MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Avoiding Underestimates for Time Series Prediction by State-Dependent Local Integration

Shunya OKUNO, Tomoya TAKEUCHI, Shunsuke HORAI, Kazuyuki AIHARA, and Yoshito HIRATA

METR 2017–22 November 2017

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.keisu.t.u-tokyo.ac.jp/research/techrep/index.html

Title: Avoiding Underestimates for Time Series Prediction by State-Dependent Local Integration

Authors: Shunya Okuno[1,2], Tomoya Takeuchi[1], Shunsuke Horai[1], Kazuyuki Aihara[1], and Yoshito Hirata[1*]

Affiliations:
[1]Institute of Industrial Science, The University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan.
[2]Disaster Reduction & Environmental Engineering Department, Kozo Keikaku Engineering Inc., Nakano-ku, Tokyo 164-0011, Japan.

*Correspondence to: yoshito@sat.t.u-tokyo.ac.jp

Abstract: A series of data points ordered by time is called a time series; examples include financial data and weather observations. Without a mechanistic model for the underlying dynamics, time series prediction is the only solution for inferring future states. However, minima and maxima are commonly underestimated in this approach—a prominent problem when predicting natural disasters, particularly floods. We propose a set of methods for addressing this problem. First, we introduce derivative delay coordinates, which improve prediction when a minimum or maximum is encountered. Second, we refine each prediction with continuous model-free prediction. Third, we integrate these predictions locally using the expert advice method and weighting better predictions more heavily. Our examples demonstrate that we can improve short-term prediction, especially for extremes, when compared with conventional individual model-free prediction based on normal delay coordinates.

One Sentence Summary: We propose a set of methods that avoids the general maxima/minima underestimation problem of time series prediction.

Short title: Avoiding Underestimates for Time Series Prediction

Main Text:

INTRODUCTION

Extracting useful information from various models is a central topic in time series analysis, which is a field of research based on statistics, information theory, and machine learning. In the 1970s, methods were proposed that selected the best model from the available choices (*1*-*3*) (see Nakamura et al. (*4*) for a comparison for nonlinear time series). However, in the 1990s, a new approach that weights all the models rather than selecting the best one has emerged in the context of information theory (*5*), machine learning (*6*), and nonlinear time series modeling (*7*). Among them, the expert advice model by Cesa-Bianchi and Lugosi (*8*) is easy to implement and has been extended with methods that discount losses (*9*) and state-dependent weight models (*10*). Although these extensions assist time series prediction, unfortunately, local changes in state-dependent weighting can have a global influence on the model. In addition, the preparation of an ensemble of time series predictions is another key component of this state-dependent weighting.

One remaining important problem in time series modeling is the underestimation of maxima and minima. This issue is most prominently observed in prediction of natural disasters, particularly

floods (*11, 12*). Kilminster (*13*) provided a partial solution for this underestimation problem by minimizing the prediction error as well as making an invariant measure for the model similar to that of observations. However, it is difficult to extend this approach to high-dimensional dynamics. Thus, to address the underestimation problem, we take a different approach: (i) we prepare a set of coordinates that is good at predicting minima and maxima, (ii) make each model-free prediction continuous to form an ensemble, and (iii) integrate the ensemble of predictions locally in a state-dependent way to improve the time series prediction of minima and maxima and thus the overall prediction. The proposed approach can be applied to high-dimensional dynamics.

METHODS SUMMARY

We first build a set of coordinates that is good at predicting minima and maxima. In the practice of econophysics (*14, 15*), the difference of consecutive measurements is taken (*16*), and this hints at a solution. We observe that (a) when a scalar observation is at a maxima or minima, it is very difficult to directly predict because the local spatial neighbors in state space are usually located inward and they cause the underestimation. But (b) if we take the difference of consecutive measurements, namely, if we look at the first derivative of the measurements, then we can find local spatial points equally likely from the inside and the outside and can prevent underestimation (Fig. 1, A-C). These observations are ours, although Wichard (*16*) also used predictions on differences. If we regard the difference of consecutive measurements as the result of an observation function, the embedding theorems using delay coordinates, or a vector consisting of successive measurements, by Takens (*17*), Sauer et al. (*18*), Stark (*19*), and Stark et al. (*20*) still hold for most cases. Thus, we call this approach the derivative delay coordinates (DeDeCs) method. In addition, we further extend the idea of DeDeCs to mixing the usual observations and their differences by a certain ratio to produce various viewpoints for finding local spatial neighbors, as suggested by Ye and Sugihara (*21*) and prepare a richer ensemble of time series predictions. We call this approach parameterized coordinates. For each prediction, the infinite-dimensional delay coordinates (InDDeCs) (*22*) circumvent the need to choose the embedding dimension and speed up calculations.

Second, we make continuous model-free predictions. When we apply the method of analogues, namely, the zero-th order prediction (*23*), it becomes discontinuous when neighbor points composed of past points enter or leave the neighborhood. Thus, for the current point, we subtract the distances to the top K nearest neighbors from the distances to the $(K+1)$ nearest neighbors to obtain the ratios of their distance differences. Using these ratios, we take a weighted average for the p-steps-ahead points of the top K nearest neighbors to obtain the p-steps-ahead prediction (Fig. 1D). Constructing the prediction this way ensures that the obtained prediction becomes continuous: when the Kth nearest neighbor and the $(K+1)$th nearest neighbor are exchanged, the weight for the Kth nearest neighbor becomes zero. We call this model the local continuous model (LoCoMo).

Third, we combine the ensemble of time series predictions locally so that the integration of the predictions becomes state-dependent. The idea for the integration is similar to that of LoCoMo. Suppose that we have, as a meta model, a set of representative points chosen from past time points in the InDDeCs. Here the meta model is used to infer the current point during the integration, while representative points are a subset of the past time points for describing the meta model. In addition, suppose that for each of these points, a set of individual weights is

defined for combining the various time series predictions. Given the current point, we find the (K'+1) closest representative points and their distances. For the top K' closest representative points, we subtract their distances to the current point from the distance of the current point to the (K'+1)th closest representative point. We let the ratios of the distance differences be meta weights, and use the meta weights to take a weighted average of the individual weights for various predictions to obtain a state-dependent set of weights for the current point (Fig. 1E). Using this state-dependent set of weights, we can merge the various time series predictions locally without their global influence. In addition, we update the weights locally by considering the absolute differences in the predicted values as well as the first- and second-order differences so that the proposed prediction has the appropriate dynamical behavior. Defining the state-dependent set of weights in this way makes the integrated prediction continuous, even after integration because where the (K')th and (K'+1)th representative points are exchanged, the meta weight for the (K')th representative point becomes zero and does not influence the resulting state-dependent set of weights. We call this integration state-dependent expert advice (StaDEA) because it can be regarded as an extension of expert advice (*8*).

RESULTS

We demonstrated the proposed method using the toy models of Lorenz (*24*) and Rössler (*25*). The proposed prediction achieves the smallest prediction errors in these examples (Figs. 2A and S1A for whole data and Figs. 2B and S1B when we restricted our evaluations on maxima and minima only). Further, local minima and maxima are predicted better than when normal InDDeCs or DeDeCs are used, as shown in Fig. 2, C-F, and Fig. 1, C-F.

To further demonstrate the proposed method, we applied the proposed prediction to flood forecasting using a previous competition dataset (see (*26*) for a related competition). When we predicted 24 hours ahead, the highest peak in the year was not underestimated (Figs. 3 and S2; compare these figures with Figs. 1 and 2 of (*26*)). Rather than always having a delay from the actual values in the conventional prediction, the proposed method often predicted the peaks sometimes without a delay. As the results, even though we could access the measurements of every 6 hours while in the related competition, access to the measurements of every 15 minutes was possible, our root mean square error was smaller than any of neural networks among the competition entries of (*26*) (see Table S1). Furthermore, when we compare our prediction with the ARMAX models under the conditions that we have only the measurements of every 6 hours, our prediction yielded the smaller root mean square error than the ARMAX models up to the fourth order of the autoregressive term. The order of the moving average term is zero. In this light, we believe that the proposed method advances the practice of time series prediction to a great extent.

There is another note on peak predictions in this flood example. The proposed method tended to overestimate peaks slightly. But, in this application, overestimates have less harm than underestimates because if we underestimate floods, we have more social risks. How to further improve our prediction is an open future problem.

Lastly, we applied the proposed prediction to forecasting wind power, which is important for power-grid management, especially for abrupt power changes (*27*), which can be regarded as maxima or minima of the change rates. We used a dataset consisting of data from the Tohoku

area of Japan (*28*). We predicted the increments for every hour and transformed the data back to the original axis of the area's total wind power in GW. We compared our methods with an autoregressive and moving average (ARMA) method, which is dominantly used for predicting hourly wind power outputs (*29*). The proposed DeDeCs and StaDEA achieved smaller errors than ARMA up to 4 hours ahead (Fig. 4, A and B). For each prediction (Fig. 4, C-F), the DeDeCs roughly grasped the increasing trends (Fig. 4D) and decreasing trends (Fig. 4, E and F), which helped move the StaDEA's predictions towards the truth and improved its overall prediction performance (Fig. 4A). Even if we refined the time resolution from 1 hour moving average to 10 minutes moving average, we obtained the similar results (Fig. S3).

DISCUSSION

The proposed method can also be applied to other variations of the method of analogues. For instance, we applied the idea of barycentric coordinates with linear programming (*30*) instead of InDDeCs with LoCoMo to the proposed method. The proposed prediction with barycentric coordinates achieves the smallest prediction errors in the toy models (Fig. S4). Thus, our results do not seem to depend on specific prediction models, although we intended to use InDDeCs to take into account long-term correlations in real time series of wind powers and floods. Furthermore, it is known that flood series have multifractal properties (*31*, *32*). Thus, how to exploit the multifractal properties for time series prediction seems apparently an important future question.

The proposed method is somewhat tolerant to observational noise. We checked the robustness of the proposed method using the toy models of Lorenz and Rössler with 5% Gaussian observational noise. The results show that the proposed method obtains the smallest prediction error for these noisy data (Fig. S5).

The proposed method is also robust for the number of neighbors used for StaDEA. Even if we changed the number of neighbors for StaDEA, we had the similar results as shown in Fig. S6.

The proposed method also has some limits. As we can see in Figs. 2C-F and S1C-F, maxima and minima tended to be underestimated when the prediction steps become larger. It is probably because even using the proposed method, we could not have sufficient spatial or temporal resolutions, and hence could not isolate neighbors for predicting maxima and minima accurately. How to further improve the prediction accuracy is an open problem.

Using the proposed prediction method, we have almost overcome the problem of underestimation commonly observed in time series prediction. Because we do not explicitly consider state space, we can treat the underlying dynamics well, even if it exists in a high-dimensional space. In addition, the algorithm scales with the length of time series. Thus, the proposed prediction runs online and is suitable for data streams. We hope that the proposed method for integrating individual predictions becomes an enduring bridge between the machine learning and nonlinear dynamics communities, as our findings may also be applied to other problems associated with statistical regression, and not just to time series prediction.

MATERIALS AND METHODS

Local continuous model (LoCoMo)

Suppose that a time series is given by $x(t) \in M$, where M is a manifold represented by delay coordinates or the infinite-dimensional delay coordinates (InDDeCs). We also need to define a distance function $d: M \times M \to \{0\} \cup R^+$; let $D(t) = \{d(x(t), x(s)) | s = 1,2, \dots, t-1\}$. We may define a distance function in normal delay coordinates (*17*, *18*) or in InDDeCs, as we defined in (*22*). We then sort the elements of $D(t)$ in ascending order and record the time indices as $\tau_1(t)$, $\tau_2(t)$,…, $\tau_{t-1}(t)$. Using the first $K+1$ components, we determine the ith element $w_i(t)$ for the weights $w(t)$ up to the Kth element as

$$w_i(t) = \frac{d(x(t),x(\tau_{K+1}(t))) - d(x(t),x(\tau_i(t)))}{\sum_{j=1}^{K}\{d(x(t),x(\tau_{K+1}(t))) - d(x(t),x(\tau_j(t)))\}}. \quad \text{(S1)}$$

During the preparation of this manuscript, we noticed that a similar but different weighting function was previously used by McNames (*33*, *34*). We provide the p-steps-ahead prediction using

$$\frac{\sum_{i=1}^{K} w_i(t) x(\tau_i(t)+p)}{\sum_{i=1}^{K} w_i(t)}, \quad \text{(S2)}$$

for usual coordinates, or

$$x(t) + \frac{\sum_{i=1}^{K} w_i(t)\{x(\tau_i(t)+p) - x(\tau_i(t))\}}{\sum_{i=1}^{K} w_i(t)}, \quad \text{(S3)}$$

for DeDeCs. If we define the weights in this way, the constructed model is continuous even if the Kth element and the (K+1) element are exchanged because $w_K(t) = 0$ at that position. In addition, it is a local model and at each point, only (K+1) nearest neighbors influence the prediction. More generally, this type of a weighted average has a mathematical support that guarantees the approximation accuracy (*35*).

Parametrized coordinates

According to Takens' embedding theorem (*17*, *18*), there is a freedom with respect to how we define an observation function. For example, if we have an observable $s_1(t)$, for the future values of which we predict, we may consider delay coordinates by letting

$$(1-\rho_1)s_1(t) + \rho_1(s_1(t) - s_1(t-1)) \quad \text{(S4)}$$

or

$$(1-\rho_1)s_1(t) + \rho_1(1-\rho_2)(s_1(t) - s_1(t-1)) + \rho_1\rho_2(s_1(t) - 2s_1(t-1) + s_1(t-2)) \quad \text{(S5)}$$

be an observable, where $0 \le \rho_i \le 1$ for $i = 1,2$. Because there is a freedom as to how we define an observation function, we can reconstruct a state for the underlying dynamics through the newly created observation function of Eq. (S4). In all the examples except for the flood dataset, we prepared a time series prediction with delay coordinates using Eq. (S4) and mixing Eqs. (S2) and (S3) of the proposed LoCoMo in the proportion of $(1-\rho_1):\rho_1$. In the flood example, we prepared a time series prediction with delay coordinates using Eq. (S5) and employing Eq. (S3) of the proposed LoCoMo. Further, we combined the inputs of the other eight observables $\{s_i(t) | i = 2,3, \dots, 9\}$ at the upstreams as additional coordinates as follows:

$$\sum_{j=1}^{4}\sum_{i=2}^{9}\{r_{i,2j-1}^{k}\, s_i(t-j+1) + r_{i,2j}^{k}\,(s_i(t-j+1) - s_i(t-j))\}, \quad \text{(S6)}$$

where $r_{i,j}^{k}$ is chosen randomly from a uniform distribution between 0 and 0.1. In addition, k runs from 1 to 100, i.e., we consider 100 different such variables. We use the formula of Eq. (S6) so that we can treat a multivariate time series as a set of scalar time series each of which contains

multivariate information. We combined Eq. (S5) with Eq. (S6) using the L_1-norm and the InDDeCs to find neighboring points for predicting the future values using Eq. (S3).

State dependent expert advice (StaDEA)

Suppose that we have L p-steps-ahead predictions at time t, as $\{f_p^l(t)|l = 1,2, \dots, L\}$. In addition, we assume that we have a meta model for the StaDEA that specifies the neighbors of the current point for taking the weighted average of the predictions locally. This meta model is composed of a set of representative points chosen from the past time points represented by the InDDeCs. In addition, the meta model may be implemented by LoCoMo.

Suppose that the meta model specifies the (K'+1) nearest neighbors $u_1^p(t), u_2^p(t), \dots, u_{K'+1}^p(t)$ from the set of the representative points for the p-steps-ahead prediction. In addition, let $\mu_{u_m^p(t)}(t)$ be the row vector describing the weights for expert advice at time t, whose lth element is for the lth prediction $f_p^l(t)$. Moreover, we need to define the accumulated error of $u_m^p(t)$ for the lth prediction of the p-steps-ahead prediction at time t by $E_p^l(\mathrm{t}, u_m^p(t))$, which is initially set to zero.

At each time, we apply the following procedure. First, we obtain the meta weights $v_m^p(t)$ for the top K' representative points ($\mathrm{m} = 1, 2, \dots, K'$) in the meta model as follows:

$$v_m^p(t) = \frac{\tilde{d}\left(u_{K'+1}^p(t), x(t)\right) - \tilde{d}\left(u_m^p(t), x(t)\right)}{\sum_{n=1}^{K'} \tilde{d}\left(u_{K'+1}^p(t), x(t)\right) - \tilde{d}\left(u_n^p(t), x(t)\right)}, \quad \text{(S7)}$$

where $\tilde{d}: M \times M \to \{0\} \cup R^+$ is a distance function for the meta model. The obtained integration model then becomes continuous. If the denominator equals zero, we assign 1/K' to all $v_m^p(t)$.

Second, we integrate the predictions by

$$\hat{f}_p(t) = \left[\sum_{m=1}^{K'} v_m^p(t)\mu_{u_m^p(t)}(t)\right]\left[f_p^l(t)\right], \quad \text{(S8)}$$

where $\left[f_p^l(t)\right]$ is a column vector whose lth component is $f_p^l(t)$.

Third, after obtaining new data point $s_1(t+1)$, or the observable for which we predicted, we update the accumulated prediction errors $E_p^l(t, u_m^p(t-p+1))$ by

$$E_p^l(\mathrm{t}, u_m^p(t-p+1)) = \theta E_p^l\left(\mathrm{t}-1, u_m^p(t-p+1)\right) + (1-\theta)v_m^p(t-p+1)\Big\{\big|s_1(t+1) - f_p^l(t-p+1)\big| + \alpha\big|(s_1(t+1) - s_1(t)) - \left(f_p^l(t-p+1) - f_p^l(t-p)\right)\big| + \beta\big|(s_1(t+1) - 2s_1(t) + s_1(t-1)) - (f_p^l(t-p+1) - 2f_p^l(t-p) + f_p^l(t-p-1))\big|\Big\}, \quad \text{(S9)}$$

for the top K' representative points, and

$$E_p^l(\mathrm{t}, u_m^p(t-p+1)) = E_p^l\left(\mathrm{t}-1, u_m^p(t-p+1)\right), \quad \text{(S10)}$$

for the other representative points, where $0 < \theta < 1$. Using the first- and second-order differences, as shown in Eq. (S9), we can take into account the behavior of the underlying dynamics including the velocity and the acceleration.

Fourth, we update the weights for expert advice using

$$\mu_{u_m^p(t-p+1),l}(t+1) = \frac{Exp[-\zeta E_p^l(\mathrm{t},u_m^p(t-p+1))]}{\sum_{l'} Exp[-\zeta E_p^{l'}(\mathrm{t},u_m^p(t-p+1))]}. \quad \text{(S11)}$$

Parameter settings

We set $\theta = 0.5$ throughout the paper except for the case of the flood example, where we also use $\theta = 0.99$. We chose 3 nearest neighbors, i.e., *K'=3*, for the StaDEA if not mentioned. In addition, we set the decaying factor λ of InDDeCs to $\lambda_i = 0.5^{0.5^{0.25i}}$ (*i=0,1,…,36*), following the work of (*22*). We chose to use 10 nearest neighbors for the InDDeCs.

For the barycentric coordinates with linear programming, we chose the set of 10, 20, and 40 as the number of nearest neighbors, and chose the set of 3, 7, and 10 for the embedding dimensions.

Moreover, we chose the mixing rate ρ_1 to be between 0 and 1 evenly every 0.05 except for the flood dataset. In the flood dataset, we chose the mixing rates ρ_1 and ρ_2 between 0 and 1 evenly every 0.1.

ARMA method

An ARMA model of the form $x_t = \mu + \sum_{i=1}^{p} \alpha_i \, x_{t-i} + \sum_{i=1}^{q} \beta_i \, \varepsilon_{t-i}$, where $\mathrm{x_t}$ stands for the hourly wind power output and ε_t is white noise with mean zero and the variance σ^2, was built for each month of the evaluation period. The order (p, q) and $p + q + 1$ unknown parameters $(\mu, \alpha_1, \dots, \alpha_p, \beta_1, \dots, \beta_q, \sigma)$ were estimated using the power output data observed in the same month a year ago, e.g., an ARMA model for April 2012 was built using the observations in April 2011, thus we obtained twelve different ARMA models. We used R package "forecast" (*36*) for the parameter estimation and the hourly forecasts up to 6 hours ahead. We chose *p* and *q* so that we minimized the Akaike Information Criterion (*37*).

ARIMA method

An Autoregressive Integrated Moving Average (ARIMA) model of the form $\left(1 - \sum_{i=1}^{p} \alpha_i \, L^i\right)(1 - L)^d x_t = (1 + \sum_{i=1}^{q} \beta_i \, L^i)\varepsilon_t$, where L is the lag operator, $\mathrm{x_t}$ stands for the 10-minute average wind power output and ε_t is white noise with mean zero and the variance σ^2, was built for each month of the evaluation period. The order (p, d, q) and $p + q + 1$ unknown parameters $(\alpha_1, \dots, \alpha_p, \beta_1, \dots, \beta_q, \sigma)$ were estimated using the last 1000 observations to forecast the power output of 6 hours (36 points) ahead, e.g., an ARIMA model was built using the observations from 2012/03/25 01:20:00 to 2012/03/31 23:50:00 (1000 observations) to give a forecast of the power output from 2012/04/01 00:00:00 to 2012/04/01 05:50:00 (36 points). We used R package "forecast" (*36*) for the parameter estimation.

Lorenz'63 model dataset

We used the model of Lorenz (*24*) as our first toy example. By choosing the initial condition $(\mathrm{x}, \mathrm{y}, \mathrm{z}) = (0.1, 0.1, 0.1)$, setting the integration step to 0.01, throwing away the initial transients, and recording every 10 points, we generated a time series of length 4,000. We used the first 200 points to generate the meta model and the first 2,000 points as a database. We used the last 1,000 points for evaluating our predictions. We also generated noisy data by adding 5% Gaussian

observational noise to the model.

Rössler model dataset

We used the Rössler model (*25*) for the next example. By choosing the initial condition of $(x, y, z) = (0.1, 0.1, 0.1)$, setting the integration step to 0.1, and throwing away the initial transients, we generated a time series of length 4,000. We used the first 200 points to generate the meta model and the first 2,000 points as a database. We use the last 1,000 points for evaluating our predictions. We also generated noisy data by adding 5% Gaussian observational noise to the model.

Flood dataset

We used a dataset used 10 years ago for the time series prediction competition "Artificial Neural Network Experiment (ANNEX 2005/2006)" organized by Prof. Christian W. Dawson and his collaborators. These datasets consist of three time periods: 1 October 1993 to 31 March 1994, 1 October 1994 to 31 March 1995, and 1 October 1995 to 31 March 1996. In each period, water heights at three upstream sites, rainfall data at five places, and the water height at the target place were measured every 6 hours. Except for these pieces of information, there was no geographical information given. We used the first time period as a database to predict 24 hours ahead in the second time period. See Dawson et al. (*26*) for a similar competition, as we could not find the exact literature containing the results on the datasets we have for comparison.

In this example, we used a meta model consisting of 37 time points sampled every 20 points for the database.

Wind power dataset

The wind power dataset we used was a continuation of a dataset used in (*28*), which was based on a collaboration between the Japan Wind Power Association and the Ogimoto Lab, Institute of Industrial Science, the University of Tokyo. The dataset was provided by the Ogimoto Lab. We used the total amount of wind power generated in the area covered by the Tohoku Electric Power Co., Inc., Japan. The time period was between 1 April 2011 and 31 March 2013. Although the original measurements were every minute, we took an average every hour (Fig. 4) or every 10 minutes (Fig. S3). We predicted the increments of the total amount of wind power in the area up to 6 hours ahead in a resolution of 1 hour (Fig. 4) or 10 minutes (Fig. S3) from the beginning of the dataset. We used the time period between 1 April 2012 and 31 March 2013 for evaluating our predictions. We transformed the predicted time series back to the axis of wind power in the unit of GW to evaluate our predictions.

References and Notes:

1. H. Akaike, A new look at the statistical model identification, *IEEE Trans. Automat. Contr.* **AC-19**, 716-723 (1974).
2. G. Schwarz, Estimating the dimension of a model, *Ann. Stat.* **6**, 461-464 (1978).
3. J. Rissanen, *Stochastic Complexity in Statistical Inquiry* (World Scientific, Singapore, 1989).
4. T. Nakamura, D. Kilminster, K. Judd, A. Mees, A comparative study of model selection methods for nonlinear time series, *Int. J. Bifurcat. Chaos* **14**, 1129-1146 (2004).

5. F. M. J. Willems, Y. M. Shtarkov, T. J. Tjalkens, The context tree weighting method: Basic properties, *IEEE Trans. Inf. Theor.* **41**, 653-664 (1995).
6. N. Littlestone, M. K. Warmuth, The weighted majority algorithm, *Inf. Comput.* **108**, 212-261 (1994).
7. M. B. Kennel, A. I. Mees, Context-tree modeling of observed symbolic dynamics, *Phys. Rev. E* **66**, 056209 (2002).
8. N. Cesa-Bianchi, G. Lugosi, *Prediction, Learning, and Games* (Cambridge University Press, Cambridge, UK, 2006).
9. A. Chernov, F. Zhadanov, Prediction with expert advice under discounted loss, In *Proc. of ALT 2010, Lecture Notes in Artificial Intelligence* **6331**, 255-269 (2010).
10. B. C. Csáji, A. Kovács, J. Váncza, Adaptive aggregated predictions for renewable energy systems, In *Adaptive Dynamic Programming and Reinforcement Learning ADPRL), 2014 IEEE Symposium on*, pages 1-8, IEEE, 2014.
11. T. S. Hu, K. C. Lam, S. T. Ng, River flow time series prediction with a range-dependent neural network, *Hydrol. Sci. J.* **46**, 729-745 (2001).
12. P. C. Nayak, K. P. Sudheer, D. M. Rangan, K. S. Ramasastri, A neuro-fuzzy computing technique for modeling hydrological time series, *J. Hydrol.* **291**, 52-66 (2004).
13. D. Kilminster, "Modelling Dynamical Systems via Behaviour Criteria," PhD thesis, Department of Mathematics & Statistics, University of Western Australia, 2002.
14. R. N. Mantegna, H. E. Stanley, *An Introduction to Econophysics: Correlation and Complexity in Finance* (Cambridge University Press, Cambridge, UK, 2000).
15. J. D. Wichard, C. Merkwirth, M. Ogorzałek, Detecting correlation in stock market, *Physica A* **344**, 308-311 (2004).
16. J. D. Wichard, An adaptive forecasting strategy with hybrid ensemble models, In *Proc. of 2016 International Joint Conference on Neural Networks (IJCNN)*, 24-29 July 2016, DOI:10.1109/IJCNN.2016.7727375.
17. F. Takens, Detecting strange attractors in turbulence, *Lect. Notes Math.* **898**, 366-381 (1981).
18. T. Sauer, J. A. Yorke, M. Casdagli, Embeddology, *J. Stat. Phys.* **65**, 579-616 (1991).
19. J. Stark, Delay embeddings for forced systems. I. deterministic forcing, *J. Nonlinear Sci.* **9**, 255-332 (1999).
20. J. Stark, D. S. Broomhead, M. E. Davies, J. Huke, Delay embeddings for forced systems. II. stochastic forcing, *J. Nonlinear Sci.* **13**, 519-577 (2003).
21. H. Ye, G. Sugihara, Information leverage in interconnected ecosystems: Overcoming the curse of dimensionality, *Science* **353**, 922-925 (2016).
22. Y. Hirata, T. Takeuchi, S. Horai, H. Suzuki, K. Aihara, Parsimonious description for predicting high-dimensional dynamics, *Sci. Rep.* **5**, 15736 (2015).
23. H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis* (Cambridge University Press, Cambridge UK, ed. 2, 2004).
24. E. N. Lorenz, Deterministic nonperiodic flow, *J. Atmos. Sci.* **20**, 130-141 (1963).
25. O. E. Rössler, An equation for continuous chaos, *Phys. Lett.* **57A**, 397-398 (1976)
26. C. W. Dawson et al., A comparative study of artificial neural network techniques for river stage forecasting, In *Proc. of International Joint Conference on Neural Networks*, Montreal, Canada, July 31-August 4, 2005, pp. 2666-2670.
27. C. Gallego-Castillo, A. Cuerva-Tejero, O. Lopez-Garcia, A review on the recent history of wind power ramp forecasting, *Renew. Sust. Energ. Rev.* **52**, 1148-1157 (2015).
28. T. Ikegami, K. Kataoka, K. Ogimoto, T. Saitou, Development of wind power data for power

supply-demand analysis and analysis of long-term wind power variability, *IEEJ T. Power Energ.* **134**, 236-247 (2014) (DOI:10.1541/ieejpes.134.236) (in Japanese).
29. M. Milligan, M. Schwartz, Y. Wan, Statistical wind power forecasting models: Results for U.S. wind farms, National Renewable Energy Laboratory, 2003.
30. Y. Hirata, M. Shiro, N. Takahashi, K. Aihara, H. Suzuki, P. Mas, Approximating high-dimensional dynamics by barycentric coordinates with linear programming, *Chaos* **25**, 013114 (2015).
31. G. Pandey, S. Lovejoy, D. Schertzer, Multifractal analysis of daily river flows including extremes for basins of five to two million square kilometres, one day to 75 years, *J. Hydrol.* **208**, 62-81 (1998).
32. R. Deidda, R. Benzi, F. Siccardi, Multifractal modeling of anomalous scaling laws in rainfall, *Water Resour. Res.*, **35**, 1853-1867 (1999).
33. 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.
34. J. McNames, Local averaging optimization for chaotic time series prediction, *Neurocomputing* **48**, 279-297 (2002).
35. F. Ferraty, P. Vieu, Nonparametric models for functional data, with application in regression, time series prediction and curve discrimination, *J. Nonparametr. Stat.* **16**, 111-125 (2004).
36. R. Hyndman, Y. Khandakar, Automatic time series forecasting: the forecast package for R, *J. Stat. Softw.* **26**, 1-22 (2008).
37. H. Akaike, A new look at the statistical model identification, *IEEE Trans. Automat. Contr.* **AC-19**, 716-723 (1974).

Acknowledgments: We thank Professor Christian W. Dawson (Loughborough University) and Professor Kazuhiko Ogimoto (the University of Tokyo) for letting us use their flood and wind power datasets, respectively. To obtain these datasets, please contact them directly.

Funding: This manuscript is partially based on the results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO). This research is partially supported by the Kozo Keikaku Engineering Inc., JSPS KAKENHI Grant Number 15H05707, and CREST, JST.

Author contributions: All the authors conceived the study. S.O. and Y.H. developed the method. S.O., T.T., and Y.H. conducted the numerical experiments and analyzed the datasets. K.A. and Y.H. got the permissions for analyzing the real datasets of wind power and floods. S.H. and K.A. provided comments during these numerical experiments and the data analysis. All the authors wrote and revised the manuscript, and agreed to submit the final version of the manuscript.

Competing interests: The authors declare that they have no competing interests.

Data and materials availability: All the pieces of information needed to evaluate the conclusions in the paper are present in the paper. As for the real datasets of floods and wind power, please make a contact with Professor Christian W. Dawson (Loughborough University) and Professor Kazuhiko Ogimoto (the University of Tokyo) as described above. The other additional data related to this paper are available from the corresponding author upon request.

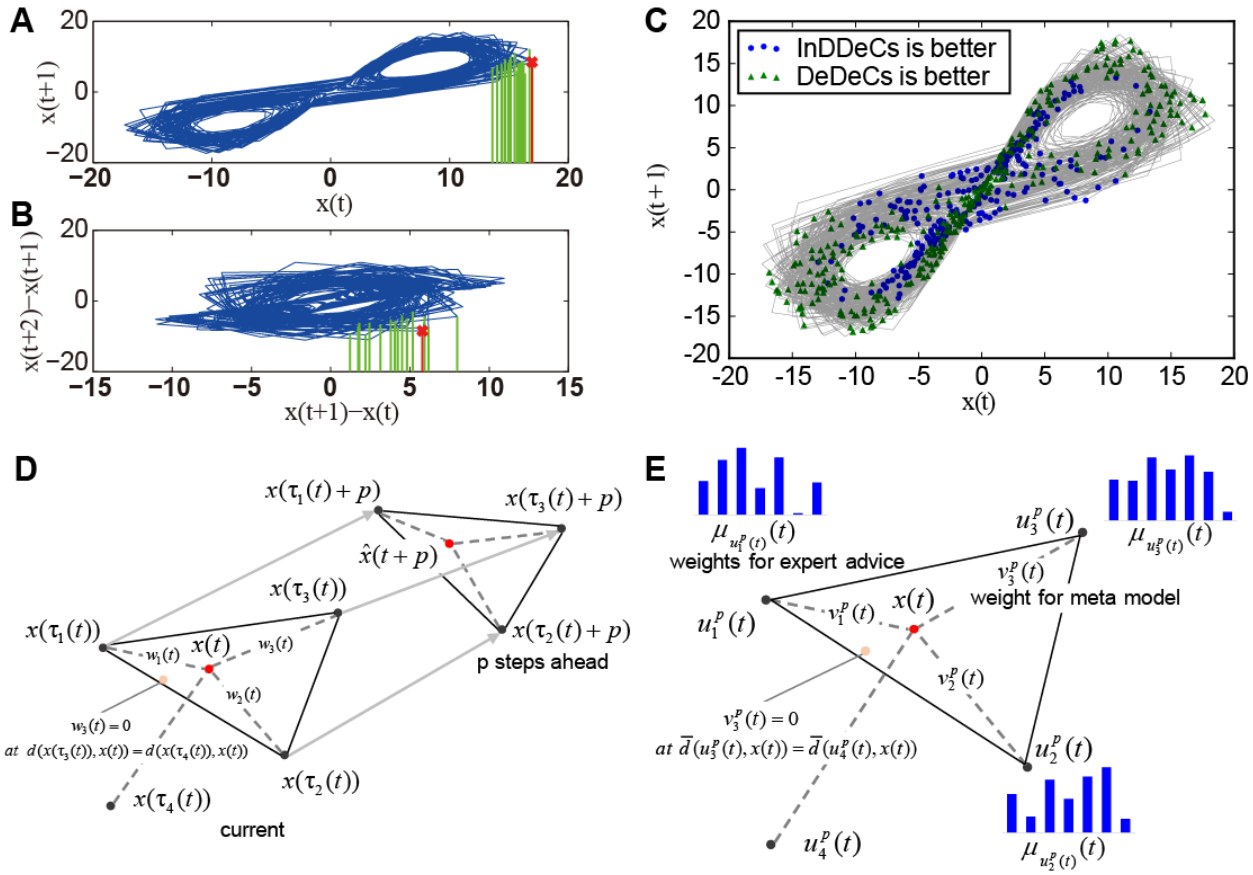

Fig. 1. Proposed methods. (A) Query of the global maximum for $x(t)$ (red line) and its neighboring points in normal delay coordinates (green lines). The query is outside of the interval specified by its neighboring points. (B) Query of the global maximum for $x(t)$ (red line) and its neighboring points in derivative delay coordinates (green lines), where the same query is within the interval specified by its neighboring points. (C) Lorenz model example, showing which coordinates are better for short-term prediction, InDDeCs or DeDeCs, depending on the position in state space. When $x(t)$ is large or small, DeDeCs are better than InDDeCs, while in the middle of the attractor, InDDeCs are superior to DeDeCs. (D) LoCoMo. (E) StaDEA.

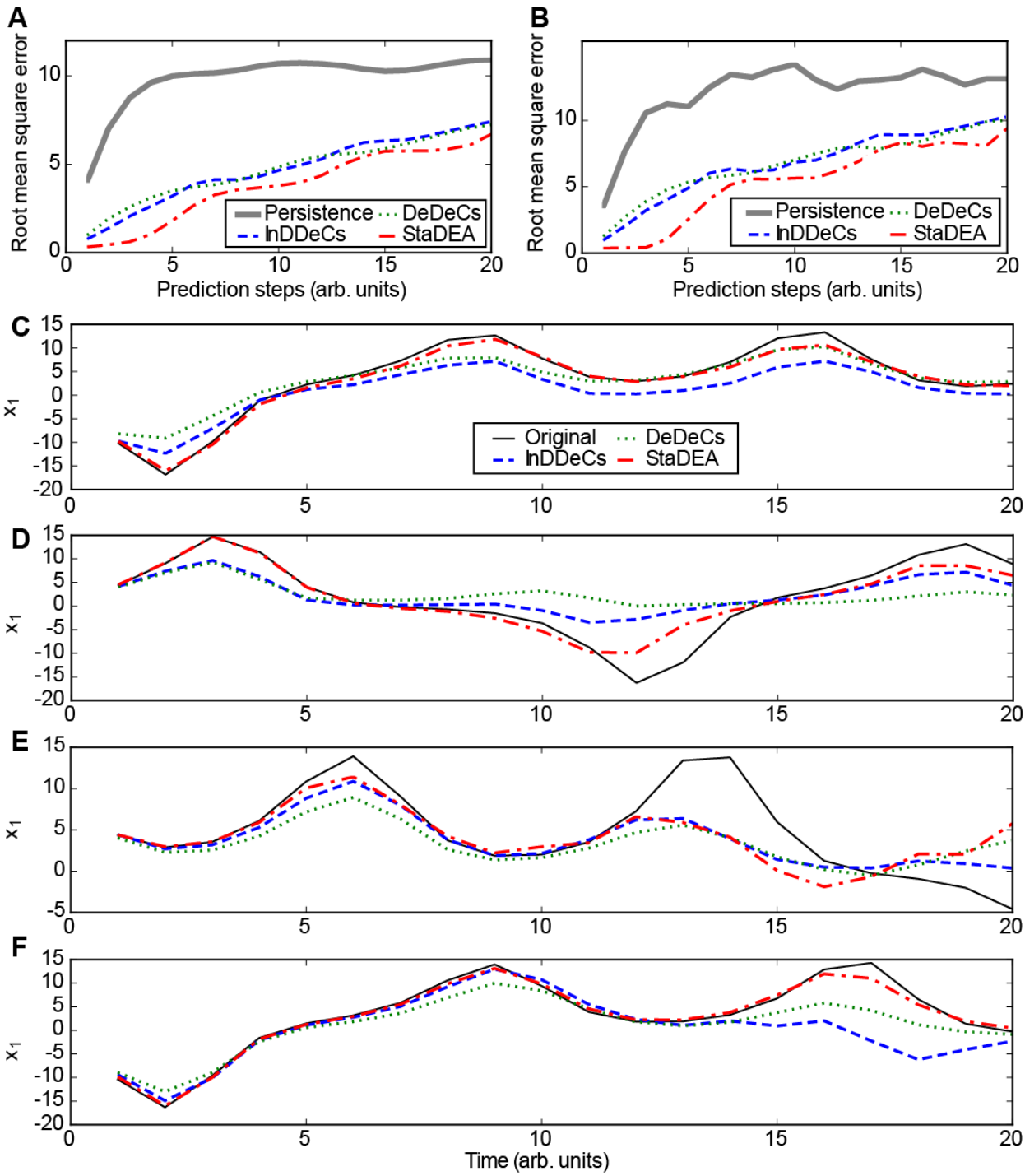

Fig. 2. Prediction performance on the Lorenz model. (A) Root mean square errors for the expert advice over InDDeCs, expert advice over DeDeCs, and the proposed StaDEA, given one prediction step. The root mean square error for the persistence prediction, where the current value is the prediction for future values, is also shown. (B) Root mean square errors of local minima and maxima. (C)-(F) Four examples of these predictions, showing a part of the time series and the corresponding part of the free running predictions from the zeroth step.

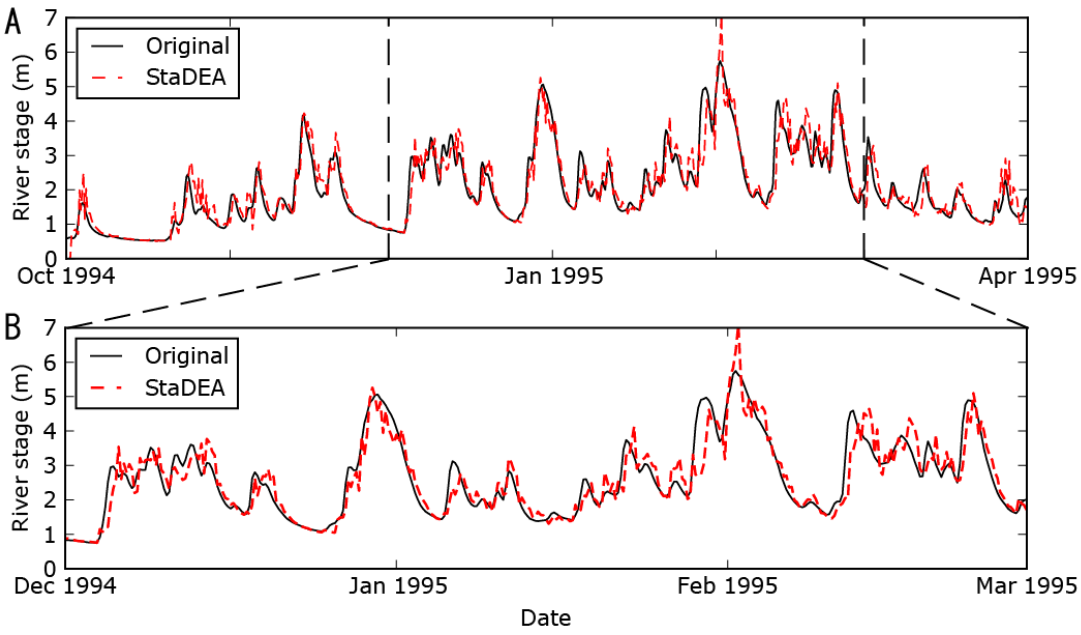

Fig. 3 Time series prediction for water height. (A) 24-hours-ahead prediction (red dash-dotted line) and actual values (black solid line). (B) Detail of (A). Here we used $\theta = 0.5$.

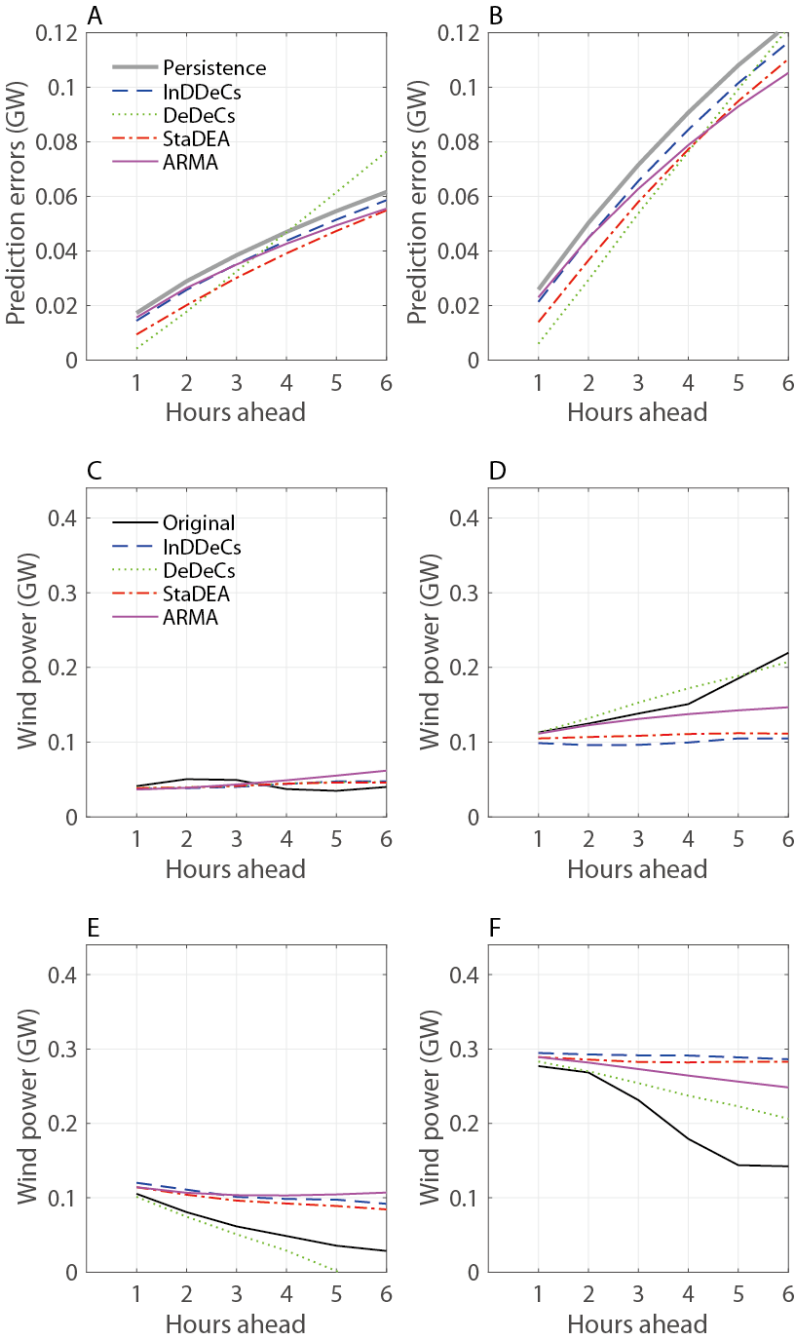

Fig. 4. Time series prediction for wind power. (A)-(B) Root mean square errors for expert advice over InDDeCs (blue dashed line), the expert advice over DeDeCs (green dotted line), StaDEA (red dash-dotted line), ARMA(purple solid line), and persistence prediction (black solid line) for whole data (A) and times when the wind power variations were within top 15% (B). (C)-(F) Examples of these predictions by expert advice over InDDeCs (blue dashed line), expert advice over DeDeCs (green dotted line), StaDEA (red dash-dotted line), ARMA (purple solid line) and ground truth (black solid line). (C)-(F) show a part of the time series and corresponding part of the free-running predictions beginning at the zeroth hour.

Supplementary Materials:

Figures S1-S6

Table S1

Supplementary Materials for

Avoiding Underestimates for Time Series Prediction by State-Dependent Local Integration

Shunya Okuno, Tomoya Takeuchi, Shunsuke Horai, Kazuyuki Aihara, and Yoshito Hirata

correspondence to: yoshito@sat.t.u-tokyo.ac.jp

This PDF file includes:

Figs. S1 to S6

Table S1

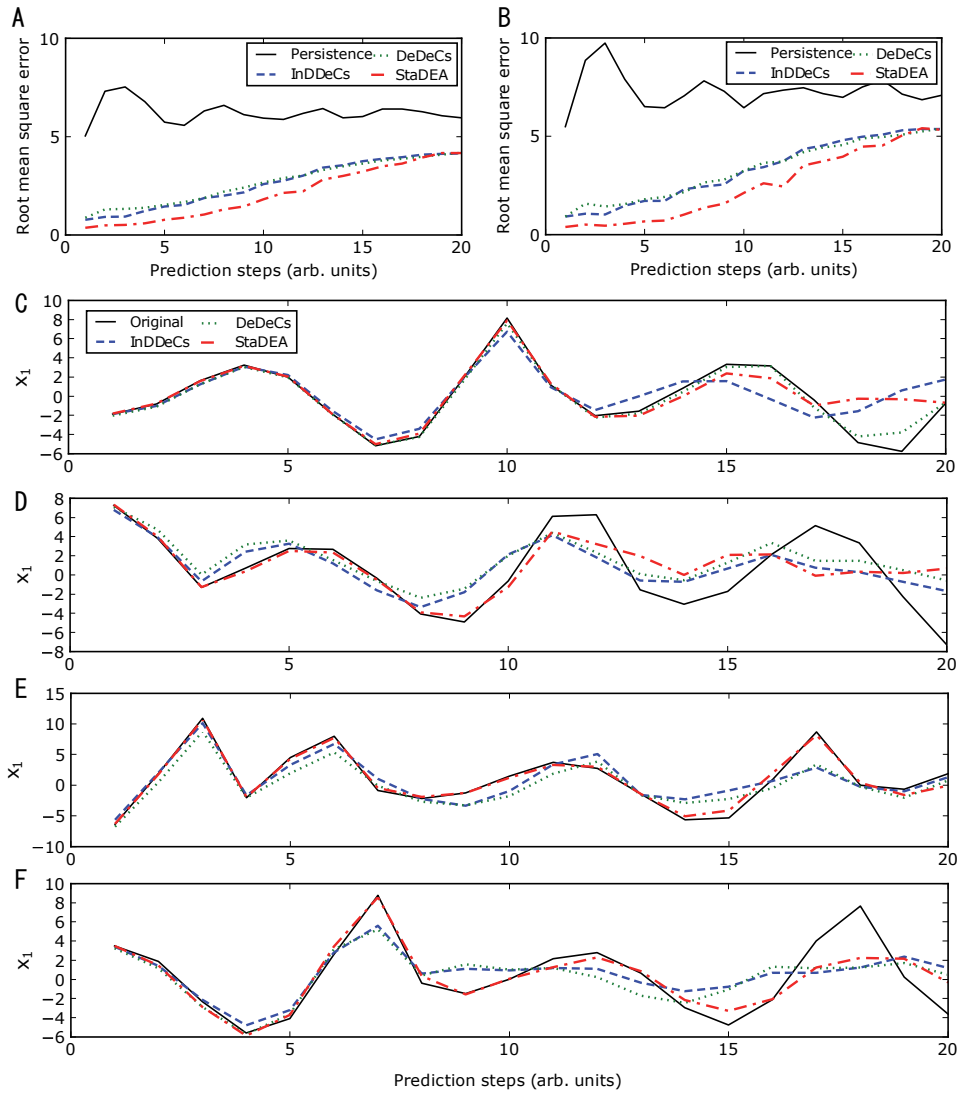

Fig. S1. Prediction performance on the Rössler model. (A) Root mean square errors of InDDeCs, DeDeCs, StaDEA, and persistence prediction. (B) Root mean square errors of local minima and maxima. (C)-(F) Examples of these predictions. See the caption of Fig. 2 to interpret the results.

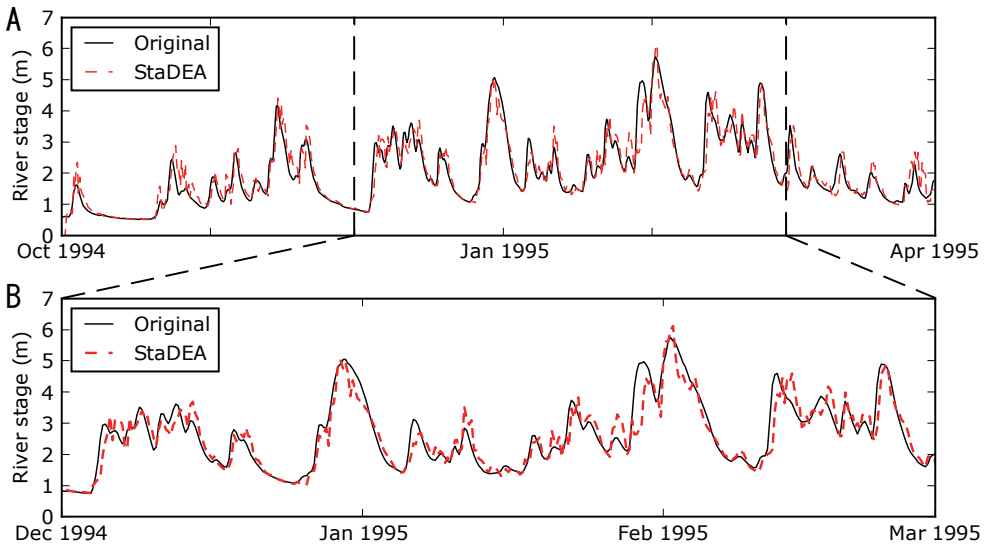

Fig. S2. Time series prediction of water height when we used $\theta = 0.99$.

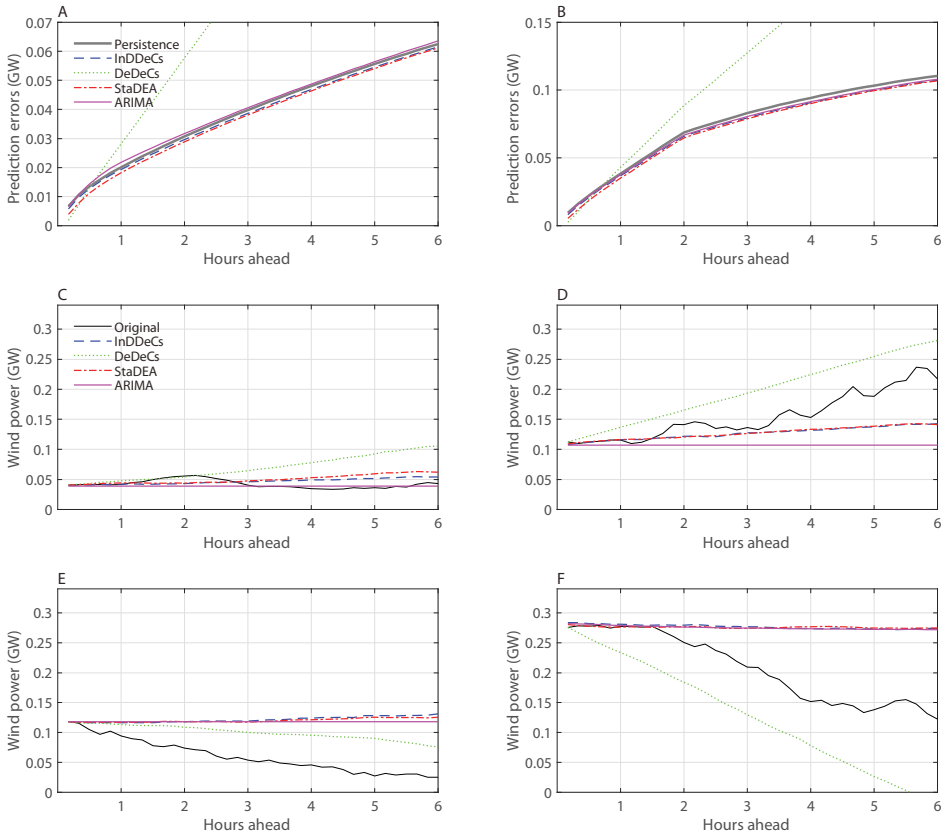

Fig. S3. Wind power predictions in the time resolution of 10 minutes. In this figure, we used the ARIMA model instead of the ARMA model because the time resolution was too fine. See the caption of Fig. 4 to interpret the results.

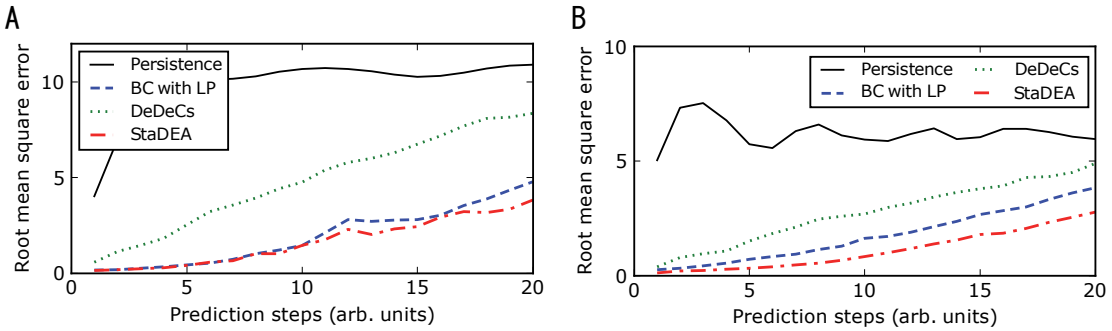

Fig. S4. Performance of the proposed method with barycentric coordinates on the Lorenz and Rössler models. Root mean square errors of barycentric coordinates with linear programming (BC with LP), DeDeCs based on BC with LP, proposed StaDEA based on BC with LP, and persistence prediction on the (A) the Lorenz model and (B) the Rössler model.

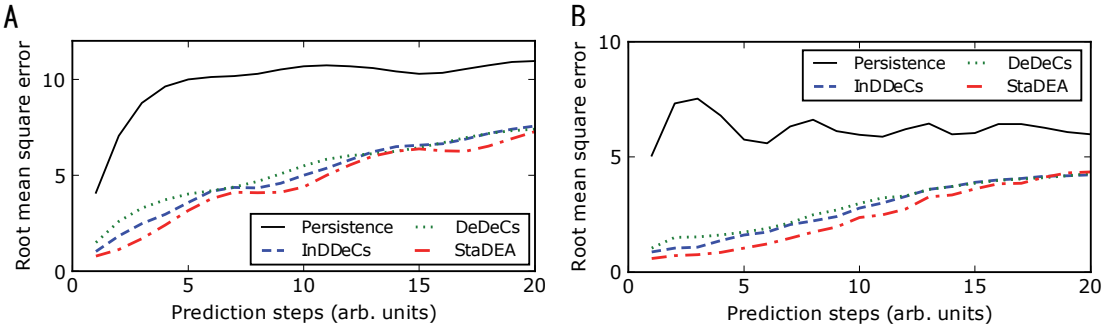

Fig. S5. Prediction performance on the Lorenz and Rössler models with 5% noise. Root mean square errors of InDDeCs, DeDeCs, proposed StaDEA, and persistence prediction on (A) the Lorenz model with 5% noise and (B) the Rössler model with 5% noise.

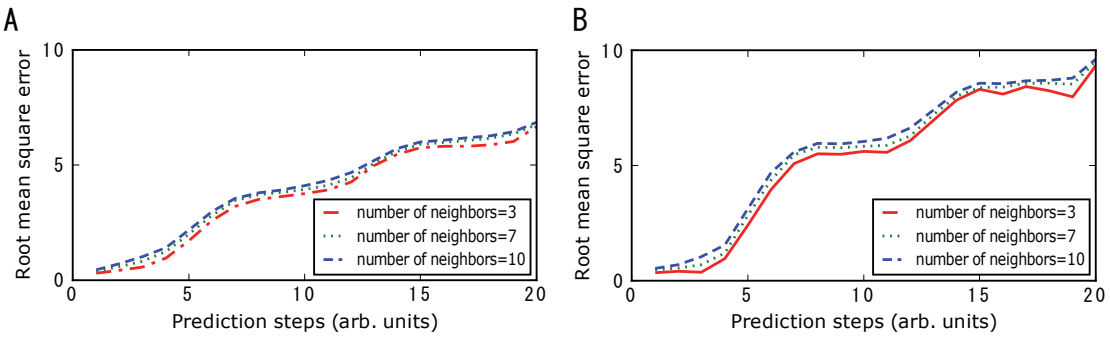

Fig. S6. Prediction errors for the Lorenz model when we change the number of neighbors for StaDEA. Panel A shows the root mean square errors for whole data and panel B shows the root mean square errors when we extract maxima and minima.

Table S1. Prediction errors for the water height 24 h ahead.

Method	Root mean square errors (m)
ARMAX model (first order)	0.4379
ARMAX model (second order)	0.4199
ARMAX model (third order)	0.4390
ARMAX model (fourth order)	0.4435
The proposed method $(\theta = 0.5)$	***0.4014***
The proposed method $(\theta = 0.99)$	***0.3880***