind speed, we could improve the time series prediction for the precipitation (Fig. 15). This result is consistent with the found topology.

There is another support for the network topology of cluster 0: The reconstructed and validated driving force for the precipitation has large correlation coefficients with those for the temperature and the wind speed (Table III). This finding also agrees with the network structure shown for the cluster 0 in Fig. 13.

III. DISCUSSION

It is common to estimate the spectrum of the Lyapunov exponents [18] for characterizing the high-dimensional dynamics. However, because such a method produces the exponents whose number is equal to the dimension, one might encounter spurious exponents [31] if we do not know the exact number for the dimension of the underlying dynamics and use higher embedding dimension. That would be our reason to recommend the estimation of the maximal Lyapunov exponent based on a distance matrix if we are not sure how large the dimension for the underlying dynamics is. By using the maximal Lyapunov exponent, one can tell at least whether the underlying dynamics is chaotic or not.

If a given time series has a high sampling rate, then we may use the time delay as Ref. [3] to control the sampling rate so that we use a fewer time points for the analysis.

The decaying factor λ works similarly to the embedding dimension. However, there is also a qualitative difference between them: when we use the conventional delay coordinates, we could not retain the information for the time before (the embedding dimension)×(the time delay), while with InDDeCs, we could retain such information, which is subject to the observational noise. This difference was the most prominent in the example on identifying directional couplings, especially the coupling direction from the wind speed to the precipitation, which is naturally prominent around low atmospheric pressures.

Theorem 2 means that achieving the one-to-one correspondence between the conventional delay coordinates and InDDeCs might be too demanding if the underlying dynamics is in a very high-dimensional space, wherein if the minimum for the local Lyapunov exponent would be negative and its absolute value is large, the appropriate decaying parameter could be very close to zero. In such a case, we might follow the approach of Berry et al. [9], reducing the dimension and giving up the estimation of the invariant measures because the invariant measures had been somehow optimized for characterizing low-dimensional dynamics historically. This approach in Berry et al. [9] sounds reasonable, judging from the fact that we could identify the directional couplings and the slow drivers correctly in the wide ranges of the decaying parameters (see Figs. 7 and 8). However, to justify this approach using InDDeCs, another set of mathematical support must be prepared, which would be an open question.

Here we assume that InDDeCs are used for some application of nonlinear time series analysis where we need a distance matrix, namely, each distance for every pair of time points. Thus, implicitly we assume that we already have the whole dataset. If we would like to apply InDDeCs to online streaming data, we need to use the different approach that had been discussed in Ref. [5].

In this paper, we have shown theoretically that the infinite-dimensional delay coordinates (InDDeCs) have good one-to-one and metric equivalence characteristics when the parameter λ of the decaying factor is chosen appropriately. We have also demonstrated numerically that we can more quickly obtain a recurrence plot that looks similar to the one obtained from the conventional delay coordinates. InDDeCs can be also used for estimating the correlation dimensions and the maximal Lyapunov exponents, as well as identifying directional couplings and slow driving forces. Thus, in short, InDDeCs establish an “embedding” without explicitly considering the dimensions of target systems. We hope that this new tool helps elucidate the underlying mechanisms for many interesting dynamical systems.

Appendix A Proofs

A.1 Proof for one-to-one correspondence.

Let $f: M \to M$ represent a dynamical system defined as a diffeomorphism on an *m*-dimensional manifold *M*, and $g: M \to R$ an observation function. Then, a set of delay coordinates can be described by $G_d(x) = \left(g(x), g\left(f^{-1}(x)\right), g\left(f^{-2}(x)\right), \ldots, g\left(f^{-d+1}(x)\right)\right)$, while a set of infinite-dimensional delay coordinates is described by $H_\lambda(x) = \left(g(x), \lambda g\left(f^{-1}(x)\right), \lambda^2 g\left(f^{-2}(x)\right), \ldots\right)$, where we set $\lambda \in (0,1)$. Suppose that the embedding theorem by Takens [1] is satisfied and there is one-to-one correspondence between *x* and $G_d(x)$ on the manifold (similar proofs can be established for those theorems by Sauer et al. [2] and Stark [11]). To connect $G_d(x)$ and $H_\lambda(x)$, we consider some intermediate steps $H_{\lambda,c}(x) = \left(g(x), \lambda g\left(f^{-1}(x)\right), \lambda^2 g\left(f^{-2}(x)\right), \ldots, \lambda^{c-1} g\left(f^{-c+1}(x)\right)\right)$, where $c \geq d$.

Let us start by establishing one-to-one correspondence between $G_d(x)$ and $H_{\lambda,d}(x)$. Suppose $G_d(x) = G_d(y)$. Then, we have

$$\begin{aligned}&\left(g(x), g\left(f^{-1}(x)\right), g\left(f^{-2}(x)\right), \ldots, g\left(f^{-d+1}(x)\right)\right)\\ &= \left(g(y), g\left(f^{-1}(y)\right), g\left(f^{-2}(y)\right), \ldots, g\left(f^{-d+1}(y)\right)\right).\end{aligned} \tag{A1}$$

By multiplying the *n*th component by λ^{n-1}, we have

$$
\begin{aligned}
&\left(g(x), \lambda g\left(f^{-1}(x)\right), \lambda^2 g\left(f^{-2}(x)\right), \ldots, \lambda^{d-1} g\left(f^{-d+1}(x)\right)\right) \\
&= \left(g(y), \lambda g\left(f^{-1}(y)\right), \lambda^2 g\left(f^{-2}(y)\right), \ldots, \lambda^{d-1} g\left(f^{-d+1}(y)\right)\right),
\end{aligned}
\tag{A2}
$$

which can be rewritten by $H_{\lambda,d}(x) = H_{\lambda,d}(y)$.

We also prove the converse. Suppose that $H_{\lambda,d}(x) = H_{\lambda,d}(y)$. Then, this equality means element-wise $\lambda^{n-1} g\left(f^{-n+1}(x)\right) = \lambda^{n-1} g\left(f^{-n+1}(y)\right)$. Because $\lambda \in (0,1)$, especially $\lambda \neq 0$, we have $g\left(f^{-n+1}(x)\right) = g\left(f^{-n+1}(y)\right)$ for each *n* between 1 and *d*. Thus, we have

$$
\begin{aligned}
&G_d(x) = \left(g(x), g\left(f^{-1}(x)\right), g\left(f^{-2}(x)\right), \ldots, g\left(f^{-d+1}(x)\right)\right) \\
&= \left(g(y), g\left(f^{-1}(y)\right), g\left(f^{-2}(y)\right), \ldots, g\left(f^{-d+1}(y)\right)\right) = G_d(y).
\end{aligned}
\tag{A3}
$$

Thus, the converse is also true. Thus, $H_{\lambda,d}(x) = H_{\lambda,d}(y)$ if and only if $G_d(x) = G_d(y)$.

Next, we prove that $H_{\lambda,m+1}(x) = H_{\lambda,m+1}(y)$ if and only if $H_{\lambda,m}(x) = H_{\lambda,m}(y)$ for $m \geq d$. First, we prove the case of $m = d$. Due to the above proof, $H_{\lambda,d}(x) = H_{\lambda,d}(y)$ if and only if $G_d(x) = G_d(y)$. In addition, due to the initial assumption, we have $G_d(x) = G_d(y)$ if and only if $x = y$. Because f is a diffeomorphism, we have a unique value of $g\left(f^{-d}(x)\right) = g\left(f^{-d}(x)\right)$. By multiplying by λ^d, we have $\lambda^d g\left(f^{-d}(x)\right) = \lambda^d g\left(f^{-d}(x)\right)$. This equality means that we have $H_{\lambda,d+1}(x) = H_{\lambda,d+1}(y)$. The converse is almost trivial if we start with

$$H_{\lambda,d+1}(x) = \left(g(x), \lambda g\left(f^{-1}(x)\right), \lambda^2 g\left(f^{-1}(x)\right), ..., \lambda^d g\left(f^{-d}(x)\right)\right) = \left(g(y), \lambda g\left(f^{-1}(y)\right), \lambda^2 g\left(f^{-1}(y)\right), ..., \lambda^d g\left(f^{-d}(y)\right)\right) = H_{\lambda,d+1}(y) \quad \text{(A4)}$$

and drop the last element to have

$$H_{\lambda,d}(x) = \left(g(x), \lambda g\left(f^{-1}(x)\right), \lambda^2 g\left(f^{-1}(x)\right), ..., \lambda^{d-1} g\left(f^{-d+1}(x)\right)\right) = \left(g(y), \lambda g\left(f^{-1}(y)\right), \lambda^2 g\left(f^{-1}(y)\right), ..., \lambda^{d-1} g\left(f^{-d+1}(y)\right)\right) = H_{\lambda,d}(y). \quad \text{(A5)}$$

Therefore, we proved that $H_{\lambda,d+1}(x) = H_{\lambda,d+1}(y)$ if and only if $H_{\lambda,d}(x) = H_{\lambda,d}(y)$.

Similarly, we can prove $H_{\lambda,m+1}(x) = H_{\lambda,m+1}(y)$ if and only if $H_{\lambda,m}(x) = H_{\lambda,m}(y)$ for $m \geq d$.

Using the first part of the proof once and the second part of proof inductively, we reach our proposition that $H_{\lambda}(x) = H_{\lambda}(y)$ if and only if $G_d(x) = G_d(y)$, and thus if and only if $x = y$. Thus, we have proved the one-to-one property for the infinitely dimensional delay coordinates.

A.2 Proof for equivalence between the conventional delay coordinates and the infinite-dimensional delay coordinates

Let $\overline{L}$ and $\underline{L}$ be the supremum and the infimum, respectively, for the local maximum and minimum Lyapunov exponents, which are independent of the positions on the attractor. In addition, we define

$$\delta_D(t_1,t_2) = \left\| G_D(x(t_1)) - G_D(x(t_2)) \right\|_{L_1}, \tag{A6}$$

and

$$\Delta_D(t_1,t_2) = \left\| H_{\lambda,D}(x(t_1)) - H_{\lambda,D}(x(t_2)) \right\|_{L_1}. \tag{A7}$$

Then, it is reasonable to assume that

$$\delta_D(t_1 - nD, t_2 - nD) e^{\underline{L} nD} \le \delta_D(t_1,t_2) \le \delta_D(t_1 - nD, t_2 - nD) e^{\bar{L} nD}. \tag{A8}$$

In addition, we have

$$\lambda^D \delta_D(t_1,t_2) \le \Delta_D(t_1,t_2) \le \delta_D(t_1,t_2), \tag{A9}$$

and

$$\left\| H_\lambda(x(t_1)) - H_\lambda(x(t_2)) \right\|_{L_1} = \sum_{n=0}^{\infty} \lambda^{nD} \Delta_D(t_1 - nD, t_2 - nD). \tag{A10}$$

Thus, $\left\| H_\lambda(x(t_1)) - H_\lambda(x(t_2)) \right\|_{L_1}$ can be upper-bounded as

$$\begin{aligned} &\left\| H_\lambda(x(t_1)) - H_\lambda(x(t_2)) \right\|_{L_1} \le \sum_{n=0}^{\infty} \lambda^{nD} \delta_D(t_1 - nD, t_2 - nD) \\ &\le \sum_{n=0}^{\infty} \left(\lambda e^{-\underline{L}}\right)^{nD} \delta_D(t_1,t_2) = \frac{\delta_D(t_1,t_2)}{1 - \left(\lambda e^{-\underline{L}}\right)^D}, \end{aligned} \tag{A11}$$

if $\lambda e^{-\underline{L}} < 1$. Moreover, $\left\| H_\lambda(x(t_1)) - H_\lambda(x(t_2)) \right\|_{L_1}$ can be lower-bounded as

$$\begin{aligned} &\left\| H_\lambda(x(t_1)) - H_\lambda(x(t_2)) \right\|_{L_1} \ge \lambda^D \sum_{n=0}^{\infty} \lambda^{nD} \delta_D(t_1 - nD, t_2 - nD) \\ &\ge \lambda^D \sum_{n=0}^{\infty} \left(\lambda e^{-\bar{L}}\right)^{nD} \delta_D(t_1,t_2) = \frac{\lambda^D \delta_D(t_1,t_2)}{1 - \left(\lambda e^{-\bar{L}}\right)^D}, \end{aligned} \tag{A12}$$

if $\lambda e^{-\bar{L}} < 1$.

Therefore, when $\lambda < \min\{e^{\underline{L}}, e^{\bar{L}}\} = e^{\underline{L}}$, we have

$$\frac{\lambda^D \delta_D(t_1 t_2)}{1-\left(\lambda e^{-\bar{L}}\right)^D} \le \left\| H_\lambda(x(t_1)) - H_\lambda(x(t_2)) \right\|_{L_1} \le \frac{\delta_D(t_1, t_2)}{1-\left(\lambda e^{-\underline{L}}\right)^D} \qquad \text{(A13)}$$

and the two metrics $\delta_D(t_1, t_2)$ and $\left\| H_\lambda(x(t_1)) - H_\lambda(x(t_2)) \right\|_{L_1}$ are equivalent.

When a metric is sandwiched with another metric in this manner, the correlation dimensions estimated for both metrics agree with each other [32]. That is one of the reasons why we call these two metrics equivalent.

Note: The condition of $\lambda < e^{\underline{L}}$ is, intuitively, complementary for the condition described in Berry et al. [9] for dimension reduction. Thus, our condition is reasonable from this viewpoint as well.

Appendix B Details for Numerical Calculations

B.1 Calculation for distance matrices using InDDeCs

Suppose that we calculate the distance matrix $S \in R^{I \times I}$ for a scalar time series $\{s_i \in R \mid i = 1,2,\ldots,I\}$ with length I using InDDeCs. In addition, let $\bar{\underline{s}}$ be the difference between the minimum and the maximum of $\{s_i\}$. Then, we apply the following algorithm to calculate S:

For *i* from 1 to (*n*-1)

  Calculate the (*i*,1) element as follows:

$$S(i,1) := \frac{\lambda \bar{\underline{s}}}{1-\lambda} + \left| s_i - s_1 \right| . \tag{B1}$$

  Copy it to the (1,*i*) element:

$$S(1,i) := S(i,1) . \tag{B2}$$

  For *j* from 2 to (*i*-1)

    Calculate the (*i*,*j*) element as follows:

$$S(i,j) := \lambda S(i-1, j-1) + \left| s_i - s_j \right| . \tag{B3}$$

    Copy it to the (*j*,*i*) element:

$$S(j,i) := S(i,j) . \tag{B4}$$

  end

end

This algorithm means that if we go back to a time point before the beginning of the given time series, we insert the dummy value $\bar{\underline{s}}$ for the past distances. By this algorithm, we will overcome the differences of dimensions we can access.

There is another remark here: This implementation for InDDeCs is much simpler than the implementation for the conventional delay coordinates using the decomposition of Eq. (4) of Ref. [5], where we must subtract the past pairs of distances appropriately to obtain the current distances.

B.2 Estimation of correlation dimension

The time series of length 10000 was used for the calculations. After obtaining the distance matrices using InDDeCs, we threw away the components corresponding to the first 1000 points and used the remaining parts for the estimation because we need to supply dummy distances for the times before the beginning of the time series and the distances for the first 1000 points may not be calculated precisely because $0.5^{100} \sim 10^{-30}$ will approach the margin of machine errors. We found the minimum non-zero distance m and set the range of $[10m, 1000m]$ as the scaling region. The other parts were the same as Ref. [14].

B.3 Estimation of maximal Lyapunov exponents

We used the time series containing 10000 time points for estimating the maximal Lyapunov exponents. After obtaining the distance matrices using InDDeCs, we threw away the components corresponding to the first 1000 points and used the remaining parts for the estimation. We chose the five nearest neighbours avoiding points in the same strands, i.e., neighbours within 200 time points, for estimating each of the maximal Lyapunov exponents. For the flows, we found the slope by fitting a line between 50 and 100 time points forward in time. For the maps, we found the slope between the first and the second steps. The rest of the calculation was similar to Ref. [15].

B.4 Identifying directional couplings

We used the method of joint distribution for distances [24] here. After obtaining the distance matrix as described above, we subsampled the distance matrix every 10 points to reduce the temporal correlations as well as the calculation costs. Then, we applied the method of joint distribution of distances [24] directly.

B.5 Mutually coupled logistic maps

Two logistic maps [33] were coupled in the following way mutually [24]:

$$x(t+1) = (1-\eta_{yx})(3.8x(t)(1-x(t))) + \eta_{yx}(3.81y(t)(1-y(t))), \quad \text{(B5)}$$

$$y(t+1) = (1-\eta_{xy})(3.81y(t)(1-y(t))) + \eta_{xy}(3.8x(t)(1-x(t))). \quad \text{(B6)}$$

After we removed the initial transient, we generated time series of length 2000 for each set of parameters (η_{yx}, η_{xy}). Parameters η_{yx} and η_{xy} were varied between 0 and 0.2.

B.6 Two logistic maps driven by another logistic map

We considered the following coupled logistic maps:

$$x(t+1) = (1-\eta_{zx})(3.8x(t)(1-x(t))) + \eta_{zx}(3.82z(t)(1-z(t))), \quad \text{(B7)}$$

$$y(t+1) = (1-\eta_{zy})(3.81y(t)(1-y(t))) + \eta_{zy}(3.82z(t)(1-z(t))), \quad \text{(B8)}$$

$$z(t+1) = 3.82z(t)(1-z(t)). \quad \text{(B9)}$$

Namely, in this coupled system, z drives x and y, but x and y are not mutually connected. We varied η_{zx} and η_{zy} between 0 and 0.2. The rest is similar to the case of mutually coupled logistic maps.

B.7 Mutually coupled logistic map driven by another

Here we considered the following coupled systems:

$$x(t+1) = (1-\eta_{yx}-\eta_{zx})(3.8x(t)(1-x(t))) + \eta_{yx}(3.81y(t)(1-y(t))) + \eta_{zx}(3.82z(t)(1-z(t))) \text{ ,} \tag{B10}$$

$$y(t+1) = (1-\eta_{xy}-\eta_{zy})(3.81y(t)(1-y(t))) + \eta_{xy}(3.8x(t)(1-x(t))) + \eta_{zy}(3.82z(t)(1-z(t))) \text{ ,} \tag{B11}$$

$$z(t+1) = 3.82z(t)(1-z(t)) \text{ .} \tag{B12}$$

In this example, we set $\eta_{zx} = \eta_{zy} = 0.05$ and varied η_{yx} and η_{xy} between 0 and 0.2. The rest is similar to the case of the mutually coupled logistic maps.

B.8 Extracting slow-driving forces

First, we obtain a recurrence plot of observables using InDDeCs. Second, we make the granularity of the recurrence plot coarse by using box sizes of 50 and 24 respectively for the toy model and the weather example, to obtain the meta-recurrence plot [25]. This meta-recurrence plot corresponds to a recurrence plot of slow-driving forces. Third, we apply the method of Ref. [26] to reproduce the time series of driving forces. Note that the method of Ref. [26] for reconstructing an original time series from a recurrence plot has mathematical support [32,34].

B.9 The Hénon map driven by the Lorenz model and the Rössler model

We fed the Lorenz model (Eqs. (B13)-(B16)) and the Rössler model (Eqs. (B17)-(B20)) to the

Hénon map (Eqs. (B21)-(B22)) in the following way:

$$\dot{x} = -10(x - y)\,, \tag{B13}$$

$$\dot{y} = -xz + 28x - y\,, \tag{B14}$$

$$\dot{z} = xy - \frac{8}{3}z\,, \tag{B15}$$

$$\tilde{x}(t) = \frac{x(t) - \bar{x}}{\sigma_x}\,, \tag{B16}$$

$$\dot{u} = -(v + w)\,, \tag{B17}$$

$$\dot{v} = u + 0.36v\,, \tag{B18}$$

$$\dot{w} = 0.4 + w(u - 4.5)\,, \tag{B19}$$

$$\tilde{u}(t) = \frac{u(t) - \bar{u}}{\sigma_u}\,, \tag{B20}$$

$$p(t+1) = 1 - 1.2\big(1 + 0.05\tilde{x}(0.0005t)\big)p(t)^2 + 0.3\big(1 + 0.1\tilde{u}(0.002t)\big)q(t)\,, \tag{B21}$$

$$q(t+1) = p(t)\,. \tag{B22}$$

After removing the initial transient, we generated a series of $p(t)$ with 20000 time points.

B.10 Weather data at Akita and their analysis

We used the temperature, the precipitation, the solar irradiance, and the wind speed measured at Akita, Japan. The measurements were given every 10 minutes between 1 January 2010 and 31 May 2016. If a measurement was missing, we inserted its most recent valid value instead and preprocessed the dataset. For extracting slow-driving forces, we subsampled the measurements every hour. For detecting directional couplings, we divided the dataset into 77 segments corresponding to each month and analysed the segments.

B. 11 Validating reconstructed slow driving forces and network structure by time series prediction

We used the radial basis function model in Ref. [28] to test whether additional time series improved time series prediction. When we used the reconstructed slow driving forces, we interpolate the time series so that the sampling intervals become the same as the original time series. Then we normalized the additional time series so that their standard deviations become 10% of the time series we predicted. By increasing the number of additional time series taken into account, we validated whether or not the reconstructed slow driving forces and/or the found topologies are appropriate. For each case, we generated 10 time series predictions by choosing different centres for the radial basis functions. In Figs. 9 and 10, we used two-dimensional delay coordinates with additional dimensions for the additional time series. In Fig. 15, we used the usual delay coordinates spanning the time window of 2 hours for the precipitation as well as the temperature and wind speed when they were considered.

ACKNOWLEDGEMENTS

We thank Prof. Motomasa Komuro, Dr. Shunsuke Horai, and Dr. Miwa Fukino for the helpful discussions. The dataset of the weather at Akita was provided by the Japan Meteorological Agency. The dataset is commercially available from the Japan Meteorological Business Support Center. This manuscript is partially based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO). This research is also partially supported by Core Research for Evolutional Science and Technology (CREST) Grant Number JPMJCR14D2, Japan Science and Technology Agency (JST), JSPS KAKENHI Grant Number 15H05707, and Kozo Keikaku Engineering Inc., Japan.

REFERENCES

1. F. Takens, *Lect. Notes Math.* **898**, 366 (1981).
2. T. Sauer, J. A. Yorke, M. Casdagli, *J. Stat. Phys.* **65**, 579 (1991).
3. A. Fraser, H. Swinney, *Phys. Rev. A* **33**, 1134 (1986).
4. M. B. Kennel, R. Brown, H. D. I. Abarbanel, *Phys. Rev. A* **45**, 3403 (1992).
5. Y. Hirata, T. Takeuchi, S. Horai, H. Suzuki, K. Aihara, *Sci. Rep.* **5**, 15736 (2015).
6. J. D. Farmer, J. D., J. J. Sidorowich, Predicting chaotic dynamics, In *Dynamic Patterns in Complex Systems* eds. Kelso, J. A. S., Mandell, A. J. & Shlesinger, M. F., World Scientific, Singapore, pp.265-292.
7. D. B. Murray, *Physica D* **68**, 318 (1993).
8. J. McNames, A nearest trajectory strategy for time series prediction, In *Proc. of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling,* Katholieke Universiteit Leuven, Belgium, July 1998, pp.112-128.
9. T. Berry, J. R. Cressman, Z. Gregurić-Ferenček, T. Sauer, *SIAM J. Appl. Dyn. Syst.* **12**, 618 (2013).
10. J. H. Friedman, J. L. Bentley, R. A. Finkel, *ACM Trans. Math Soft.* **3**, 209 (1977).
11. J. Stark, *J. Nonlinear Sci.* **9**, 255 (1999).
12. J.-P. Eckmann, S. O. Kamphorst, R. Ruelle, *Europhys. Lett.* **4**, 973 (1987).

13. N. Marwan, M. C. Romano, M. Thiel, J. Kurths, *Phys. Rep.* **438**, 237 (2007).

14. P. Grassberger, I. Procaccia, *Phys. Rev. Lett.***50**, 346 (1983).

15. M. T. Rosenstein, J. J. Collins, C. J. De Luca, *Physica D* **65**, 117 (1993).

16. H. Kantz, *Phys. Lett. A* **185**, 77 (1994).

17. O. E. Rössler, *Phys. Lett.* **57A**, 397 (1976).

18. M. Sano, Y. Sawada, *Phys. Rev. Lett.* **55**, 1082 (1985).

19. Y. Hirata, K. Judd, D. Kilminster, *Phys. Rev. E* **70**, 016215 (2004).

20. J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. D. Farmer, *Physica D* **58**, 77 (1992).

21. T. Schreiber and A. Schmitz, *Phys. Rev. Lett.* **77**, 635 (1996).

22. M. Hénon, *Commun. Math. Phys.* **50**, 69 (1976).

23. Y. Hirata, K. Aihara, *Phys. Rev. E* **81**, 016203 (2010).

24. Y. Hirata, *et al., PLoS ONE* **11**, e0158572 (2016).

25. M. C. Casdagli, *Physica D* **108**, 12 (1997).

26. Y. Hirata, S. Horai, K. Aihara, *Eur. Phys. J. Spec. Top.* **164**, 13 (2008).

27. E. N. Lorenz, *J. Atmos. Sci.* **20**, 130 (1963).

28. K. Judd and A. I. Mees, *Physica D* **82**, 426 (1995).

29. N. Gershenfeld, *The Nature of Mathematical Modeling*, Cambridge University Press, Cambridge UK, 1999.

30. M. Tanio, Y. Hirata, H. Suzuki, *Phys. Lett. A* **373**, 2031 (2009).

31. T. Sauer, J. A. Tempkin, J. A. Yorke, *Phys. Rev. Lett.* **81**, 4341 (1998).

32. Y. Hirata, M. Komuro, S. Horai, K. Aihara, *Int. J. Bifurcat. Chaos* **25**, 1550168 (2015).

33. R. May, *Nature* **261**, 459 (1976).

34. A. Khor, M. Small, *Chaos* **26**, 043101 (2016).

35. R. Hegger, H. Kantz, and T. Schreiber, *Chaos* **9**, 413 (1999).

Figure captions

FIG. 1. Recurrence plots of the Rössler model obtained by the infinite-dimensional delay coordinates (InDDeCs) and the conventional three-dimensional delay coordinates. Panels a, d, and g show the recurrence plots obtained by InDDeCs with $\lambda = 0.1, 0.5,$ and 0.9, respectively. Panels b, e, and h show the recurrence plots obtained by the conventional delay coordinates using all the distances with the embedding dimensions of 1, 3, and 10, respectively. Panels c, f, and i show the recurrence plots obtained by the k-d tree [10] with the embedding dimenisons of 1, 3, 10, respectively. In panels a, b, d, e, g, and h, exactly 1% of places have the plotted points, while in panels c, f, and i, nearly 1% point of distances was used to plot the recurrence plots.

FIG. 2. The average accuracy of the obtained recurrence plots among 10 trials against to those for the conventional delay coordinates with the time points between 1001 and 2000. In panel a, we compared recurrence plots (the red solid line) obtained by the InDDeCs using $\lambda = 0.1$ with recurrence plots obtained by the decomposition of Eq. (4) of Ref. [5] using the L_1 metric (the blue dashed line) and recurrence plots obtained by the k-d tree [10] (the green dash dotted line) using the embedding dimension of 1. In panels b and c, the similar figures are shown with $\lambda = 0.5$ and 0.9 and the embedding dimensions of 3 and 10, respectively.

FIG. 3. The average times required among 10 trials for the InDDeCs (the red solid line) and the conventional delay coordinates with a normal implementation using the L_2 metric (the black dotted line), as well as the decomposition of Eq. (4) of Ref. [5] using the L_1 metric (the blue dashed line) to obtain the corresponding distance matrices given a length of time series. In addition, the calculation times required by the k-d tree were shown in the green dash-dotted line. In panel a, we used $\lambda = 0.1$ and the embedding dimension of 1. In panel b, we used $\lambda = 0.5$ and the embedding dimension of 3. In panel c, we used $\lambda = 0.9$ and the embedding dimension of 10. For these calculations, we used a computer with 2.7 GHz 12-Core Intel Xeon E5 with 64 GB memory.

FIG. 4. Correlation dimensions and maximal Lyapunov exponents estimated using InDDeCs. Panels a and b are for correlation dimensions and Panels c-f are for maximal Lyapunov exponents. Panels a and c are for the Hénon map. Panels b and d are for the Lorenz model with sets of parameters ($R = 28, \sigma = 10, b = 8/3$) and ($R = 40, \sigma = 16, b = 4$), respectively. Panel e is for the Ikeda map and Panel f is for the Rössler model. For each panel, the estimation using InDDeCs is shown in the blue solid line and the estimation in the literature is shown in the red dashed line.

FIG. 5. Surrogate tests using correlation dimensions. We used the random shuffle surrogates [20] (the black dotted lines), phase randomized surrogates [20] (the blue dash-dotted lines), and iterative amplitude adjusted Fourier transform (IAAFT) surrogates [21] (the green dashed lines), respectively. Each of the two lines show 95% confidence intervals obtained by 40 surrogates each. The red solid thick line shows the lines obtained from the original data. In panel a, we show the results for the autoregressive linear model, and in panel b, we show the results for the Hénon map. The length of time series generated was 2000 each. We applied the end-to-end matching [35] as a preprocessing to avoid spurious high frequency components during the Fourier transforms.

FIG. 6. Surrogate tests using the maximum Lyapunov exponents. See the caption of Fig. 5 to interpret the results.

FIG. 7. Identifying directional couplings depending on parameter λ of InDDeCs and coupling strengths η_{yx} and η_{xy} in the examples of coupled logistic maps. In the first and second rows, we used mutually coupled logistic maps x and y. In the third and fourth rows, we used logistic maps x and y driven by another logistic map z. In the fifth and sixth rows, we

used mutually coupled logistic maps x and y driven by another logistic map z. In each row, the left, centre, and right columns correspond to the results of $\lambda = 0.1, 0.5$ and 0.9, respectively. In each panel, the logarithms for the p-values were shown. In the white regions, the p-values were smaller than 0.01, representing significance. The darker regions show greater p-values, which are not significant.

FIG. 8. Driving forces and their reconstructions. (a) Correlation coefficients between driving forces and their reconstructions using InDDeCs depending on the parameter λ. The blue solid line corresponds to the driving force constructed by the Lorenz model, while the red dash-dotted line corresponds to the driving force constructed by the Rössler model. (b) The original driving force of the Lorenz model (the black solid line) and its reconstruction (the blue dashed line) when $\lambda = 0.62$. (c) The original driving force of the Rössler model (the black solid line) and its reconstruction (the red dashed line) when $\lambda = 0.62$. In Panels (b) and (c), we plotted the reconstructed driving forces so that the means, standard deviations, and direction are matched with the original driving forces.

FIG. 9. The validation for the reconstructed driving forces using time series prediction in the example of Fig. 8. The prediction errors are shown by box plots. When we take into account the

reconstructed slow driving forces up to the second one, the prediction errors have improved significantly. We used $\lambda = 0.62$ as an example.

FIG. 10. The validations for the reconstructed driving forces for the weather data at Akita, Japan. Panels a, b, c, and d correspond to the temperature, the solar irradiance, the precipitation, and the wind speed. For the temperature, the solar irradiance, and the precipitation, the first driving force significantly reduced the prediction errors. For the wind speed, we found the third driving force reduced the prediction errors. Thus, later we selected these driving forces to conduct the further analysis.

FIG. 11. Reconstructed driving forces and directional couplings for the weather at Akita, Japan. Panels a, b, c, and d are the validated principal components for the driving forces reconstructed by using InDDeCs for the temperature, solar irradiance, precipitation, and wind speed. Panels e and f represent the directional couplings identified using the method of joint distribution for distances using (e) the conventional delay coordinates and (f) InDDeCs. The grey scales show the p-values in the logarithm using base 10. Namely, the white regions correspond to significant pairs of the time and coupling direction with the significance level of 0.01. The darker regions have higher p-values, which are not significant.

FIG. 12. Parts of time series for the weather at Akita, Japan. Panel a corresponds to the temperature, Panel b corresponds to the precipitation, Panel c corresponds to the solar irradiance, and Panel d corresponds to the wind speed. Because the time series data shown correspond to the beginning of January, which means the winter season, we could not see the clear daily cycle for the temperature in panel a for the first four days, possibly due to the accumulated snow.

FIG. 13. Estimated network structures. By classifying the estimated network structures using the k-mean algorithm [30], we identified two typical structures, which are denoted by cluster 0 (panel a) and cluster 1 (panel b).

FIG. 14. The frequency of estimated network structures depending on the month within a year.

FIG. 15. Validation for the estimated network structures. Based on the results for the cluster 0 of Fig. 13, the precipitation is driven by the temperature and/or the wind speed. Thus, we tested whether the measurements for the temperature and the wind speed help to improve the prediction for the precipitation. We found that by taking into account the temperature and the wind speed, the 1 hour ahead time series prediction for the precipitation has been improved

significantly. Thus, the finding in Fig. 13 seems appropriate.

TABLE I. Estimated values using InDDeCs and values known in the literature for correlation dimensions. Each values shown with $\pm$ represent the mean and standard deviation for the estimated values.

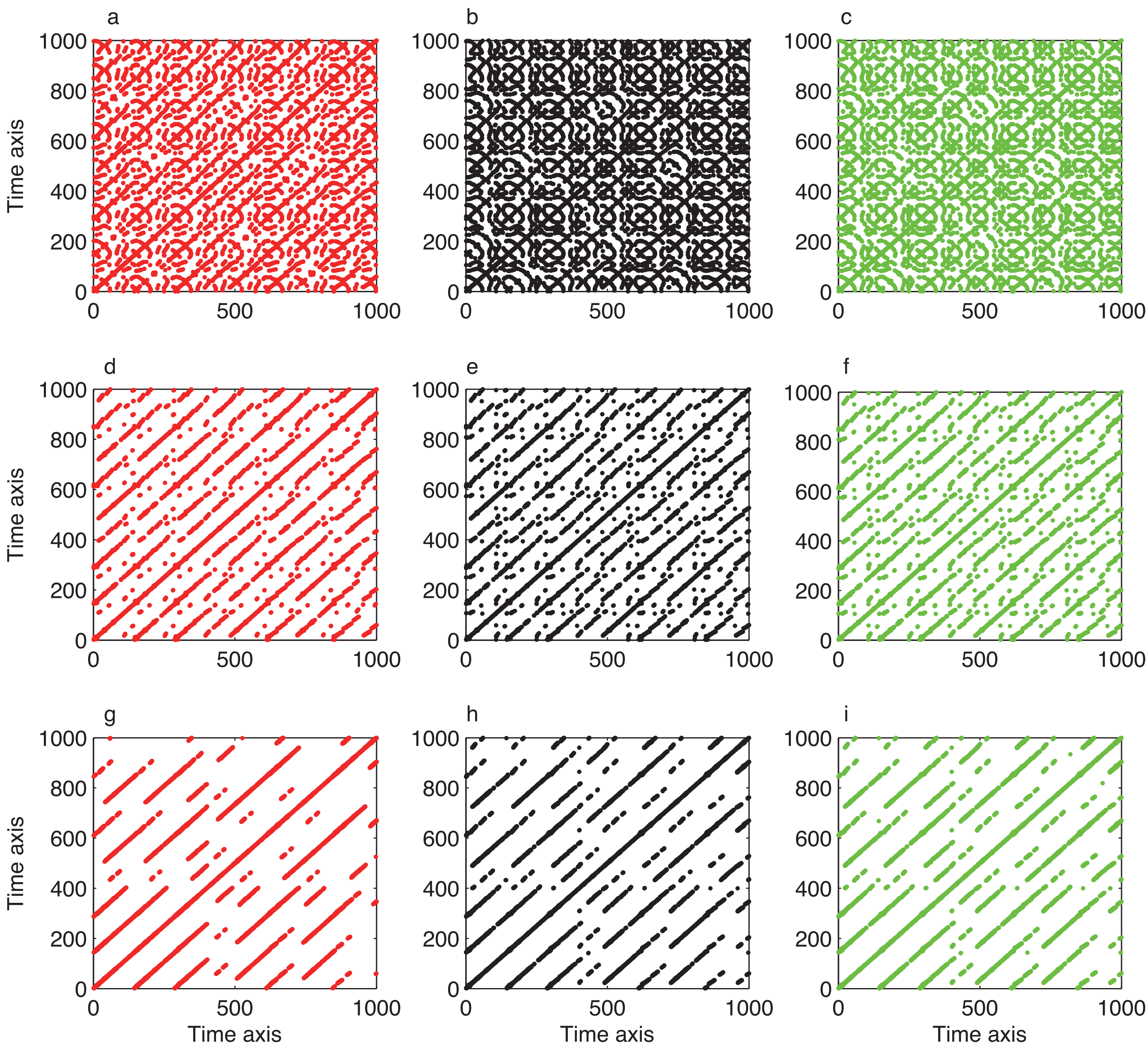
	Estimated using InDDeCs	Known in the literature
Hénon map (mean over $\lambda \in [0.01, 0.1]$)	1.2307 ± 0.0135	1.21 ± 0.01 (Ref. [14])
Lorenz model ($R = 28, \sigma = 10, b = 8/3$) (mean over $\lambda \in [0.01, 0.99]$)	1.8488 ± 0.2967	2.05 ± 0.01 (Ref. [14])

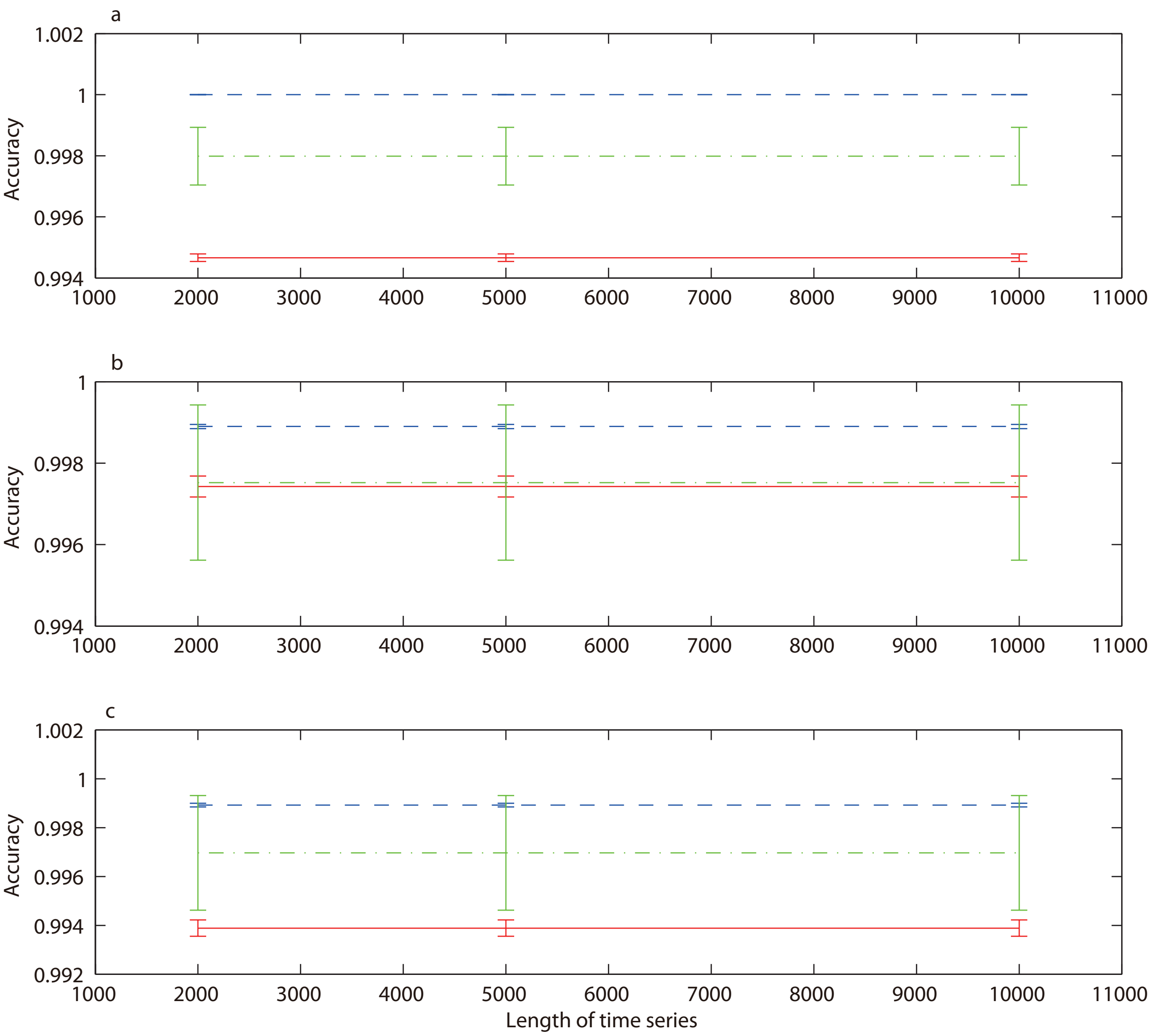
TABLE II. Estimated values using InDDeCs and values known in the literature for maximal Lyapunov exponents. Each values shown with $\pm$ represent the mean and standard deviation for the estimated values.

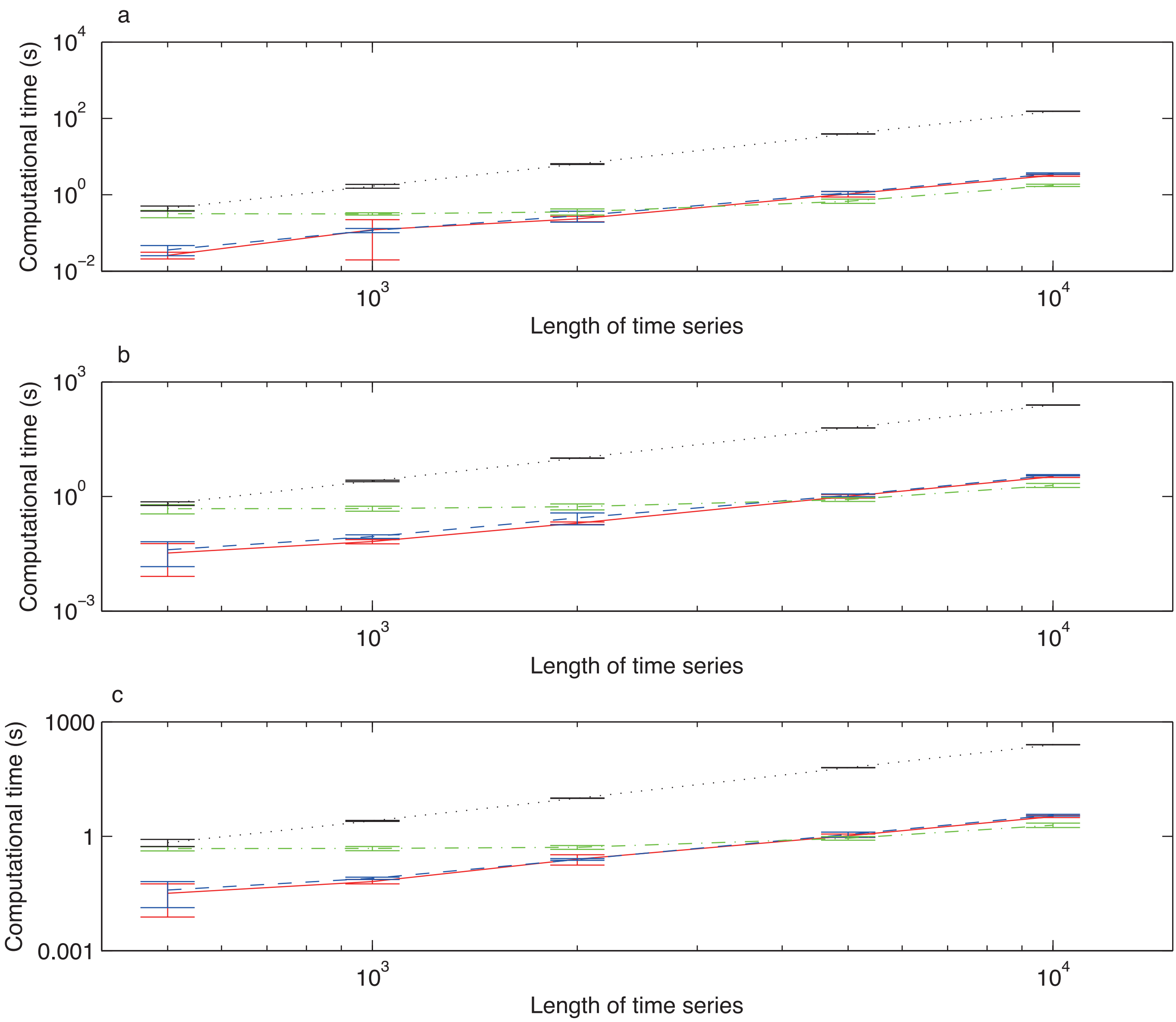
	Estimated using InDDeCs	Known in the literature
Hénon map (bits/obs.) (mean over $\lambda \in [0.01, 0.1]$)	0.6240 ± 0.0075	0.6223 (Ref. [19], metric entropy)
Ikeda map (bits/obs.) (mean over $\lambda \in [0.01, 0.1]$)	0.6651 ± 0.1024	0.7450 (Ref. [19], metric entropy)
Lorenz model (nats/unit time) ($R = 40, \sigma = 16, b = 4$) (mean over $\lambda \in [0.01, 0.99]$)	1.3846 ± 0.4472	1.37 ± 0.08 (Ref. [18])
Rossler model (nats/unit time) (mean over $\lambda \in [0.01, 0.99]$)	0.0608 ± 0.0116	0.073 ± 0.004 (Ref. [18])

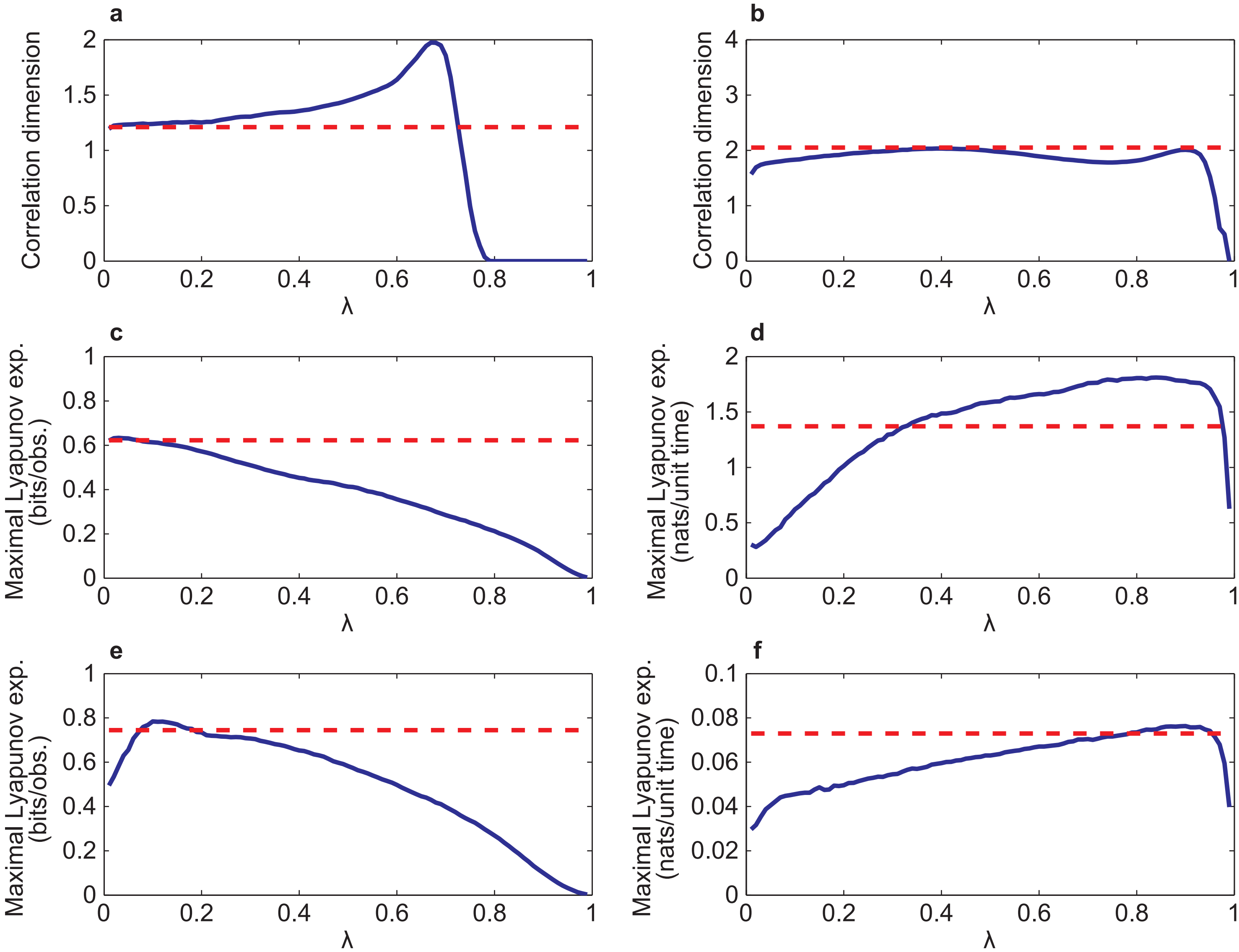
TABLE III. Correlation coefficients between the validated principal driving forces reconstructed from the temperature, the precipitation, the solar irradiance, and the wind speed at Akita, Japan.

	Temperature	Solar irradiance	Precipitation	Wind speed
Temperature	1	0.4315	0.5095	-0.0370
Solar irradiance	0.4315	1	0.3221	-0.0316
Precipitation	0.5095	0.3221	1	0.2774
Wind speed	-0.0370	-0.0316	0.2774	1

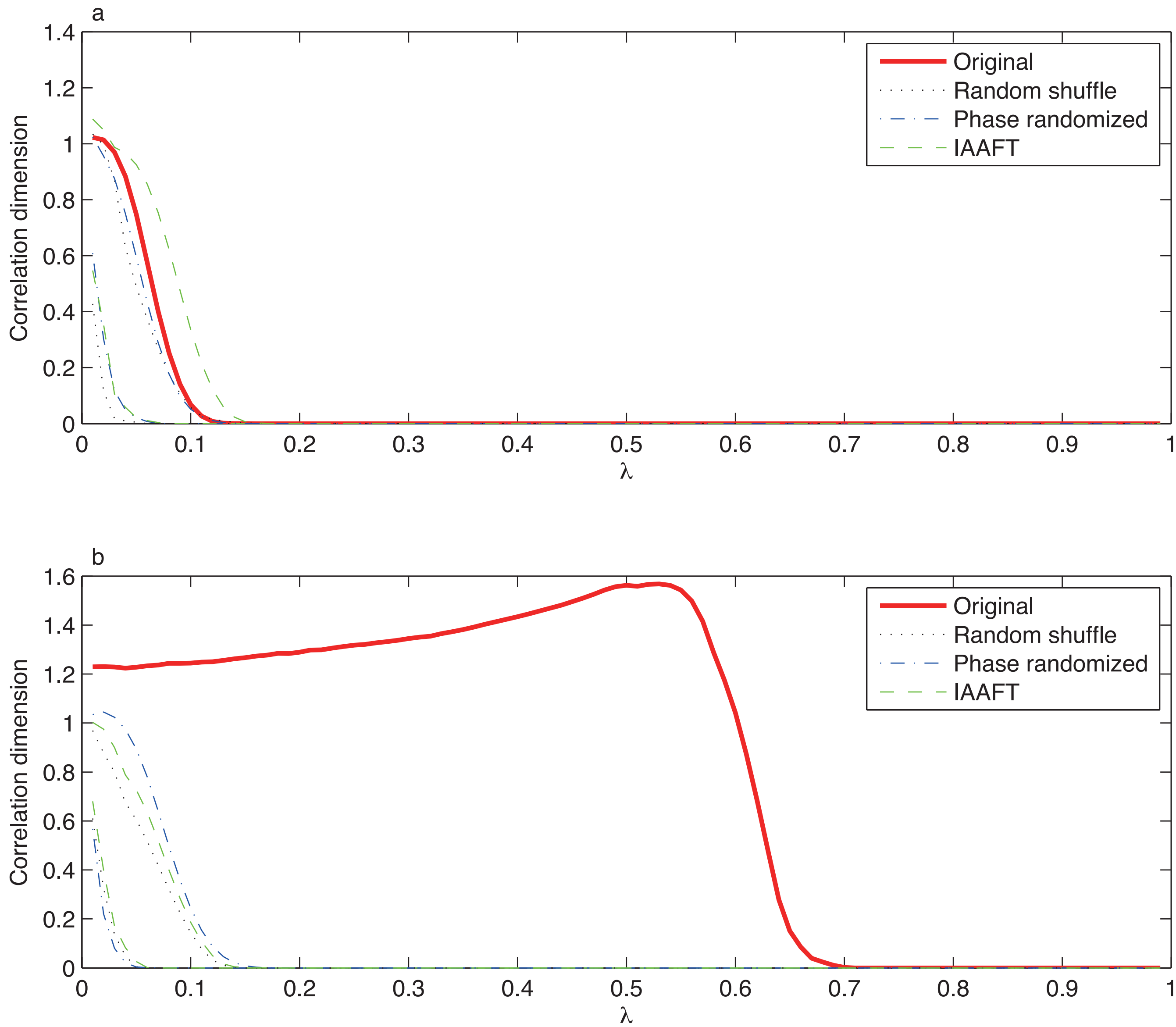

a
b
c
d
e
f
g
h
i
Time axis
Time axis
Time axis
Time axis
Time axis
Time axis
0
500
1000
200
400
600
800

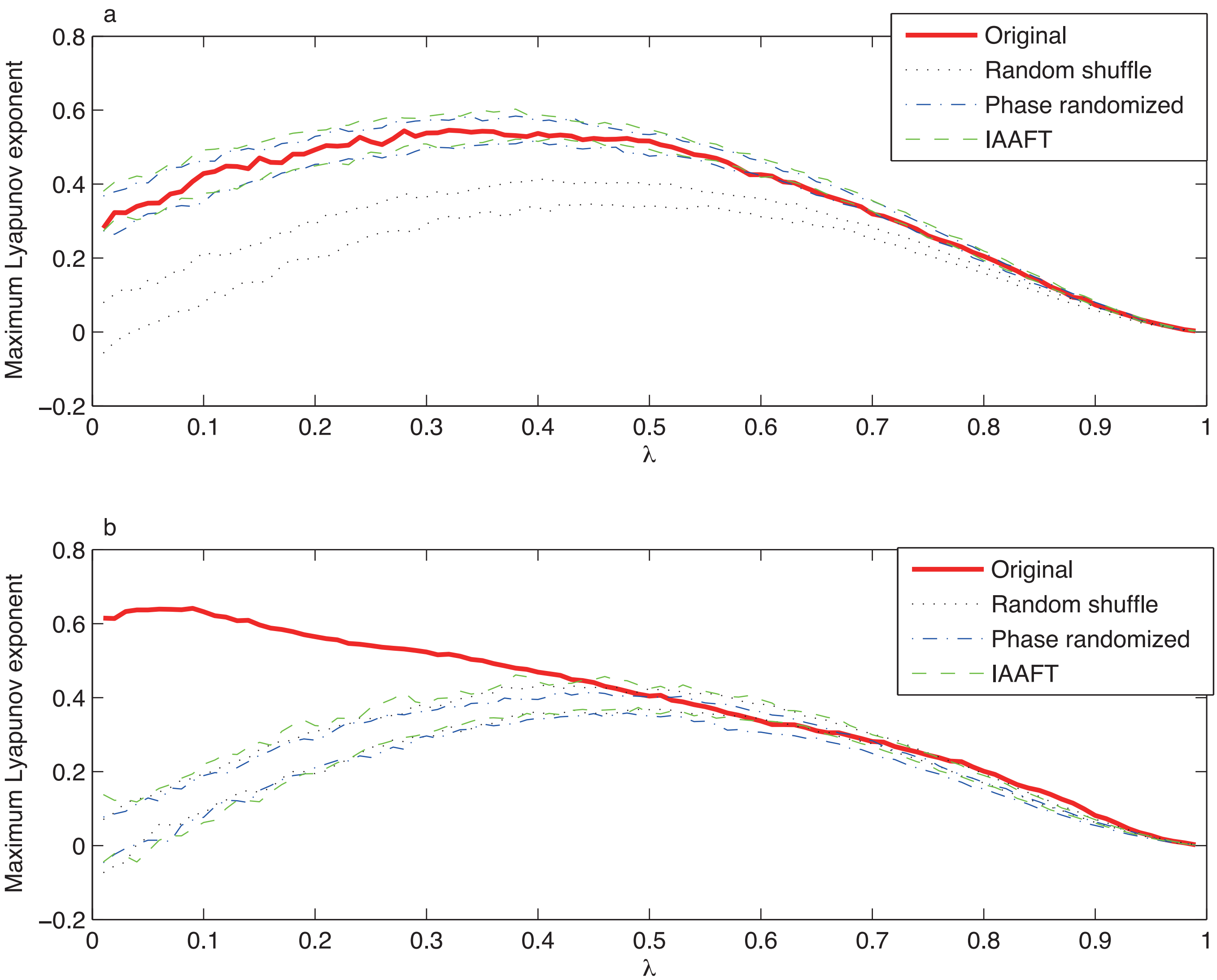

a
b
c
Accuracy
1.002
1
0.998
0.996
0.994
0.992
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
Length of time series


a
10^4
10^2
10^0
10^{-2}
Computational time (s)
10^3
10^4
Length of time series
b
10^3
10^0
10^{-3}
Computational time (s)
10^3
10^4
Length of time series
c
1000
1
0.001
Computational time (s)
10^3
10^4
Length of time series

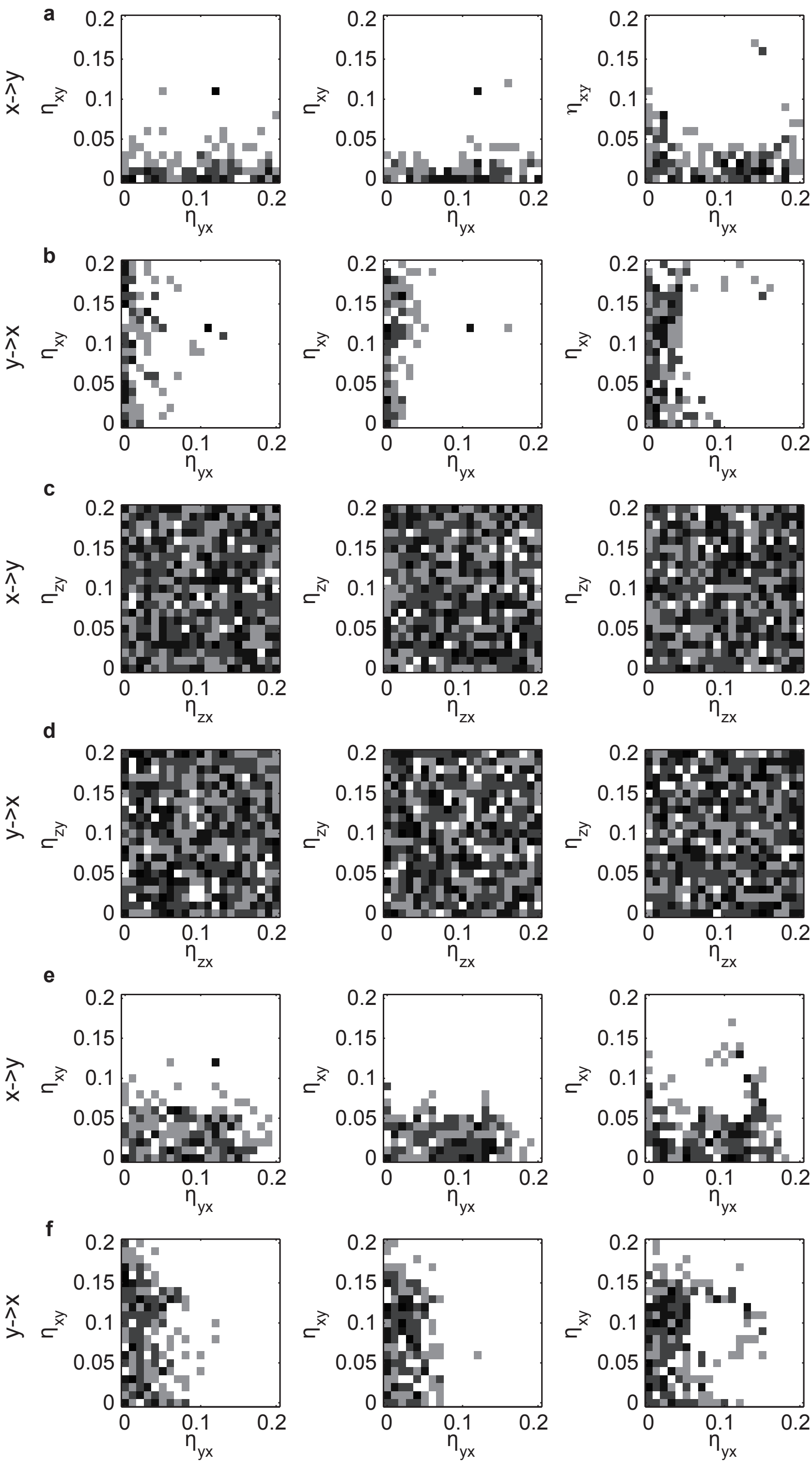


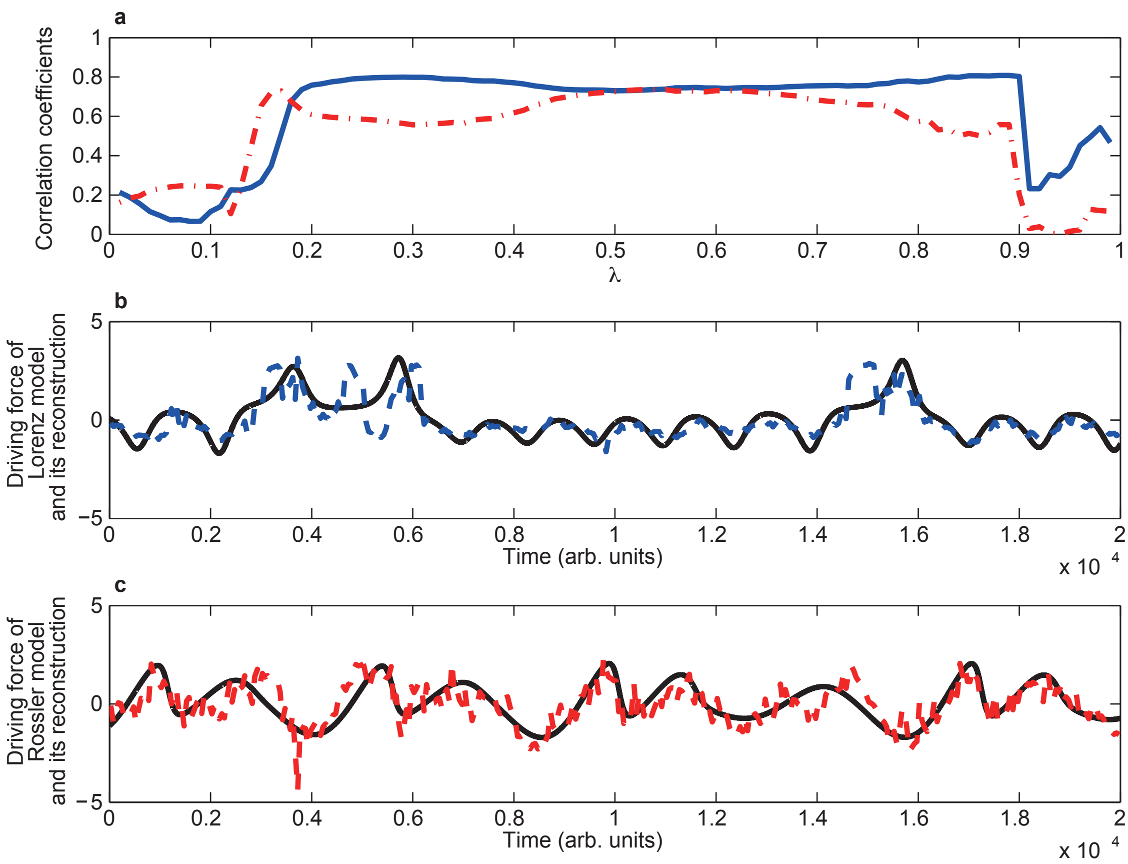
a
Correlation dimension
λ
b
Correlation dimension
λ
c
Maximal Lyapunov exp. (bits/obs.)
λ
d
Maximal Lyapunov exp. (nats/unit time)
λ
e
Maximal Lyapunov exp. (bits/obs.)
λ
f
Maximal Lyapunov exp. (nats/unit time)
λ


a
Correlation dimension
λ
Original
Random shuffle
Phase randomized
IAAFT
b
Correlation dimension
λ
Original
Random shuffle
Phase randomized
IAAFT

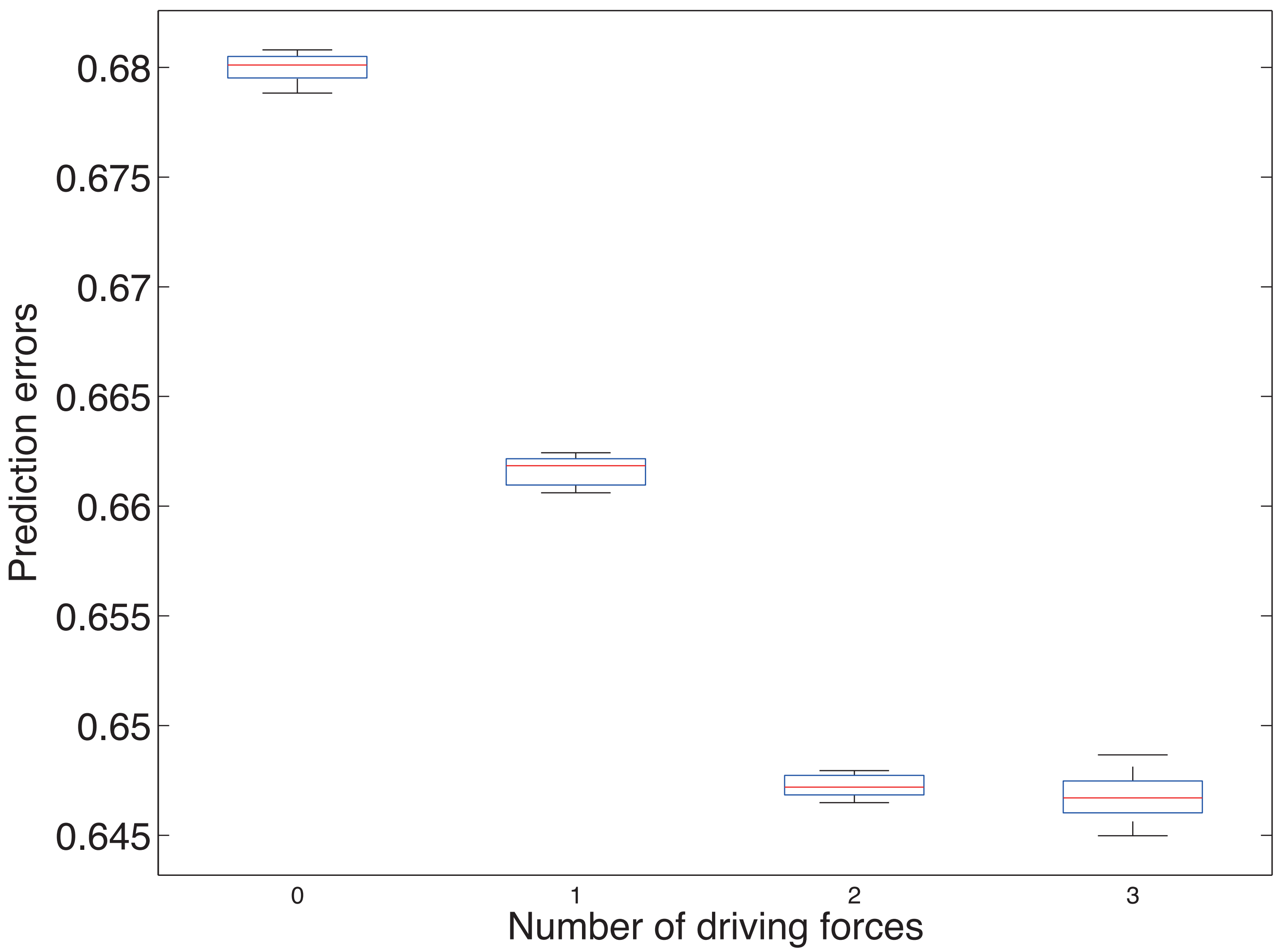


a
b
Original
Random shuffle
Phase randomized
IAAFT
Maximum Lyapunov exponent
λ

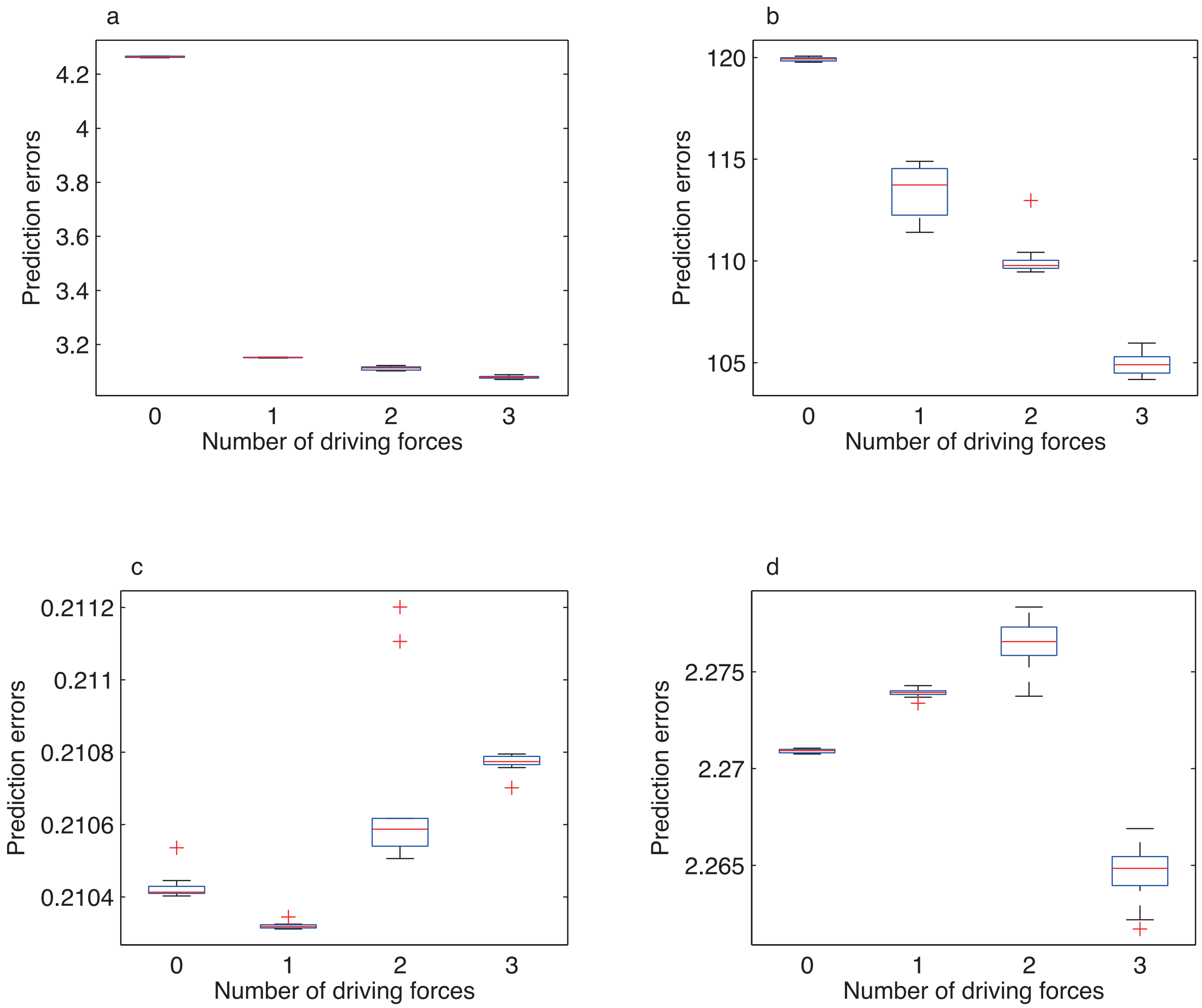

a
b
c
d
e
f
x->y
y->x
x->y
y->x
x->y
y->x
η_{xy}
η_{yx}
η_{zy}
η_{zx}
0
0.05
0.1
0.15
0.2

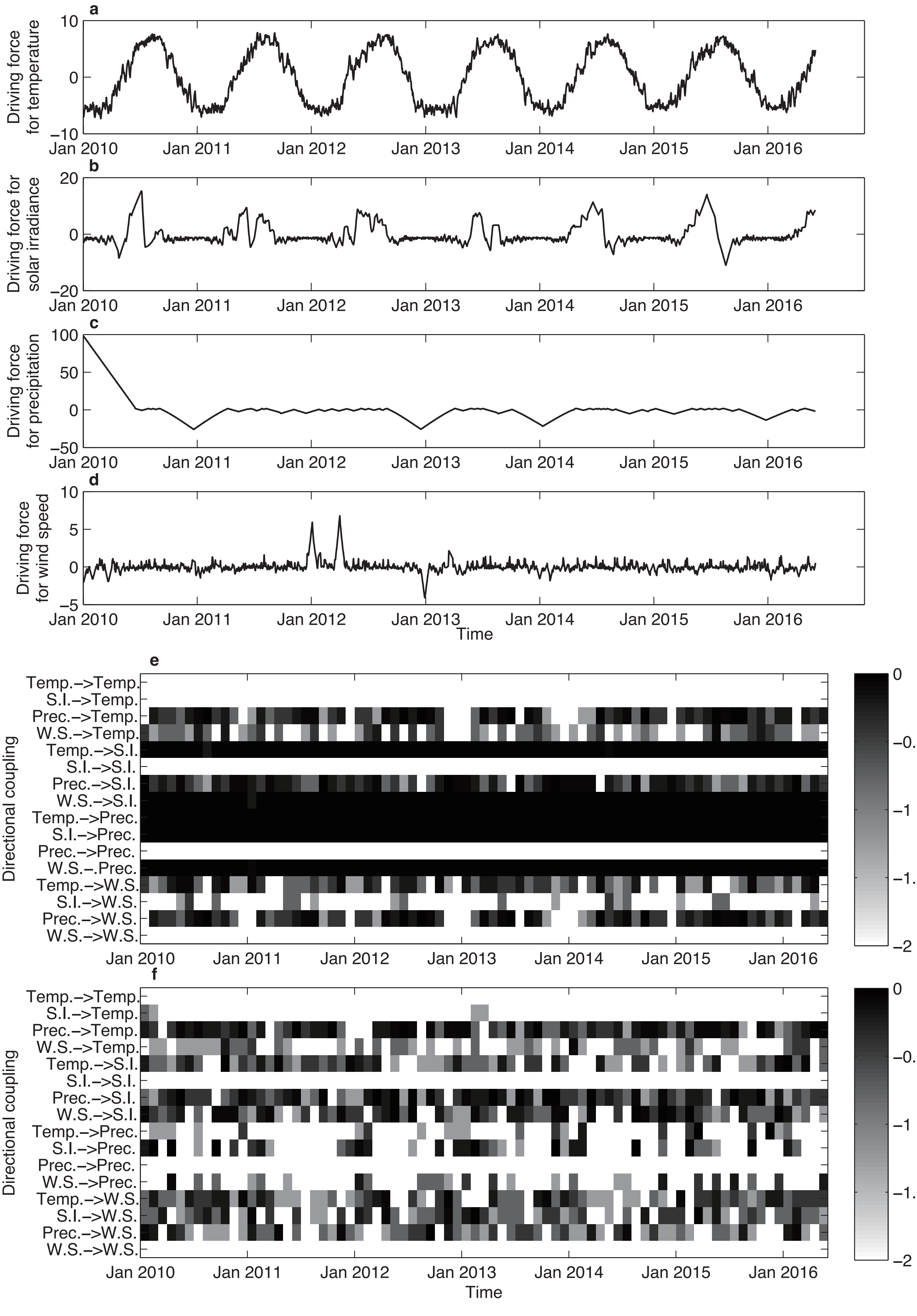


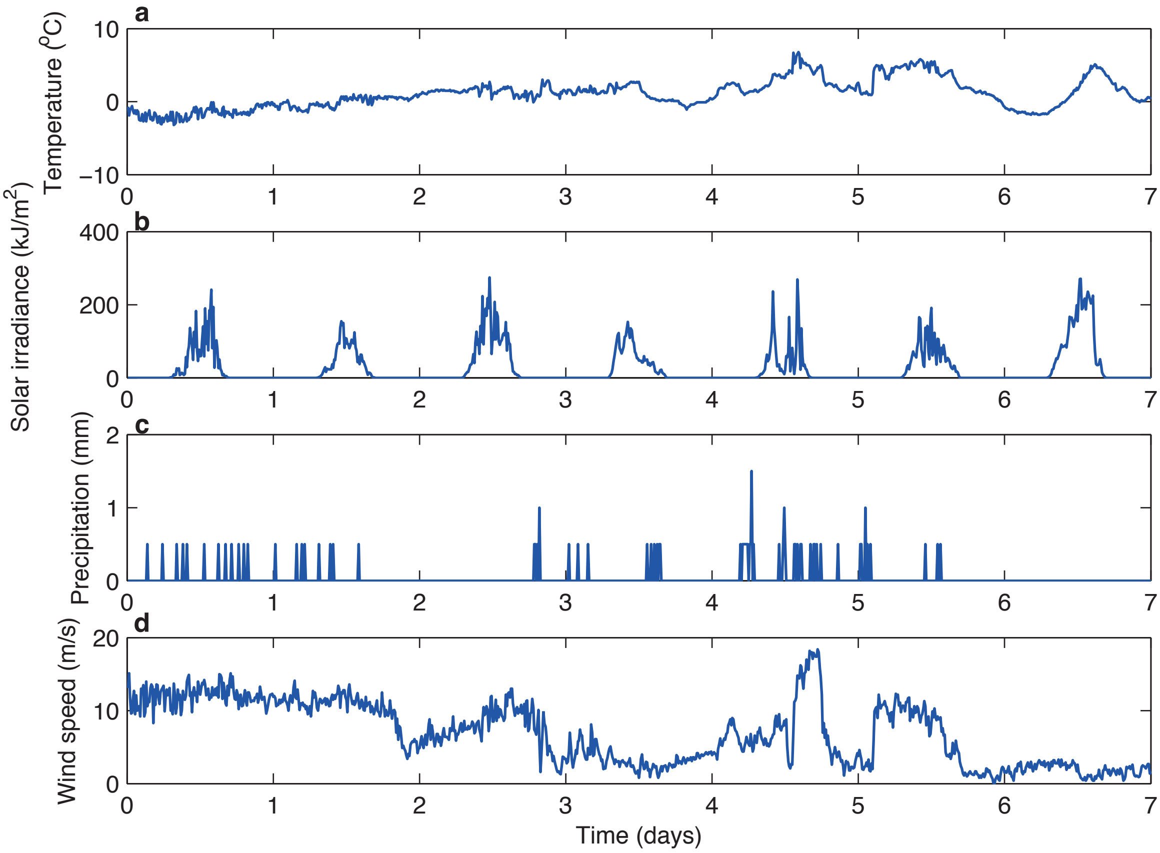
a
Correlation coefficients
1
0.8
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
λ
b
Driving force of
Lorenz model
and its reconstruction
5
0
−5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time (arb. units)
x 10 4
c
Driving force of
Rossler model
and its reconstruction
5
0
−5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time (arb. units)
x 10 4



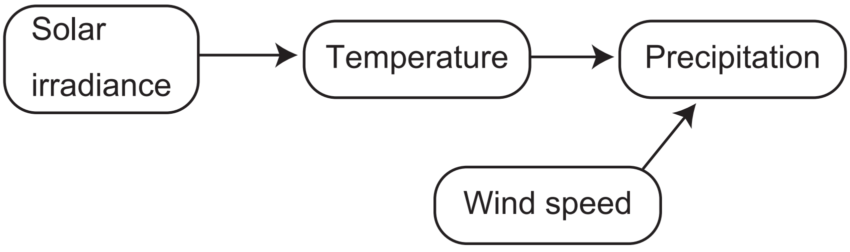
0.68
0.675
0.67
0.665
0.66
0.655
0.65
0.645
Prediction errors
0
1
2
3
Number of driving forces


a
b
c
d
Prediction errors
Number of driving forces

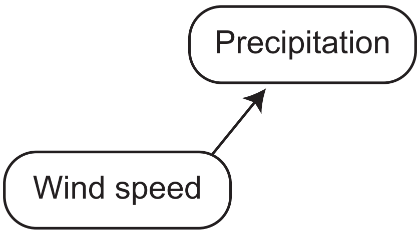

a
Driving force for temperature
10
0
−10
b
Driving force for solar irradiance
20
0
−20
c
Driving force for precipitation
100
50
0
−50
d
Driving force for wind speed
10
5
0
−5
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Time
e
Directional coupling
Temp.–>Temp.
S.I.–>Temp.
Prec.–>Temp.
W.S.–>Temp.
Temp.–>S.I.
S.I.–>S.I.
Prec.–>S.I.
W.S.–>S.I.
Temp.–>Prec.
S.I.–>Prec.
Prec.–>Prec.
W.S.–.Prec.
Temp.–>W.S.
S.I.–>W.S.
Prec.–>W.S.
W.S.–>W.S.
0
−0.5
−1
−1.5
−2
f
Directional coupling
Temp.–>Temp.
S.I.–>Temp.
Prec.–>Temp.
W.S.–>Temp.
Temp.–>S.I.
S.I.–>S.I.
Prec.–>S.I.
W.S.–>S.I.
Temp.–>Prec.
S.I.–>Prec.
Prec.–>Prec.
W.S.–>Prec.
Temp.–>W.S.
S.I.–>W.S.
Prec.–>W.S.
W.S.–>W.S.
0
−0.5
−1
−1.5
−2
Time

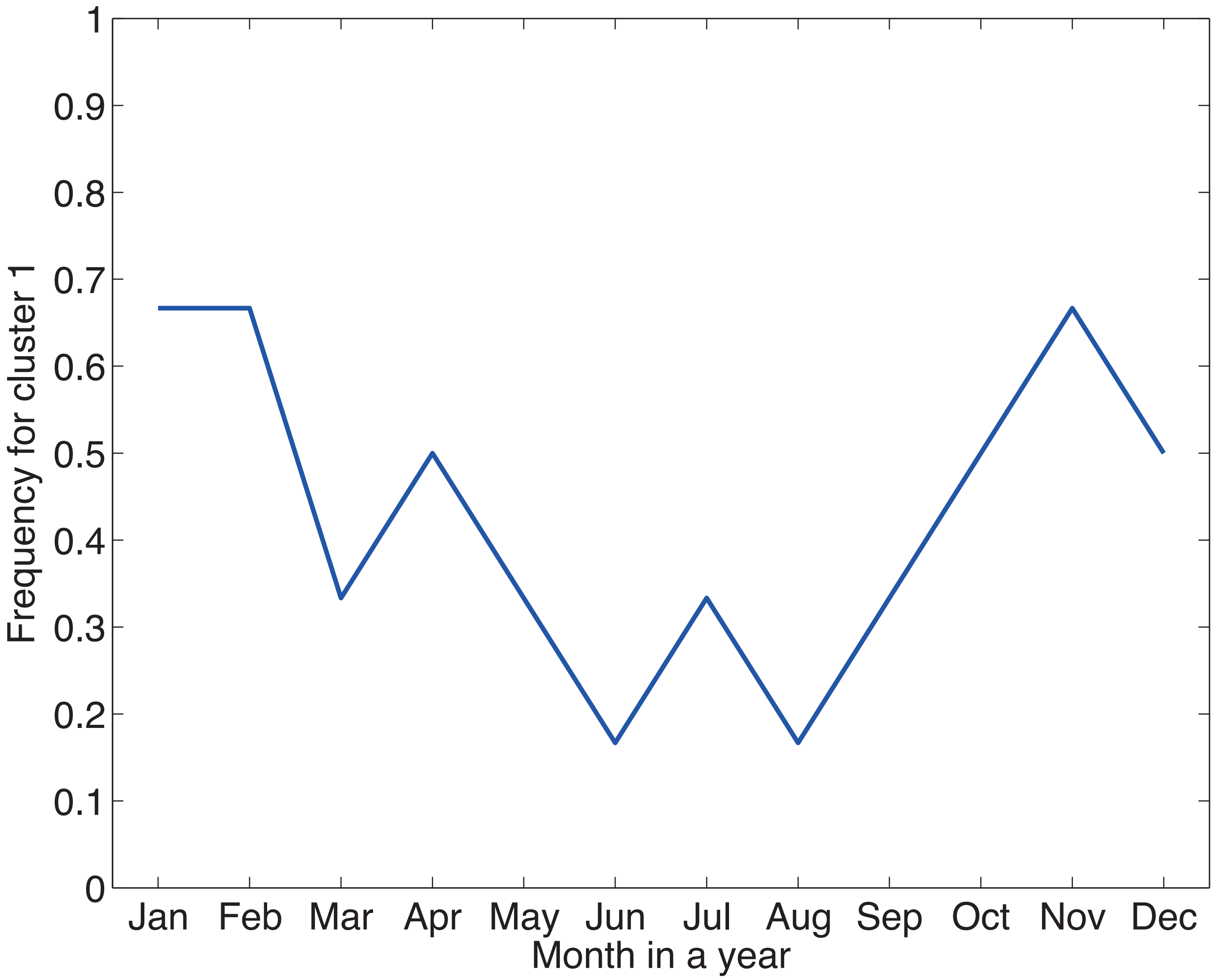

a
Temperature (°C)
10
0
−10
b
Solar irradiance (kJ/m²)
400
200
0
c
Precipitation (mm)
2
1
0
d
Wind speed (m/s)
20
10
0
0 1 2 3 4 5 6 7
Time (days)

a

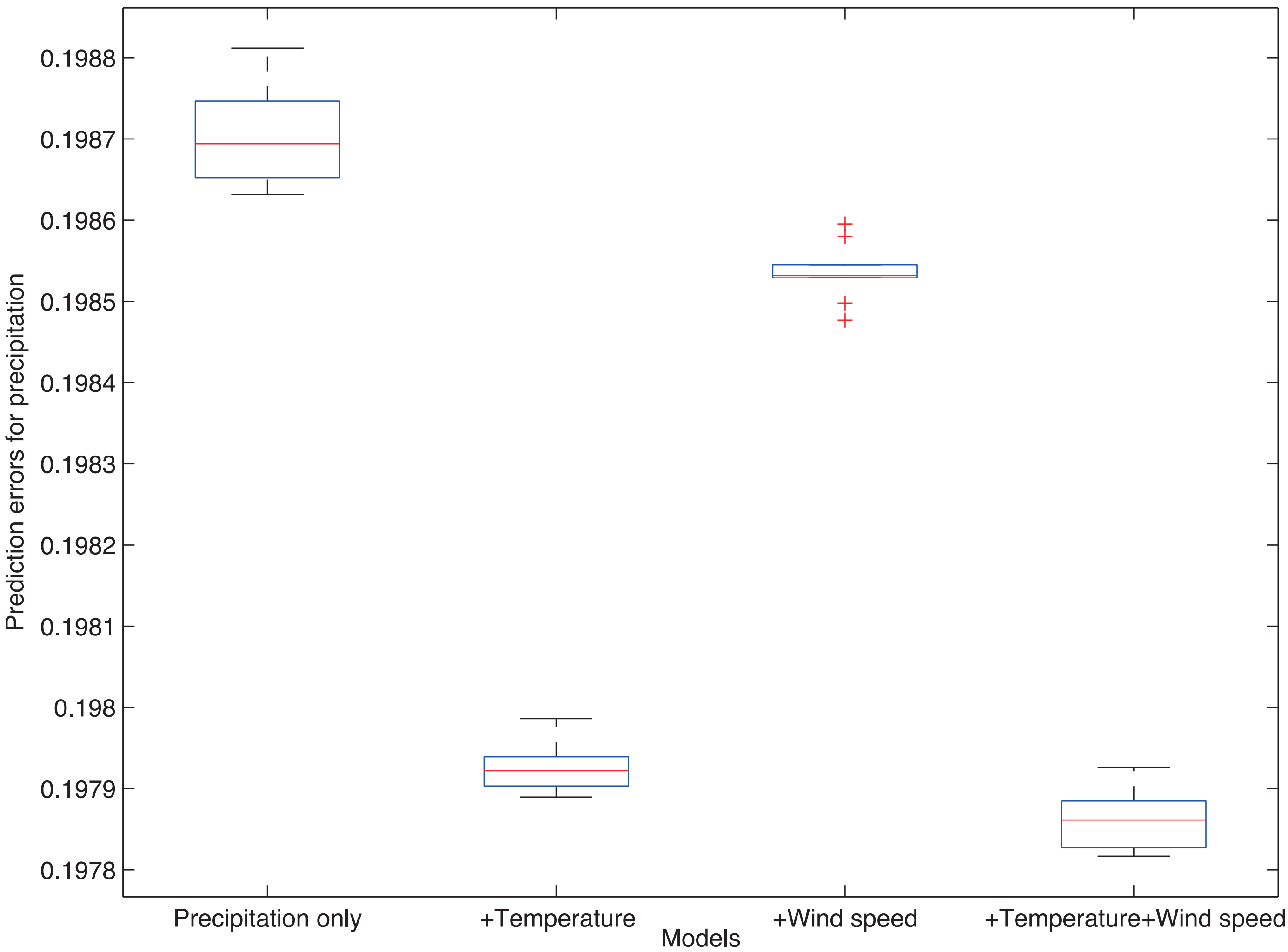

Solar irradiance
Temperature
Precipitation
Wind speed

b


Precipitation
Wind speed



Frequency for cluster 1
Month in a year
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec



Prediction errors for precipitation
0.1988
0.1987
0.1986
0.1985
0.1984
0.1983
0.1982
0.1981
0.198
0.1979
0.1978
Precipitation only
+Temperature
+Wind speed
+Temperature+Wind speed
Models