Zhou F, Zhou HM, Yang ZH *et al.* A 2-stage strategy for non-stationary signal prediction and recovery using iterative filtering and neural network. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 34(2): 318–338 Mar. 2019. DOI 10.1007/s11390-019-1913-0

A 2-Stage Strategy for Non-Stationary Signal Prediction and Recovery Using Iterative Filtering and Neural Network

Feng Zhou[1], Hao-Min Zhou[2], Zhi-Hua Yang[1], and Li-Hua Yang[3,4], *Senior Member, IEEE*

[1]*School of Information Science, Guangdong University of Finance and Economics, Guangzhou 510320, China*

[2]*School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, U.S.A.*

[3]*Guangdong Province Key Laboratory of Computational Science, Guangzhou 510275, China*

[4]*School of Mathematics, Sun Yat-sen University, Guangzhou 510275, China*

E-mail: fengzhou@gdufe.edu.cn; hmzhou@math.gatech.edu; zhyang@gdufe.edu.cn; mcsylh@mail.sysu.edu.cn

Received October 14, 2018; revised January 18, 2019.

Abstract Predicting the future information and recovering the missing data for time series are two vital tasks faced in various application fields. They are often subjected to big challenges, especially when the signal is nonlinear and non-stationary which is common in practice. In this paper, we propose a hybrid 2-stage approach, named IF2FNN, to predict (including short-term and long-term predictions) and recover the general types of time series. In the first stage, we decompose the original non-stationary series into several "quasi stationary" intrinsic mode functions (IMFs) by the iterative filtering (IF) method. In the second stage, all of the IMFs are fed as the inputs to the factorization machine based neural network model to perform the prediction and recovery. We test the strategy on five datasets including an artificial constructed signal (ACS), and four real-world signals: the length of day (LOD), the northern hemisphere land-ocean temperature index (NHLTI), the troposphere monthly mean temperature (TMMT), and the national association of securities dealers automated quotations index (NASDAQ). The results are compared with those obtained from the other prevailing methods. Our experiments indicate that under the same conditions, the proposed method outperforms the others for prediction and recovery according to various metrics such as mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE).

Keywords iterative filtering, factorization machine, neural network, time series, data recovery

1 Introduction

Signal prediction and recovery are two common issues in various fields, such as wind power forecasting[1], financial stock market prediction[2,3], water quality prediction[4] and so on[5]. In these applications, the signals often exhibit dynamical, non-linear or even chaotic properties, which make the signal forecasting and recovery face significant challenges.

Most of the existing methods for forecasting time series can be classified into three categories: the ones based on statistical techniques, those using machine learning techniques, and the ones using hybrid strategies. In the category of statistical approaches, there are autoregressive integrated moving average (ARIMA), generalized autoregressive conditional heteroskedasticity (GARCH) volatility[6], the smooth transition autoregressive model (STAR)[7], and Holt-Winters exponential smoothing (HW)[8], just to name a few. These approaches are primarily based on the assumptions of stationarity in time series and linearity among normally distributed variables.

In recent years, machine learning models with-

Regular Paper

Special Section of NSFC Joint Research Fund for Overseas Chinese Scholars and Scholars in Hong Kong and Macao 2014–2017

This work was partially supported by the National Natural Science Foundation of China under Grant Nos. 11771458, 431015 and 61628203, the National Science Foundation of US under Grant Nos. DMS-1620345 and DMS-1830225, the Office of Naval Research (ONR) Award of US under Grant No. N00014-18-1-2852, the Guangdong Youth Innovation Talent Project (Natural Sciences) under Grant No. 2017KQNCX083, the Guangdong Philosophy and Social Science Project of China under Grant No. GD15CGL11, and the Guangzhou Science and Technology Project of China under Grant No. 201707010495.

out these restrictive assumptions have been proposed, and they can outperform the statistical methods[9−12]. Thus, machine learning approaches, such as support vector machine (SVM)[13,14], gradient boosted decision trees (GBDT)[15,16], genetic algorithm[17], fuzzy system[18,19], and neural network (NN)[20−22], have been widely employed in forecasting time series. Although the machine learning based models have achieved remarkable results, there are still limitations. For example, for the neural network models or deep network models, the non-linearities are mainly handled by the activation functions, and there are few techniques addressing the non-linear interactions among the inputs.

Another emerging trend is the hybrid methods which apply wavelet analysis, empirical mode decomposition (EMD) or other signal decomposition techniques to the signal before it is fed to the machine learning models[23−28]. The signal decomposition techniques produce components that are "quasi stationary" looking, and therefore can produce better prediction results. Many of the existing hybrid methods treat each component as an independent signal which is fed into a machine learning model that is also trained independently. The inter-connections among the components are not considered. In [29], a hybrid method strategy called EMD2FNN was proposed, and it exploits the interactions among the components.

Although many of the signal prediction methods can be adopted to recover missing and damage signals during transmission, communication and storage, the traditional strategies for signal recovery are led by statistical synthesis[30−32], or spline interpolations[33,34]. Compared with the prediction problems, the task of signal recovery must match with the existing signal on both ends of an interval, while the prediction only has one end to rely on. This gives the recovery some advantages because the other end of the signal provides more information to use. On the other hand, it poses additional challenges to fulfill requirements that do not exist in the prediction case. For this reason, many of the existing hybrid strategies cannot be extended to the signal recovery.

The study in this paper is based on the EMD2FNN technique proposed in [29]. We aim at extending it to be applicable to not only the financial datasets but also other types of real-world signals, and extending it to cope with not only the one-step forecasting task but also the multi-step one. For these purposes, we adopt iterative filtering (IF)[35−37], instead of EMD[38], due to the simplicity in implementing IF and its robustness in handling different types of signals. We name the new approach IF2FNN, because a factorization machine based neural network (FNN) is used in conjunction with IF. In addition, we design IF2FNN so that it can be used not only for signal prediction, but also for signal recovery.

To illustrate the performance of our proposed method, we test its prediction capability on four time series including one artificial constructed signal (ACS), three real datasets, the length of day (LOD), the troposphere monthly mean temperature (TMMT), and the northern hemisphere land-ocean temperature index (NHLTI), where the data ACS, LOD and NHLTI are used to test the one-step forecasting task, and TMMT is applied to the multi-step forecasting task. Furthermore, to evaluate the ability of data recovery, the data TMMT, NHLTI and the national association of securities dealers automated quotations index (NASDAQ) are used in our experiments. We simulate the damaged data in a variety of ways, such as removing points in a given interval, removing points located in two randomly selected intervals, and removing randomly selected points in the series. Our results are compared with the results obtained by several existing prevailing approaches, such as the SVM, GBDT, NN and the ones combining IF to some machine learning models (for convenience, we denote them as ML). We use two ways to combine IF to ML. One is feeding each component from IF as an independent signal into an ML and training each model separately. The other is inputting all of the components from IF into one ML predictor model. For convenience, we denote the former one as IF2ML-SP, where SP is the abbreviation of segregation prediction, and IF2ML for the later choice. In this paper, the SVM, GBDT, NN models, the Holt-Winters model[39,40] and the smooth extrapolation approach that can be viewed as a single-layer perception are treated as ML for our comparison. The performances of all the methods are measured by mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE).

The rest of this paper is organized into the following sections. We review the necessary ingredients including IF and FNN in Section 2. Our proposed forecasting and data recovery techniques based on IF2FNN are presented in Section 3. We show the simulation results in Section 4. The paper concludes with a discussion in Section 5.

2 Review on IF and FNN

2.1 Iterative Filtering

As an alternative to the traditional methods, such as Fourier or wavelet transforms, EMD has been proven, by numerous studies, to be effective in analyzing non-stationary time series in recent years. It has received considerable attention[35,41,42] and been applied to many disciplines such as ocean science, biomedicine, speech signal processing, image processing, pattern recognition, and financial forecasting[43−45].

Many different algorithms have been proposed to improve EMD. Examples include the strategies based on moving average[46], partial differential equation (PDE)[47,48], operators[49,50], filtering[35−37], and optimizations[42,51]. We adopt the iterative filtering (IF)①[35−37] method in this paper for two reasons. First, it is simple, intuitive and efficient in computation. Second, it is robust for complicated non-stationary signals. In the IF algorithm, each IMF (intrinsic mode function) is produced by convolving iteratively the signal with a low pass filter $w(t)$, $t \in \mathbb{R}$, for example, a Fokker-Planck filter which has the nice property of being compactly supported and smooth on its entire domain. Algorithm 1 describes the detail steps.

Algorithm 1 . Iterative Filtering

Require: given a signal $x(t)$, $IMF = \{\}$
while the number of extrema of $x \geqslant 2$ **do**
1) $x_1 = x$
while the stopping criterion is not satisfied **do**
2) compute the filter length l_n for x_n
3) $x_{n+1}(t) = x_n(t) - \int_{-l_n}^{l_n} x_n(t+y)w(y)\mathrm{d}y$
4) $n = n + 1$
end while
5) $IMF = IMF \cup \{x_n\}$
6) $x = x - x_n$
end while
7) $IMF = IMF \cup \{x\}$

2.2 FNN Model

In recent few years, neural networks (NNs) have achieved tremendous success in many fields including image processing[52,53], speech recognition[54], computer vision[55], and natural language processing[56]. Although NNs exhibit great advantages in learning patterns from large training data, the use of NNs on small datasets, such as a few time series, has received less attention, and it is unclear about how to employ NNs for effectively learning nonlinear interactive features.

Factorizationm machines (FMs), originally introduced for collaborative recommendations[57], are a popular way to capture feature interactions. Similar to SVMs, FMs form a general class of predictors that are able to estimate reliable parameters under the very high sparsity assumption, and they have strong abilities to learn the nonlinear interactive features. Motivated by FM, we proposed an improved NN model by incorporating FM ideas and named it FNN for simplicity in [29]. Compared with the NN model, FNN has two properties. 1) It is able to capture the role of factorized interactions between features, which is an advantage that most of the existing generalized linear models do not have. 2) FNN has the same level (linear) of computation complexity as that of the NN model. The architecture of FNN is depicted in Fig.1.

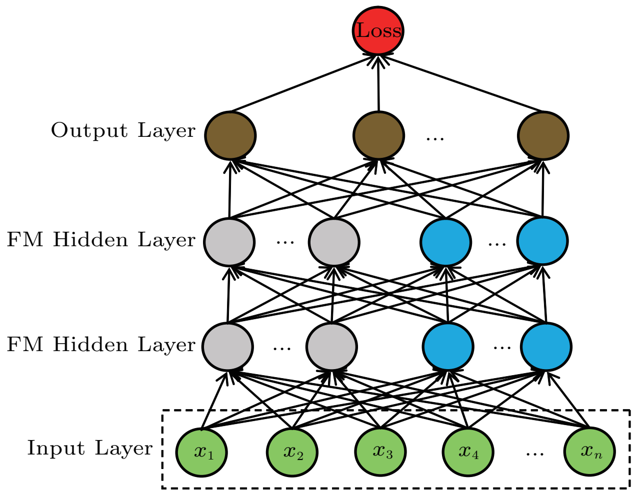


Fig.1. Architecture of FNN.

With a down-top description, the elements of $\boldsymbol{y}^I \in \mathbb{R}^{m_1}$ are the activation outputs of the first FM hidden layer calculated as

$$y_j^I = f(\sum_{i=1}^{n} w_{i,j}^I x_i),\ j = 1, 2, \ldots, m_1 - k_1,$$

$$y_{m_1-k_1+j}^I = f(\frac{1}{2}((\sum_{i=1}^{n} v_{i,j}^I\ x_i)^2 - \sum_{i=1}^{n}(v_{i,j}^I)^2\ x_i^2)),$$
$$j = 1, 2, \ldots, k_1,$$

where $\boldsymbol{x} = \{x_i\}_{i=1}^n$ is the input feature, $w_{i,j}^I$ is the undetermined linear weight, $\boldsymbol{v}_i^I \in \mathbb{R}^{k_1}$ is the undetermined latent weight corresponding to the feature x_i, k_1 is the user-specified dimension of $\{\boldsymbol{v}_i^I\}_{i=1}^n$, and f is an activation function that can be taken as Sigmoid[58], tanh, ReLU[59], PReLU[60], ELU[61] and so on. Similarly, the

①The code of IF was released in website: https://github.com/Acicone/Iterative-Filtering-IF, Jan. 2019.

outputs of the second FM hidden layer can be computed as

$$y_j^H = f(\sum_{i=1}^{m_1} w_{i,j}^H y_i^I),\ j = 1, 2, \ldots, m_2 - k_2,$$

$$y_{m_2-k_2+j}^H = f(\frac{1}{2}((\sum_{i=1}^{m_1} v_{i,j}^H\ y_i^I)^2 - \sum_{i=1}^{m_1}(v_{i,j}^H)^2\ (y_i^I)^2)),$$
$$j = 1, 2, \ldots, k_2,$$

where $w_{i,j}^H$ is the undetermined linear weight, $\boldsymbol{v}_i^H \in \mathbb{R}^{k_2}$ is the undetermined latent weight corresponding to y_i^I, and k_2 is the user-specified dimension of $\{\boldsymbol{v}_i^H\}_{i=1}^{k_1}$. The neurons in the output layer are obtained from

$$y_j^O = f(\sum_{i=1}^{m_2} w_{i,j}^O y_i^H),\ j = 1, 2, \ldots, o,$$

where $w_{i,j}^O$ is the undetermined connection weight. At last, it is the loss layer, where one can choose different loss functions to measure the error for different tasks. Such as, a commonly used loss function is the squared loss defined by $\ell(y_i^O, y_i) = \frac{1}{2}(y_i^O - y_i)^2$ for regression task, and the log-loss function, i.e., $\ell(y_i^O, y_i) = -y_i \log y_i^O - (1 - y_i)\log(1 - y_i^O)$, is usually taken for classification. Furthermore, in order to alleviate the influence of over-fitting, we take the L_2 regularized term into this layer. Thus, the expression of the loss layer is computed as follows:

$$L(\boldsymbol{y}^O, \boldsymbol{y}) = \sum_{i=1}^{o} \ell(y_i^O, y_i) + \frac{\alpha}{2}(\sum_{i=1}^{n}\sum_{j=1}^{m_1-k_1}(w_{i,j}^I)^2 + \sum_{i=1}^{m_1}\sum_{j=1}^{m_2-k_2}(w_{i,j}^H)^2 + \sum_{i=1}^{m_2}\sum_{j=1}^{o}(w_{i,j}^O)^2 + \sum_{i=1}^{n}\sum_{j=1}^{k_1}(v_{i,j}^I)^2 + \sum_{i=1}^{m_1}\sum_{j=1}^{k_2}(v_{i,j}^H)^2),$$

where $\alpha > 0$ denotes the regularized parameter, and $\boldsymbol{y} = \{y_i\}_{i=1}^{o}$ is the true observation.

To optimize the FNN model, the back propagation process is used to iteratively calculate the gradient in each layer, and then the stochastic gradient descent (SGD) is adopted to update the weights until convergence.

3 Proposed Approaches

In this section, an improved framework of the traditional signal decomposition based approach, i.e., IF2FNN, is presented to predict and recover the non-stationary signals. The improved one first decomposes the original non-stationary time series into several "quasi stationary" IMFs by the IF method. And then, all IMFs are fed as the inputs into the FNN predictor model. Next, we describe the details about using IF2FNN in forecasting and data recovery.

3.1 IF2FNN for Non-Stationary Signal Prediction

The process of IF2FNN is similar to that of EMD2FNN proposed in [29]. The differences between them are: 1) IF2FNN replaces the signal decomposition algorithm EMD by IF, which was demonstrated to be more robust in coping with non-stationary signals; 2) only the one-step forecasting task in financial field has been discussed in [29]. In this paper, both one-step and multi-step forecasting tasks with the IF2FNN technique will be discussed. Furthermore, the forecasting task is no longer limited to the financial data, which is also extended to the general applications. Algorithm 2 lists the steps of IF2FNN for prediction.

Algorithm 2 . IF2FNN for Prediction

Require: $\boldsymbol{x} \in \mathbb{R}^n$ is the pre-handle non-stationary time series; M represents the temporal window size; o denotes the size of the predicted values, where $o = 1$ for one-step task, and $o > 1$ for multi-step task

Step 1. Decompose the data $\boldsymbol{x}$ into a set of IMFs $\{\boldsymbol{c}_k\}_{k=1}^{L}$ by using the IF method, where $\boldsymbol{c}_k \in \mathbb{R}^n$ represents the k-th IMF.

Step 2. Construct samples from all IMFs $\{\boldsymbol{c}_k\}_{k=1}^{L}$; we take the set constituted of M consecutive values from each IMF as the input features, the set constituted of the next o values of x is the corresponding label, i.e., $\{c_{k,i}, \ldots, c_{k,i+M-1}\}_{k=1}^{L}$ forms the features of the i-th sample, and $\{x_{i+M}, \ldots, x_{i+M+o-1}\}$ is its label, where $i = 1, 2, \ldots, n - M - o + 1$.

Step 3. Divide the samples into training, validation and testing sets according to a specific proportion.

Step 4. Learn the undetermined FNN predictor from the training dataset, and use the validation set to select its hyper-parameters.

Step 5. Carry out the predicting process on the testing set, i.e., $\hat{\boldsymbol{y}} = FNN[S_t]$, where S_t denotes the input features of the testing dataset, and $\hat{\boldsymbol{y}} \in \mathbb{R}^o$ stands for the predictions.

3.2 IF2FNN for Data Recovery

Another novelty of this paper is to apply IF2FNN to deal with the task of data recovery. Since the missing data must match the given data on both sides of the interval, it complicates the issue of data recovery. The challenges mainly are as follows. 1) How to effectively model the interactions of the information from both sides to better recover the missing data? 2) When the numbers of IMFs on both sides of the missing interval are not consistent, how to construct the samples from both sides?

To overcome the challenges, we present a method, graphically illustrated in Fig.2, by using two models for data recovery, where the model $\mathcal{P}_{\mathrm{L}}$ ($\mathcal{P}_{\mathrm{R}}$) is empowered the ability of data recovery completely according to the information of the left (right) side on the missing data area. The details are listed in steps 2–4 of Algorithm 3. Specially, the samples in training $\mathcal{P}_{\mathrm{L}}$ ($\mathcal{P}_{\mathrm{R}}$) are constructed from the IMFs of $\boldsymbol{x\_left}$ ($\boldsymbol{x\_right}$), which are located on the left (right) of the missing data area. Since both the $\mathcal{P}_{\mathrm{L}}$ and the $\mathcal{P}_{\mathrm{R}}$ models are only decided by the data from one side of the missing data interval, it eliminates the influence of the inconsistency of the number of IMFs during the training process.

In the testing phase, i.e., the process of recovering the missing data by using the learned models $\mathcal{P}_{\mathrm{L}}$ and $\mathcal{P}_{\mathrm{R}}$, the input features (i.e., the contextual IMFs' information of the missing data) are located in both sides of the missing data, which is different from the training process. Since the numbers of IMFs of both sides, $\boldsymbol{x\_left}$ and $\boldsymbol{x\_right}$, might not be equal, the directly constructed inputs are not suitable for the $\mathcal{P}_{\mathrm{L}}$ and the $\mathcal{P}_{\mathrm{R}}$ models. We need modify the extracted features

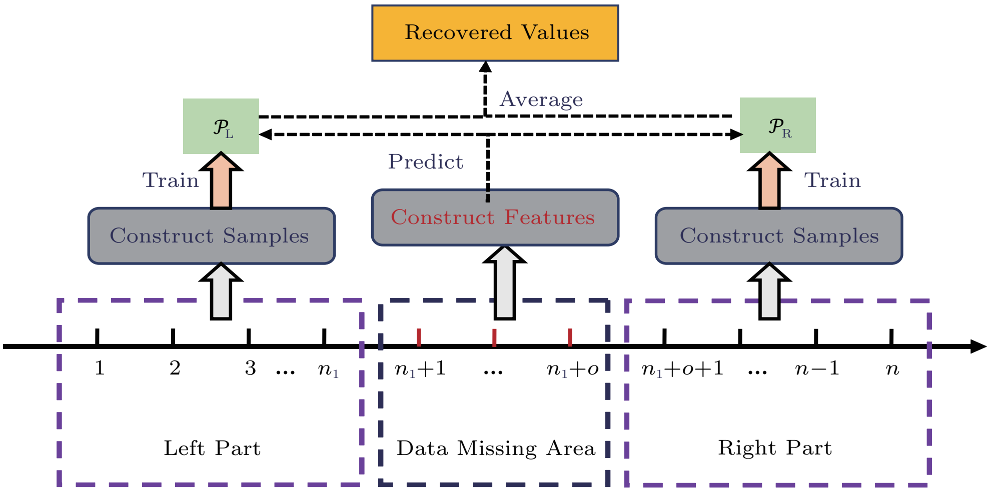


Fig.2. Graphical interpretation of the architecture of our proposed data recovery method.

Algorithm 3 . IF2FNN for Data Recovery

Require: $\boldsymbol{x} \in \mathbb{R}^n$ is the pre-handle non-stationary time series, which contains o continuously missing data points $\{x_{n_1+1}, x_{n_1+2}, \dots, x_{n_1+o}\}$. $\boldsymbol{x\_left} \in \mathbb{R}^{n_1}$ and $\boldsymbol{x\_right} \in \mathbb{R}^{n_2}$ $(n_1 + n_2 = n - o)$ denote the given data from the left and the right parts on the missing interval respectively; M represents the temporal window size on each side.

Step 1. Decompose $\boldsymbol{x\_left}$ and $\boldsymbol{x\_right}$ into IMFs $\{\boldsymbol{c\_left_k}\}_{k=1}^{L_1}$ and $\{\boldsymbol{c\_right_k}\}_{k=1}^{L_2}$ respectively by the IF method, where $\boldsymbol{c\_left_k}$, and $\boldsymbol{c\_right_k}$ represent the k-th IMF of $\boldsymbol{x\_left}$ and $\boldsymbol{x\_right}$ respectively, and L_1 might not equal L_2.

Step 2. Construct samples from the left and the right sides on the missing data interval respectively; for the left part, we take the set constituted of o consecutive values from $\boldsymbol{x\_left}$ as the label; the values, ranged in the closest $2M$ points to the label, are composed of the input features, that is to say, $\{x\_left_{i+1}, \dots, x\_left_{i+o}\}$ constitutes the label set, $\{c\_left_{k,i-M+1}, \dots, c\_left_{k,i}, c\_left_{k,i+o+1}, \dots, c\_left_{k,i+o+M}\}_{k=1}^{L_1}$ are formed as the input features, where $i = M, M+1, n_1-M-o$; similarly, the samples of the right part can be obtained.

Step 3. Divide the left and the right samples into training and validation datasets as a special proportion respectively.

Step 4. Learn the undetermined FNN $\mathcal{P}_{\mathrm{L}}$ and $\mathcal{P}_{\mathrm{R}}$ predictor on the left and the right training datasets respectively, and use the left and the right validation datasets to select the hyper-parameters in $\mathcal{P}_{\mathrm{L}}$ and $\mathcal{P}_{\mathrm{R}}$ respectively.

Step 5. Generate two sets of input features (denoted as S_1, S_2) to recover the missing data, where S_1 is used as the input of $\mathcal{P}_{\mathrm{L}}$, and S_2 is used in $\mathcal{P}_{\mathrm{R}}$. If $L_1 = L_2$, $S_1 = S_2 = \{c\_left_{k,n_1-M+1}, \dots, c\_left_{k,n_1}, c\_right_{k,1}, \dots, c\_right_{k,M}\}_{k=1}^{L_1}$; otherwise (might as well suppose that $L_1 > L_2$), we first find the components where the two sides do not have in common according to comparing their oscillations' patterns (or the oscillations' speed). Then, we supplement the trivial components to the right IMFs (compared with the left, those components originally do not exist) to generate the input features set S_1, and delete the components, which do not match with the right IMFs, from the left IMFs to generate the input features set S_2.

Step 6. Carry out the data recovery on the testing dataset, i.e., $\hat{\boldsymbol{y}} = \boldsymbol{\beta} \bullet \mathcal{P}_{\mathrm{L}}[S_1] + (\boldsymbol{1} - \boldsymbol{\beta}) \bullet \mathcal{P}_{\mathrm{R}}[S_2]$, where the operation $\bullet$ denotes dot product, $\boldsymbol{\beta} \in \mathbb{R}^o$ is the weight vector, and the elements of $\hat{\boldsymbol{y}} \in \mathbb{R}^o$ stand for the recovered values.

by either supplementing or deleting certain components from the left or right IMFs, so that the inputs for $\mathcal{P}_{\mathrm{L}}$ and $\mathcal{P}_{\mathrm{R}}$ are consistent. We describe the details in steps 5 and 6 of Algorithm 3, where we introduce two sets S_1 and S_2 to denote the input features for simplicity, the features in S_1 are fed into the model $\mathcal{P}_{\mathrm{L}}$, and the features in S_2 correspond to the model $\mathcal{P}_{\mathrm{R}}$.

For the parameter vector $\boldsymbol{\beta} = \{\beta_i\}_{i=1}^{o}$ appearing in step 6 in Algorithm 3, there are many different ways to set its value. We just provide a few alternatives for reference as follows.

1) Simple average: the simplest way is to set $\beta_i = 0.5$, for $i = 1, 2, \dots, o$.

2) Length average: the values of the vector $\boldsymbol{\beta}$ depend on the dimensions (lengths) of $\boldsymbol{x\_left}$ and $\boldsymbol{x\_right}$, e.g., $\beta_i = \frac{n_1}{n_1+n_2}$, for $i = 1, 2, \dots, o$.

3) Moving average: the principle of this way is that the recovered data on the left depends more on the model $\mathcal{P}_{\mathrm{L}}$, and the recovered data on the right relies more on the model $\mathcal{P}_{\mathrm{R}}$. Hence, the vector $\boldsymbol{\beta}$ can be set to control the variant influence of the models, $\mathcal{P}_{\mathrm{L}}$, $\mathcal{P}_{\mathrm{R}}$, on $\hat{y}_i$, e.g., $\beta_i = \frac{2o-i+1}{3o+1}$, for $i = 1, 2, \dots, o$.

4 Simulation Results and Evaluations

This section first describes the implementation details including the datasets, experimental setup, and the evaluation criteria. Then, we analyze and evaluate the proposed technique in both forecasting and data recovery tasks.

4.1 Datasets

To test the predicted performance of the proposed method employed to the general non-stationary time series, we compare it against the other prevailing ones in two cases, i.e., one-step and multi-step forecasting. To make the comparison representative, the datasets we use include the artificial constructed signal and the real ones exhibiting different instabilities. In terms of the instabilities from low to high, they are the artificial constructed signal (ACS), $x(t) = 0.6t + \sin(t) + \sin(3t), t \in [0, 35]$, the length of day[②] (LOD), the troposphere monthly mean temperature[③] (TMMT), the northern hemisphere land-ocean temperature index[④] (NHLTI), and the national association of securities dealers automated quotations index[⑤] (NASDAQ), whose information is shown in Table 1. We use the data ACS, LOD and NHLTI in dealing with the task of one-step forecasting, and use TMMT in the task of multi-step forecasting. In our experiments, we divide the data as training, validation and testing datasets as the ratio $7:2:1$ for one-step forecasting task. For multi-step predicting, we take the last 10 points of TMMT, which covers three complete oscillations, as the testing set, and divide the rest data as training and validation datasets as the ratio $8:2$.

Table 1. Data and Their Descriptive Statistical Analysis

Name	Size	Mean	Std.	Data Range
ACS	1 167	10.87	6.11	$[0, 35]$
LOD	1 096	2.21	0.47	1980/1/1−1982/12/31
TMMT	464	0.05	0.25	1978/12−2017/7
NHLTI	1 200	−13.13	26.57	1800/1−1900/12
NASDAQ	1 273	4 148.46	826.91	2012/1/3−2016/12/23

Furthermore, to evaluate the proposed technique in data recovery, the datasets with comparative high instabilities, i.e., TMMT, NHLTI and NASDAQ, are used in our experiments. We simulate the damaged data in three ways. The first is removing the values in the interval located around the center of the series of TMMT and NHLTI. Specifically, the missing data area of TMMT is from the points 228 to 237, covering two complete oscillations, and points 601 through 612 are that for NHLTI, which covers three complete oscillations. Second, it randomly generates two missing data areas on NHLTI, which contain 6 and 12 points respectively. At last, we randomly construct the missing data area just containing single point on NASDAQ, and repeat it five times to make our experiments more objective.

4.2 Experimental Setup

Suppose that the pre-handled data is decomposed into L IMFs via the IF method. For the prediction task, we take the set constituted of M consecutive values from each IMF as the input samples. For data recovery task, the $2M$ points from all IMFs closest to the target time point compose of the input samples set. In the stage of FNN, the activation function f from the hidden layers is taken as tanh. The activation function f in the output layer and the loss function in the loss layer are

②LOD dataset. http://hpiers.obspm.fr/eoppc/eop/eopc04/eopc04.62-now, Sept. 2018.

③TMMT dataset. https://www.nsstc.uah.edu/data/msu/t2lt/uahncdc_lt_5.6.txt, Sept. 2018.

④NHLTI dataset. https://data.giss.nasa.gov/gistemp, Sept. 2018.

⑤NASDAQ dataset. https://finance.yahoo.com, June 2018.

set as the identity function and the squared loss respectively, i.e., $f(x) = x$ and $\ell(y_i^O, y_i) = \frac{1}{2}(y_i^O - y_i)^2$, where y_i is the true value and y_i^O denotes the computation result.

The FNN model will be learned on the training samples with the input size $n = M \times L$ for the prediction and $n = 2M \times L$ for the data recovery task. In this paper, all undetermined weights are assigned with normal distributed random values initially, and then updated according to the back-propagation process on the training samples. For the hyper-parameters m_1, m_2, k_1, and k_2 mentioned in Subsection 2.2 and the parameter M representing the size of dependence on the historical information, to reduce the complexity of FNN model in our experiments, we fix $m_2 = \lfloor \frac{m_1}{2} \rfloor$, $k_1 = m_1 - 1$, $k_2 = m_2 - 1$, $M = 3 \times o$ for the prediction task, and $M = 2 \times o$ for the data recovery. The rest values of hyper-parameter m_1, the learning rate η in back-propagation process and the regularized parameter α are obtained by the grid search algorithm, which aims at finding the optimal parameters from various parameter combinations via minimizing the squared loss on the validation dataset.

4.3 Evaluation Criteria

Three accuracy measures are chosen to evaluate the predicted values $\hat{\boldsymbol{y}} \in \mathbb{R}^{o \times N}$ relative to the actual values $\boldsymbol{y} \in \mathbb{R}^{o \times N}$, where N is the sample size. They are mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE)⑥. Their definitions are given in Table 2. In our experiments, we use all three measures to judge the prediction performance. Smaller values of these indexes indicate more accurate forecast. When the results are not consistent among the indexes, we choose the relatively more stable one, MAPE as suggested by Makridakis[62], to be the main reference criterion.

Table 2. Evaluation Indexes

Measure	Expression
MAE	$\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{o}\lvert\hat{y}_{i,n} - y_{i,n}\rvert$
RMSE	$\sqrt{\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{o}(\hat{y}_{i,n} - y_{i,n})^2}$
MAPE	$\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{o}\lvert\frac{\hat{y}_{i,n} - y_{i,n}}{y_{i,n}}\rvert$

4.4 Discussion of Predicted Accuracy

The experimental data for prediction and their IMFs are shown in Fig.3. Figs.3(a)–3(d) depict the original signals and their IMFs by the IF technique corresponding to the ACS, LOD, TMMT and NHLTI respectively. We can observe that they exhibit different instabilities from low to high. The rest panels are their IMFs extracted from the original signals by IF as the frequencies changing from high to low.

After obtaining the IMFs by the IF method, we first employ our proposed FNN technique to one-step forecasting task on the time series including ACS, LOD and NHLTI. To evaluate the performance, we compare the results with those obtained by the IF2NN techniques, where the IF2FNN and the IF2NN techniques have one point in common that they receive all of the IMFs as their inputs. We also compare the IF2FNN technique against the prevailing models, such as SVM⑦, GBDT⑧, and the other traditional signal decomposition based forecasting models including Holt-Winters⑨ (HW), and neural network (NN). Since these traditional signal decomposition based techniques are achieved through separately predicting for each IMF, they will be referred to as "segregation prediction (SP)" for convenience. Thus, we denote the traditional signal decomposition based Holt-Winters and neural network models as IF2HW-SP and IF2NN-SP respectively.

In addition, a simple and natural model, called smooth extrapolation approach (denoted by EA), is presented to predict for each IMF. We denote it by IF2EA-SP for convenience. Suppose $\boldsymbol{c} \in \mathbb{R}^n$ denotes an IMF, m represents the window size. Similar to the form of moving average, the expression of the $(n+1)$-th value of $\boldsymbol{c}$ by EA is predicted as follows:

$$\hat{c} = \sum_{i=1}^{m} w_i \times c_{n-m+i},$$

where the undetermined vector $\boldsymbol{w} := \{w_i\}_{i=1}^{m}$ is decided by $\boldsymbol{c}$ itself. From the expression, EA essentially can be seen as a single-layer perception. The weights are learned as Algorithm 4. The results obtained from IF2FNN are compared with those from IF2EA-SP as well.

⑥MAPE would be meaningless if any true observation is trivial. Hence, we add the maximum of the true observations to the denominator at the case.

⑦SVM package. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html, Nov. 2018.

⑧GBDT package. https://xgboost.readthedocs.io/en/latest/python/python_intro.html, Nov. 2018.

⑨HW package. https://www.rdocumentation.org/packages/stats/versions/3.5.1/topics/HoltWinters, Nov. 2018.

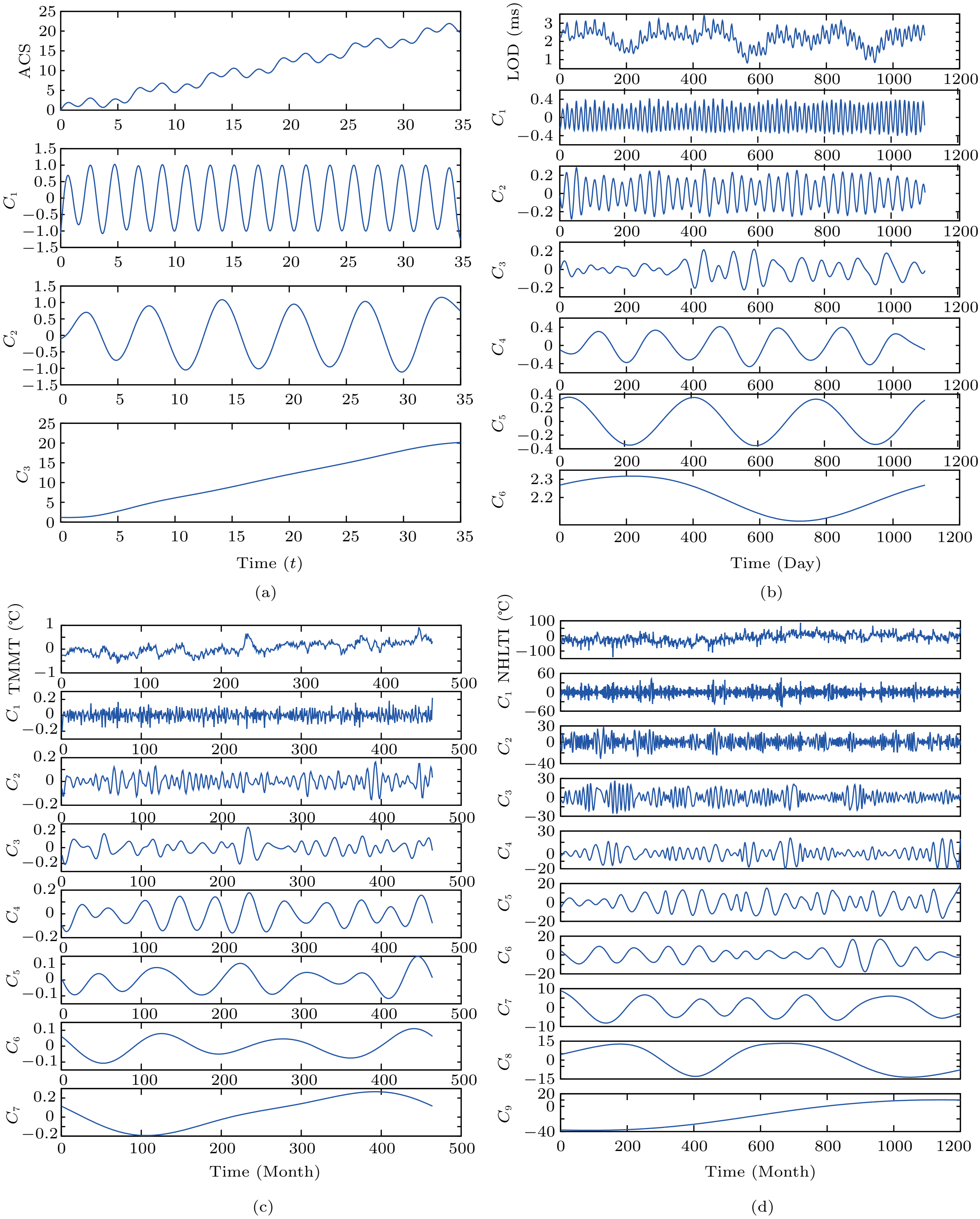


Fig.3. Original signals (top panel) and their IMFs (from the second to the last panels) by the IF approach. (a) ACS. (b) LOD. (c) TMMT. (d) NHLTI.

Algorithm 4 . Smooth Extrapolation Approach (EA)

Step 1. Given the signal $\boldsymbol{c} \in \mathbb{R}^n$, set the window dimension as m.

Step 2. Set $h = n - m$, and then find the undetermined weight vector $\boldsymbol{w}$ by the following optimal model:

$$\boldsymbol{w} = \arg\min_{\boldsymbol{w}} \alpha \|\boldsymbol{w}\|_2 + \|\boldsymbol{A} \cdot \boldsymbol{w} - \boldsymbol{y}\|_2,$$

where $\boldsymbol{A} \in \mathbb{R}^{h \times m}$, whose elements are defined as $A(i, j) = c_{i+j-1}$, and $\boldsymbol{y} \in \mathbb{R}^h$, and $y(i) = c_{i+m}$.

The results of one-step forecasting obtained from all of the predictors on ACS, LOD and NHLTI are depicted in Figs.4–6 respectively, where the top panels depict the predicted values covering the whole testing dataset, and the bottom panels give the details with the errors between the true observations and the results obtained from the predicted models including SVM, GBDT, IF2HW-SP, IF2EA-SP, IF2NN-SP, IF2NN, EMD2FNN and IF2FNN. From the plots, we have the following findings.

- For the predictions on ACS depicted in Fig.4, all techniques except SVM and GBDT are able to grasp the evolutionary trend according to the top panel, where the predictions of GBDT are all the same, fixed to around 13. This is because the overall trend of ACS

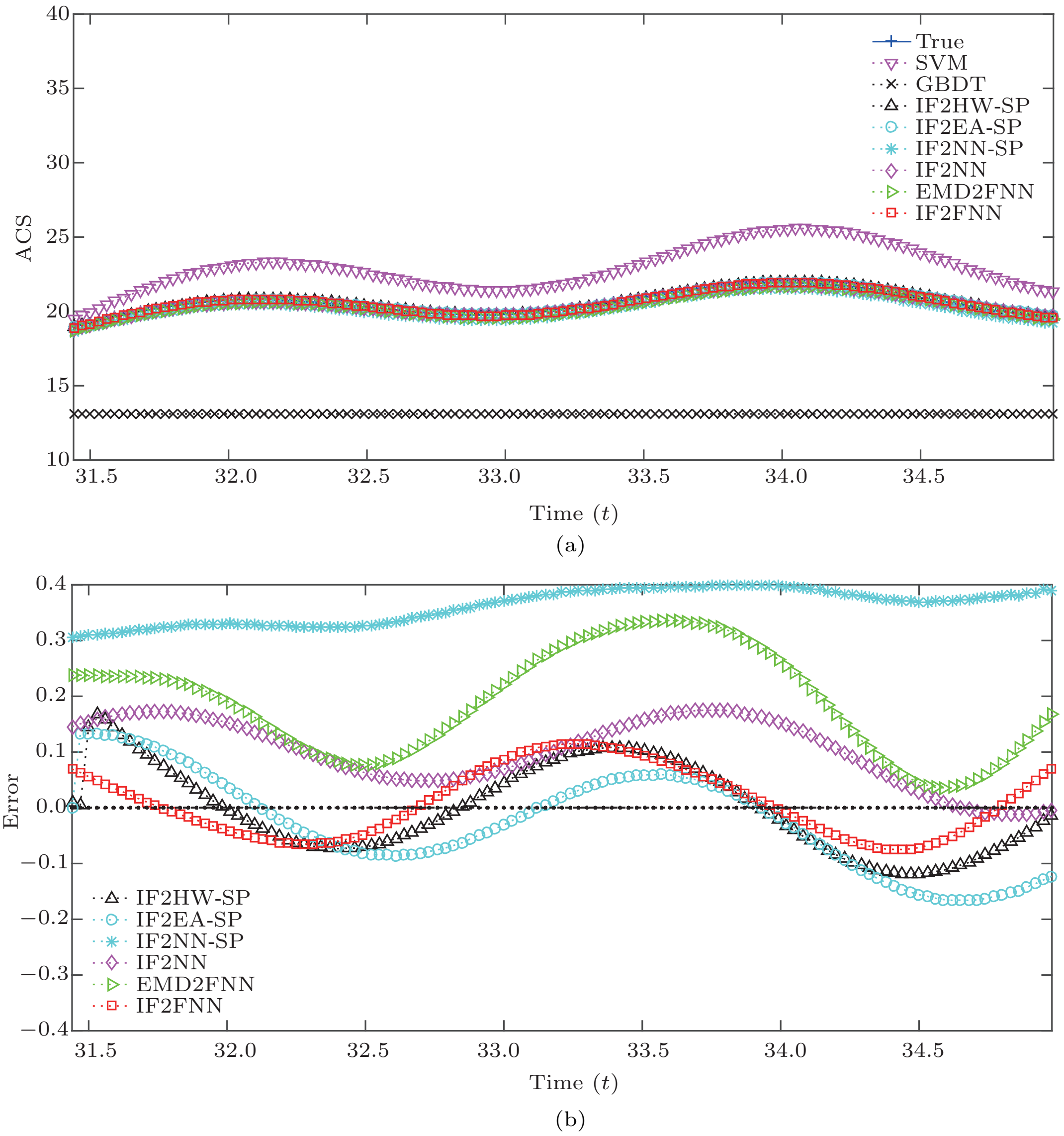


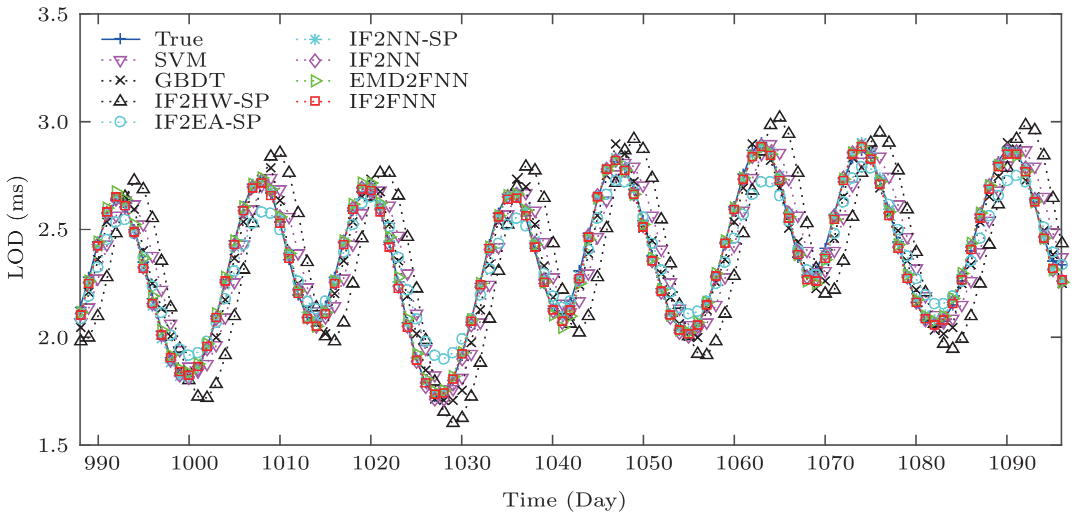
Fig.4. Results and errors of the models in one-step forecasting task on ACS. (a) Predicted values covering the whole testing dataset. (b) Errors between the true observations and the predictions of all models. As it can be seen from (a) that the results of the SVM and GBDT models are clearly poor, we omit their errors in (b).

data goes up, and the values of leaf nodes of each tree in the trained GBDT model cannot exceed this predicted value. From the detailed error panels, the errors located in peaks and troughs are larger than those located in the other places. Visually, we can order the techniques as IF2FNN > IF2EA-SP > IF2HW-SP > IF2NN > EMD2FNN > IF2NN-SP > SVM > GBDT according to the performance.

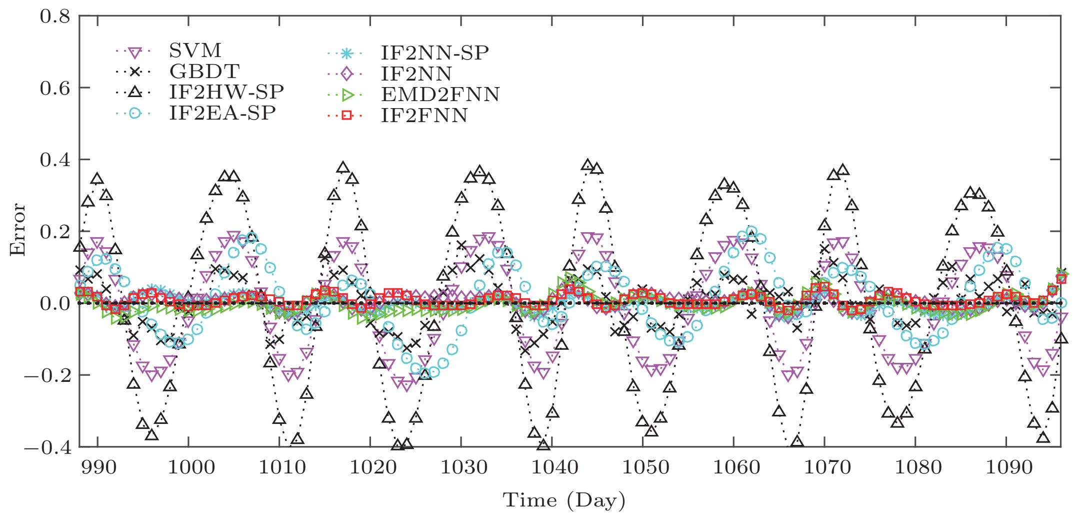
• According to Fig.5(a), the predicted results of IF2HW-SP display obvious hysteresis, and the results obtained from both SVM and IF2EA-SP have comparatively significant discrepancy with the true observations especially on the peaks and troughs. Moreover, the techniques can be visually ranked as {IF2FNN, EMD2FNN, IF2NN, IF2NN-SP} > IF2EA-SP > SVM > IF2HW-SP from the errors depicted in Fig.5(b).

• With regard to the predicted performance on NHLTI shown in Fig.6, the results from IF2HW-SP also exhibit obvious hysteresis according to Fig.6(a). Furthermore, the increased instability of NHLTI compared with ACS and LOD increases the discrepancies between the true observations and results from the predicted approaches. According to Fig.6(b), we can approximately order the techniques by {IF2FNN, IF2NN, IF2NN-SP, IF2EA-SP} > EMD2FNN > SVM > GBDT > IF2HW-SP following their performance.

For the multi-step predicting task on TMMT, the results are depicted in Fig.7, where Fig.7(a) shows

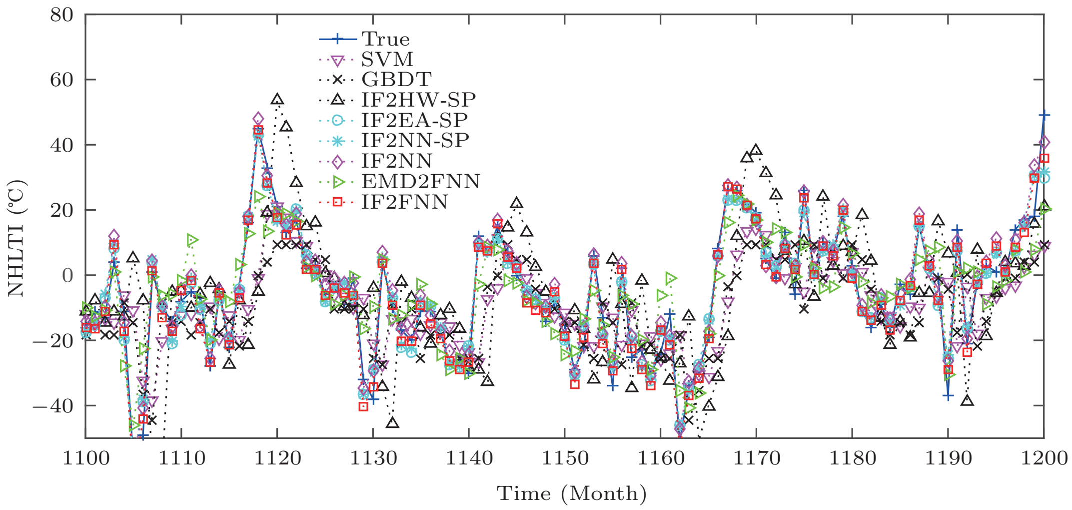


(a)

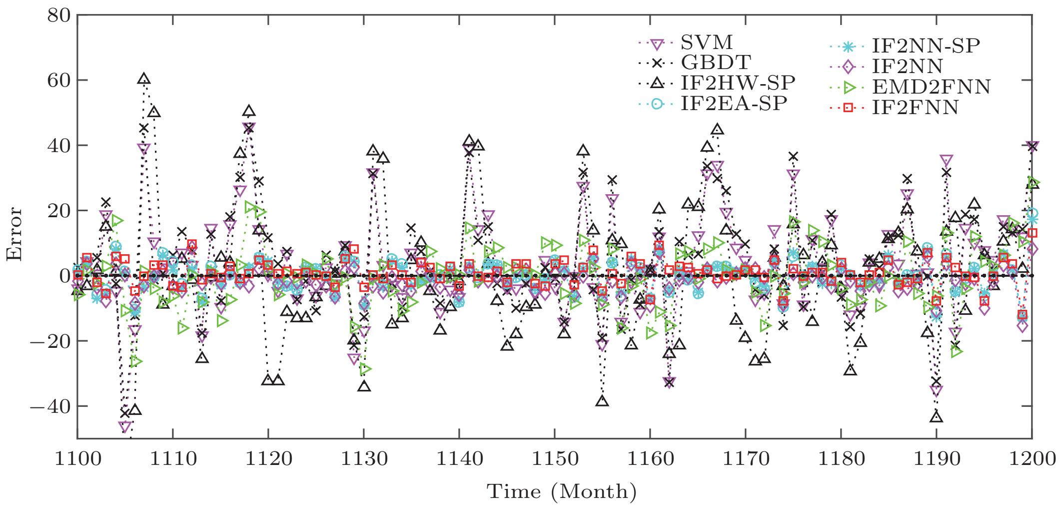


(b)

Fig.5. Results and errors of the models in one-step forecasting task on LOD. (a) Predicted values covering the whole testing dataset. (b) Errors between the true observations and the predictions of all models.



(a)



(b)

Fig.6. Results and errors of the models in one-step forecasting task on NHLTI. (a) Predicted values covering the whole testing dataset. (b) Errors between the true observations and the predictions of all models.

the true data and the predictions obtained from the models including SVM, GBDT, IF2HW-SP, IF2EA-SP, IF2NN-SP, IF2NN, EMD2FNN and IF2FNN. The errors between the predictions and the true observations are depicted in Fig.7(b). According to the figure, we can roughly conclude that {IF2FNN, IF2NN, IF2NN-SP} > SVM > IF2HW-SP > GBDT > EMD2FNN > IF2EA-SP.

In order to know more clearly about the effects from the proposed approach in both one-step and multi-step prediction tasks, we evaluate the models with the indexes listed in Table 2. The values are listed in Table 3–Table 6. From the results, we have the following findings.

• The values listed in Table 3–Table 6 are basically consistent with our conclusions above, i.e., the models can be ordered as IF2FNN > IF2EA-SP > IF2HW-SP > IF2NN > EMD2FNN > IF2NN-SP > SVM > GBDT according to their performances for ACS, by IF2FNN > IF2NN > IF2NN-SP > EMD2FNN > GBDT > IF2EA-SP > SVM > IF2HW-SP for LOD, by IF2FNN > IF2NN > IF2NN-SP > IF2EA-SP > EMD2FNN > SVM > GBDT > IF2HW-SP for NHLTI, and by IF2FNN > IF2NN > IF2NN-SP > SVM > IF2HW-SP > GBDT > EMD2FNN > IF2EA-SP for TMMT.

• According to the metric of MAPE in one-step forecasting task, as the instability of the time series increases, the prediction errors from all techniques except SVM and GBDT also increase. However, the proposed one always performs the best. For the results in Table 6,

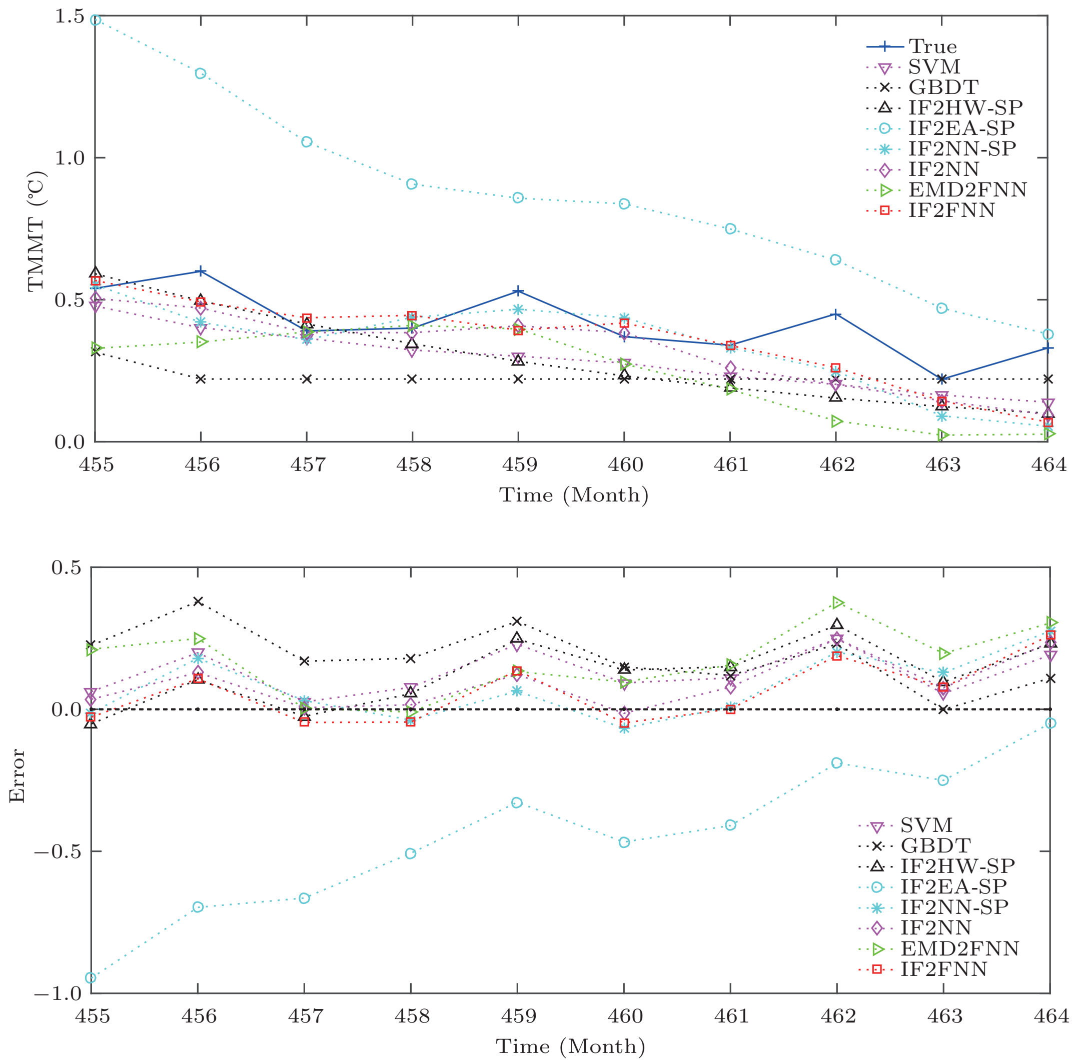


Fig. 7. Results and errors of the models in multi-step forecasting task on TMMT. (a) Predicted values covering the whole testing dataset. (b) Errors between the true observations and the predictions of all models.

IF2FNN is also the best among the models. Hence, the proposed technique performs the best not only in the one-step prediction task, but also in the multi-step one.

Table 3. Performances of Different Models for One-Step Prediction on ACS

Model	MAE	RMSE	MAPE
SVM	2.332 9	2.455 1	0.112 3
GBDT	7.460 0	7.497 9	0.361 9
IF2HW-SP	0.094 3	0.106 0	0.004 6
IF2EA-SP	0.085 4	0.101 7	0.004 2
IF2NN-SP	0.362 5	0.363 8	0.017 6
IF2NN	0.100 6	0.115 6	0.004 9
EMD2FNN	0.188 7	0.210 3	0.009 1
IF2FNN	0.052 7	0.061 2	0.002 6

Table 4. Performances of Different Models for One-Step Prediction on LOD

Model	MAE	RMSE	MAPE
SVM	0.119 4	0.133 2	0.050 9
GBDT	0.058 0	0.068 3	0.025 0
IF2HW-SP	0.240 5	0.267 5	0.102 3
IF2EA-SP	0.092 3	0.108 8	0.039 5
IF2NN-SP	0.016 4	0.020 3	0.007 0
IF2NN	0.015 4	0.018 4	0.006 6
EMD2FNN	0.018 9	0.023 9	0.008 1
IF2FNN	0.014 6	0.018 7	0.006 1

• From the results of the IF-based approaches, i.e., IF2FNN, IF2NN and IF2NN-SP, we have IF2FNN $>$ IF2NN $>$ IF2NN-SP. It reflects that the predictors can be ranked as FNN $>$ NN $>$ NN-SP.

Table 5. Performances of Different Models for One-Step Prediction on NHLTI

Model	MAE	RMSE	MAPE
SVM	12.030 0	16.276 5	0.101 8
GBDT	12.942 8	17.008 7	0.110 2
IF2HW-SP	14.848 2	19.692 5	0.201 3
IF2EA-SP	3.732 6	4.856 9	0.049 1
IF2NN-SP	3.683 3	4.793 1	0.048 6
IF2NN	3.292 2	4.253 3	0.043 9
EMD2FNN	7.188 3	9.543 6	0.081 5
IF2FNN	2.998 1	3.966 8	0.039 6

Table 6. Performances of Different Models for Multi-Step Prediction on TMMT

Model	MAE	RMSE	MAPE
SVM	0.129 0	0.149 7	0.309 3
GBDT	0.186 9	0.212 5	0.411 2
IF2HW-SP	0.139 2	0.164 1	0.354 1
IF2EA-SP	0.450 9	0.517 3	1.068 1
IF2NN-SP	0.100 6	0.132 8	0.269 1
IF2NN	0.095 4	0.126 9	0.242 8
EMD2FNN	0.173 5	0.207 5	0.445 8
IF2FNN	0.093 9	0.121 2	0.241 1

• Comparing the results of IF2FNN with those of EMD2FNN, we can obtain that the decomposition techniques have an important role in forecasting. And the results imply that IF > EMD.

4.5 Discussion of Recovered Accuracy

In this part, we first use the damaged data, which are generated by removing one missing area of TMMT and NHLTI respectively, to illustrate the IF2FNN approach for data recovery. The data on both ends of the missing data area are decomposed into IMFs by the IF method, and the results are shown in Fig.8. From it, we can see that the numbers of IMFs on both the left and the right sides are 7 for TMMT, however, they are 8 and 9 respectively for NHLTI, which are inconsistent. After comparing the oscillatory patterns of the left components with those of the right ones, we find the fifth IMF appearing in Fig.8(d) is a particular one, whose oscillating pattern is quite different from the components in Fig.8(c). Hence, we delete the fifth component from Fig.8(d) before generating the testing data's input features S_1, which is fed into the left model $\mathcal{P}_{\rm L}$, and supplement a trivial component between the fourth and the fifth components of Fig.8(c) before constructing the testing data's features set S_2 that is transmitted into the right model $\mathcal{P}_{\rm R}$.

According to step 6 in Algorithm 3, we provide three ways of selecting the parameter vector $\boldsymbol{\beta}$ for reference, which are called the simple average, length average and moving average approaches respectively in Subsection 3.2. Next, we compare their performances. Table 7 lists the values of the error metrics on the damaged data of TMMT and NHLTI with one missing data area respectively. Since the missing data are taken from the middle of the series TMMT, the values by both the simple average and the length average approaches on TMMT are the same. Moreover, the results obtained from the moving average approach on both TMMT and NHLTI datasets are the best. Hence, we decide to use the moving average method in our experiments below.

To evaluate the performance of our proposed technique in data recovery, we compare it with the models including NN, IF2NN-SP and EMD2FNN. For the NN approach, it is implemented by using two models as well, one is responsible for the left side on the missing data area and the other acts on the right side. The samples are constructed similar to the IF2FNN-based approach, but the difference is that NN does not require the IF method, and thus the input features are directly extracted from the original signal. For the IF2NN-SP approach, it is achieved using two classes of models, where the first class of models is used in successively and separately predicting the future values of each left IMF, and it similarly uses the second class of models to separately predict the historical values of each right IMF. The EMD2FNN approach in data recovery is similar to IF2FNN. Their difference is that the EMD method replaces the IF one.

The results on the damaged data TMMT and NHLTI with one missing data area are shown in Fig.9 and Fig.10 respectively. Moreover, Table 8 and Table 9 list the performance values. From them, we can conclude that the proposed approach IF2FNN is able to significantly improve the recovery accuracy, and it reduces 25.0% MAPE than the second best approach for TMMT, and 31.8% than the second best one for NHLTL.

In addition, we simulate the damaged data by randomly generating two missing data areas on NHLTI, which contain 6 and 12 points respectively. The recovered results and performance are given in Fig.11 and Table 10. The results state that the IF2FNN approach generally performs the best. But the conclusion fails if the context information is too little, e.g., the results of the missing data area $[225, 230]$, the left side on the area just contains 224 points, which makes the recovered results vulnerable by the "endpoint effect" from the signal decomposition approach.

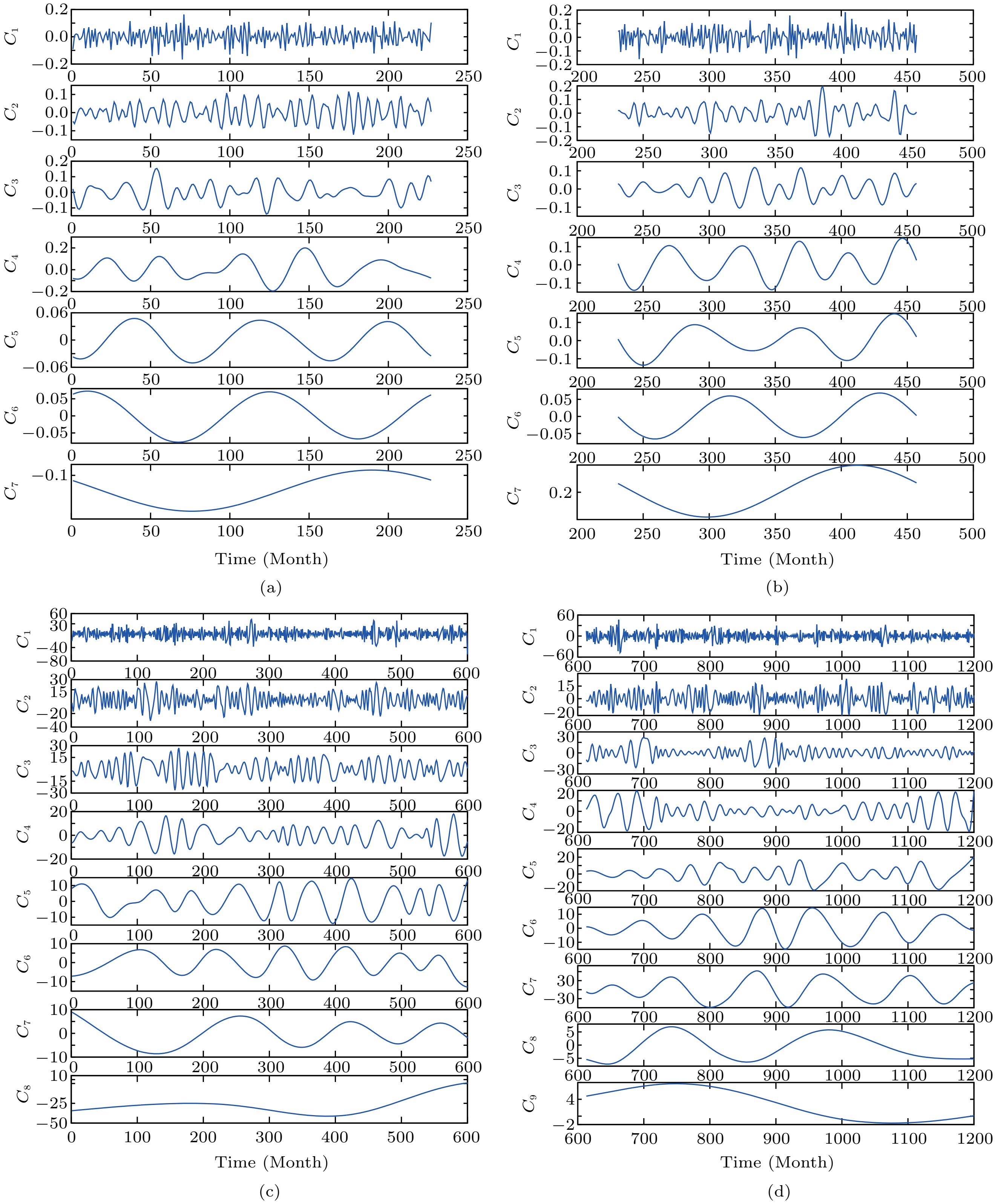


Fig.8. IMFs of the signals, located on both sides of the data missing area, obtained from the IF technique. (a) IMFs of the signal located on the left side of the data missing area (i.e., [228, 237]) of TMMT. (b) IMFs of the signal located on the right side of the data missing area (i.e., [228, 237]) of TMMT. (c) IMFs of the signal located on the left side of the data missing area (i.e., [601, 612]) of NHLTI. (d) IMFs of the signal located on the right side of the data missing area (i.e., [601, 612]) of NHLTI.

Table 7. Performances of Different Ways in Selecting β for Data Recovery

Data	Method	MAE	MSE	MAPE
TMMT	Simple average	0.249 1	0.294 8	0.513 9
	Length average	0.249 1	0.294 8	0.513 9
	Moving average	0.241 2	0.291 2	0.491 3
NHLTI	Simple average	10.797 5	13.473 3	0.788 3
	Length average	10.829 5	13.497 9	0.789 1
	Moving average	10.419 9	10.797 5	0.787 8

At last, we employ the recovery task on NASDAQ, whose IMFs are plotted in Fig.12. We simulate the damaged data by randomly selecting single missing data point from NASDAQ, and repeat the operation five times to make the discussions more objective. The randomly selected missing points and values are $(1\,084, 4\,869.31)$, $(664, 4\,442.69)$, $(221, 2\,904.82)$, $(992, 4\,927.85)$ and $(886, 5\,055.52)$. To evaluate the performance of the IF2FNN method, we compare it with the models including SVM, GBDT, IF2SVM, IF2GBDT, NN, IF2NN-SP, and EMD2FNN. The values of the recovered data and MAPE metric from the models are listed in Table 11 and Fig.13 respectively. The results still reflect that the IF2FNN approach works the best as long as the missing point is not selected too close to the endpoint 1 or 1 273. Otherwise, like the results of the third missing point $(221, 2\,904.82)$, which leads the signal decomposition based (EMD and IF based) techniques are susceptible to interference from the "endpoint effect" of the signal decomposition method.

5 Conclusions

This paper presents a novel technique integrating IF and FM-based neural network together, IF2FNN, to solve several issues about the general types of non-stationary signal, including one-step and multi-step forecasting and data recovery. The IF technique was experimentally proven to be efficient and robust in handling the complicated non-stationary signals, and

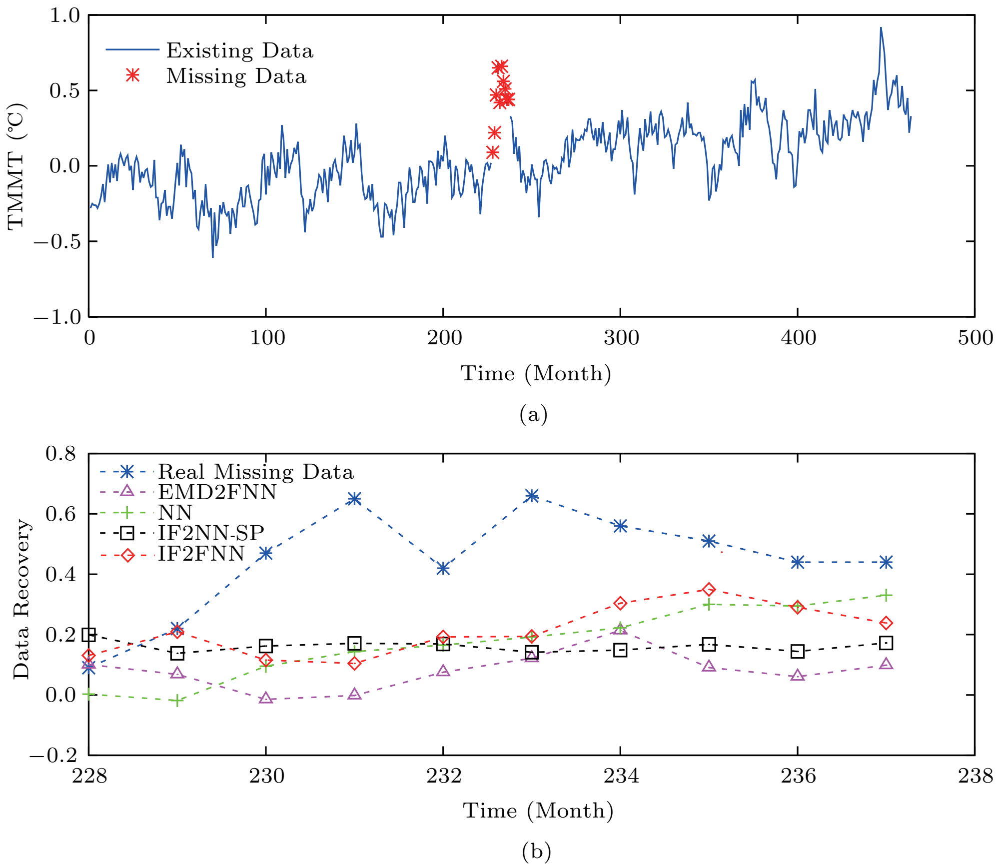
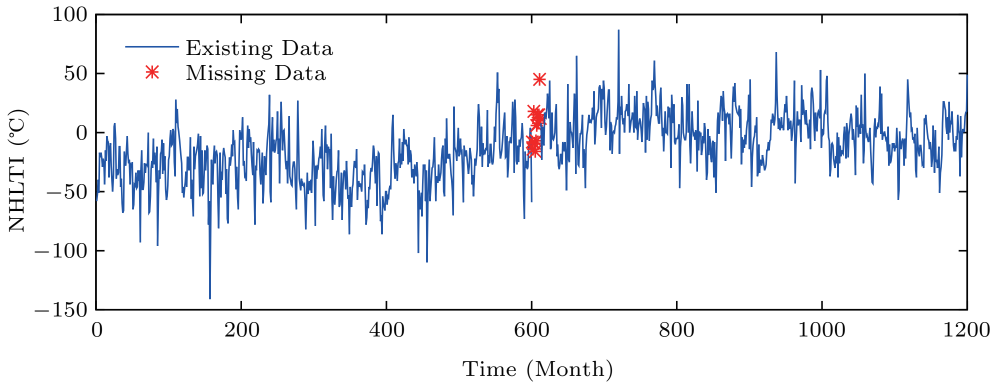
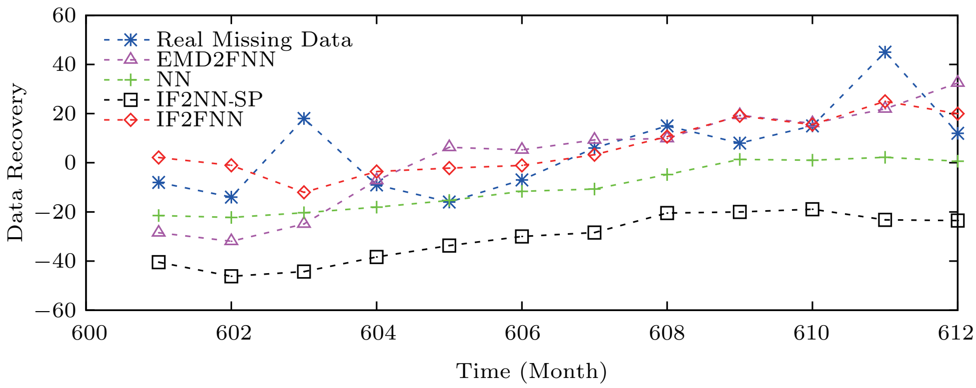


Fig.9. Damaged data and results of the models in data recovery on TMMT. (a) Damaged data and the data missing area (red star points). (b) Performance of the models including NN, IF2NN-SP, EMD2FNN, and IF2FNN.



(a)



(b)

Fig.10. Damaged data and results of the models in data recovery on NHLTI. (a) Damaged data and the data missing area (red star points). (b) Performance of the models including NN, IF2NN-SP, EMD2FNN, and IF2FNN.

was used to decompose the original nonlinear and non-stationary time series into IMFs that can be considered "quasi stationary". All of the produced IMFs were taken as inputs to the FM-based neural network, which incorporates the FM strategy to exploit the nonlinear interactions between features extracted at different time scales. Furthermore, we adopted four datasets with different instabilities including ACS, LOD, TMMT, NHLTI and NASDAQ to evaluate the proposed approach in both one-step, multi-step forecasting and data recovery tasks. The numerical experiments demonstrated that the integrated methods have significant advantages in improving the prediction and recovery accuracies, measured by MAE, RMSE and MAPE.

This study is another example that demonstrates the power of combining IF with neural network to achieve outstanding performance in time series prediction and recovery. In fact, we believe that IF can be used in conjunction with other statistical or artificial intelligence strategies to create general methods for predicting and missing data recovery. Various combinations in algorithm design can be investigated in the future to tackle problems emerging from different applications.

Table 8. Performances of Different Models for Data Recovery on TMMT

Model	MAE	MSE	MAPE
NN	0.273 2	0.306 0	0.654 2
IF2NN-SP	0.306 5	0.334 8	0.704 7
EMD2FNN	0.366 9	0.406 0	0.755 7
IF2FNN	0.241 2	0.291 2	0.491 3

Table 9. Performances of Different Models for Data Recovery on NHLTI

Model	MAE	MSE	MAPE
NN	15.466 7	19.771 6	1.155 6
IF2NN-SP	36.042 6	38.692 1	2.980 0
EMD2FNN	15.163 0	19.037 2	1.180 0
IF2FNN	10.419 9	10.797 5	0.787 8

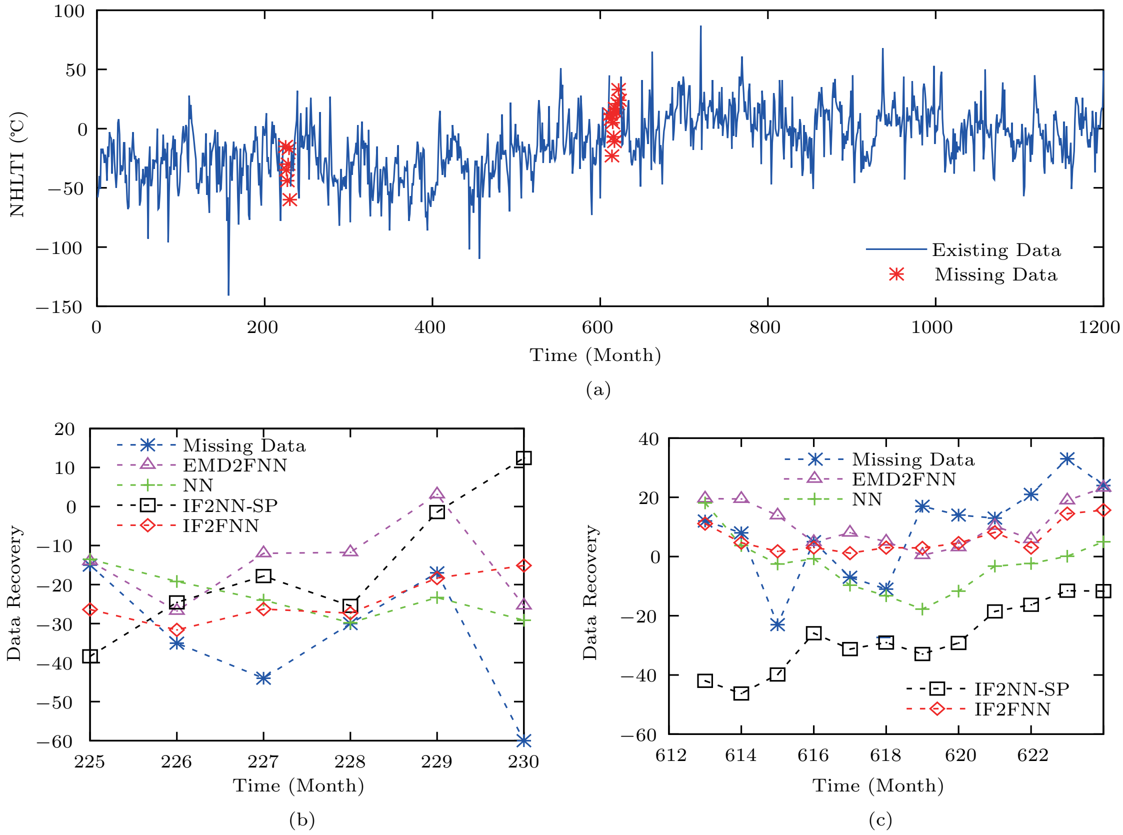


Fig.11. Damaged data and results of the models in data recovery on NHLTI. (a) Damaged data and the missing data (red star points) located in two areas. (b) Performance of the models including NN, IF2NN-SP, EMD2FNN and IF2FNN in recovering the data located in the first missing data area. (c) Performance of the models including NN, IF2NN-SP, EMD2FNN and IF2FNN in recovering the data located in the second missing data ares.

Table 10. Performances of Different Models for Two Missing Data Areas on NHLTI

Missing Data Area	Model	MAE	MSE	MAPE
[225, 230]	NN	12.450 4	16.563 0	0.316 3
	IF2NN-SP	25.431 1	33.796 5	0.788 0
	EMD2FNN	19.035 9	22.467 7	0.566 1
	IF2FNN	13.569 1	20.330 1	0.362 5
[612, 623]	NN	16.125 1	19.667 5	0.975 2
	IF2NN-SP	36.709 5	38.747 3	3.030 3
	EMD2FNN	12.260 0	15.433 0	0.869 1
	IF2FNN	10.530 7	12.716 2	0.670 1

Table 11. Results of Different Models for Recovering the Randomly Generated Single Missing Data of NASDAQ

Model	Missing Data				
	(1 084, 4 869.31)	(664, 4 442.69)	(221, 2 904.82)	(992, 4 927.85)	(886, 5 055.52)
SVM	4 788.61	4 357.83	3 163.97	4 744.32	4 726.94
GBDT	4 993.41	4 385.08	3 072.98	4 910.53	4 918.25
IF2SVM	4 845.12	4 378.04	3 164.97	4 759.20	4 695.66
IF2GBDT	4 807.63	4 441.60	3 079.83	4 873.85	4 891.52
NN	4 859.41	4 421.07	2 920.63	5 000.90	5 042.89
IF2NN-SP	4 754.61	4 435.03	2 975.04	4 964.23	5 046.83
EMD2FNN	4 864.04	4 452.21	3 002.24	4 997.11	4 954.64
IF2FNN	4 864.44	4 441.80	2 939.05	4 931.31	5 059.61

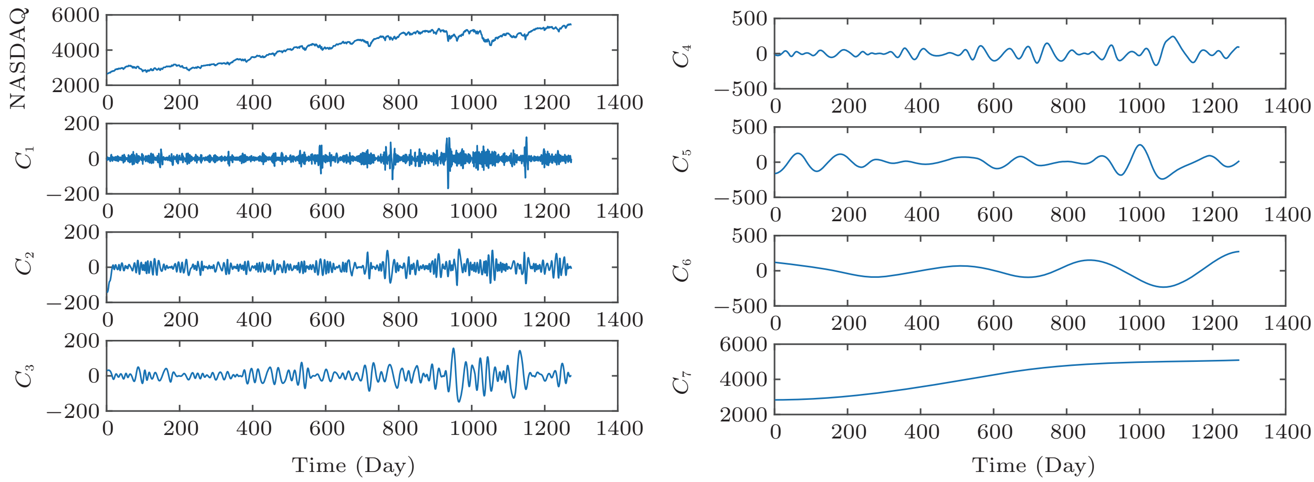


Fig.12. NASDAQ (upper right panel) and its IMFs obtained from IF.

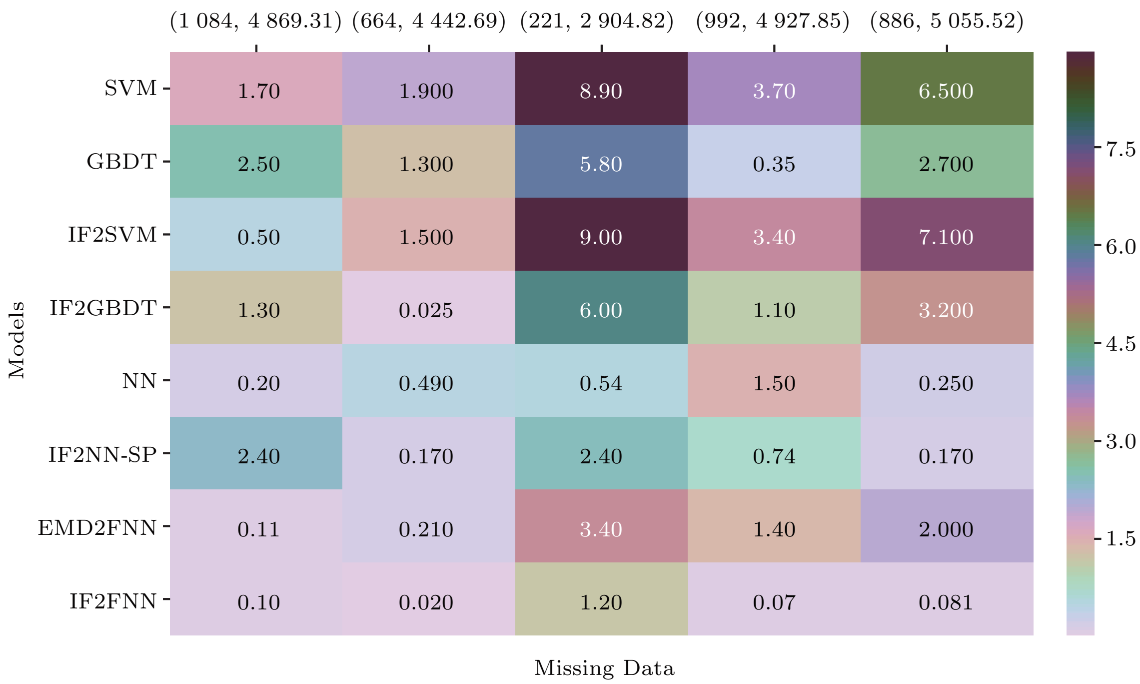


Fig.13. MAPE (%) of different models for recovering the randomly generated single missing data of NASDAQ.

References

[1] Safari N, Chung C Y, Price G C D. Novel multi-step short-term wind power prediction framework based on chaotic time series analysis and singular spectrum analysis. *IEEE Transactions on Power Systems*, 2018, 33(1): 590-601.

[2] Oh K J, Kim K J. Analyzing stock market tick data using piecewise nonlinear model. *Expert Systems with Applications*, 2002, 22(3): 249-255.

[3] Wang Y F. Mining stock price using fuzzy rough set system. *Expert Systems with Applications*, 2003, 24(1): 13-23.

[4] Faruk D Ö. A hybrid neural network and ARIMA model for water quality time series prediction. *Engineering Applications of Artificial Intelligence*, 2010, 23(4): 586-594.

[5] Kasabov N K, Song Q. DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, 2002, 10(2): 144-154.

[6] Franses P H, Ghijsels H. Additive outliers, GRACH and forecasting volatility. *International Journal of Forecasting*, 1999, 15(1): 1-9.

[7] Sarantis N. Nonlinearities, cyclical behaviour and predictability in stock markets: International evidence. *International Journal of Forecasting*, 2001, 17(3): 459-482.

[8] Kalekar P S. Time series forecasting using Holt-Winters exponential smoothing. https://c.mql5.com/forextsd/forum/69/exponentialsmoothing.pdf, Jan. 2019.

[9] Hansen J V, Nelson R D. Data mining of time series using stacked generalizers. *Neurocomputing*, 2002, 43(1/2/3/4): 173-184.

[10] Zhang G P. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 2003, 50: 159-175.

[11] Enke D, Thawornwong S. The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications*, 2005, 29(4): 927-940.

[12] Ture M, Kurt I. Comparison of four different time series methods to forecast hepatitis a virus infection. *Expert Systems with Applications*, 2006, 31(1): 41-46.

[13] Kim K J. Financial time series forecasting using support vector machines. *Neurocomputing*, 2003, 55(1/2): 307-319.

[14] Qian X Y. Financial series prediction: Comparison between precision of time series models and machine learning methods. arXiv:1706.00948, 2017. https://arxiv.org/abs/1706.00948, June 2018.

[15] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In *Proc. the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2016, pp.785-794.

[16] Ye J, Chow J H, Chen J, Zheng Z. Stochastic gradient boosted distributed decision trees. In *Proc. the 18th ACM-Conference on Information and Knowledge Management*, November 2009, pp.2061-2064.

[17] Kim K J, Han I. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications*, 2000, 19(2): 125-132.

[18] Wang Y F. Predicting stock price using fuzzy grey prediction system. *Expert Systems with Applications*, 2002, 22(1): 33-38.

[19] Shen L, Han T L. Applying rough sets to market timing decisions. *Decision Support Systems*, 2004, 37(4): 583-597.

[20] Vellido A, Lisboa P J G, Meehan K. Segmentation of the on-line shopping market using neural networks. *Expert Systems with Applications*, 1999, 17(4): 303-314.

[21] Chen A S, Leung M T, Daouk H. Application of neural networks to an emerging financial market: Forecasting and trading the Taiwan stock index. *Computers and Operations Research*, 2003, 30(6): 901-923.

[22] Rather A M, Agarwal A, Sastry V N. Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications*, 2015, 42(6): 3234-3241.

[23] Yang Z, Yang L, Qi D. Detection of spindles in sleep EEGs using a novel algorithm based on the Hilbert-Huang transform. In *Wavelet Analysis and Applications*, Qian T, Vai M I, Xu Y S (eds.), Birkhäuser, 2007, pp.543-559.

[24] Wang J Z, Wang J J, Zhang Z G, Guo S P. Forecasting stock indices with back propagation neural network. *Expert Systems with Applications*, 2011, 38(11): 14346-14355.

[25] Liu H, Chen C, Tian H Q, Li Y F. A hybrid model for wind speed prediction using empirical mode decomposition and artificial neural networks. *Renewable Energy*, 2012, 48: 545-556.

[26] Kao L J, Chiu C C, Lu C J, Chang C H. A hybrid approach by integrating wavelet-based feature extraction with MARS and SVR for stock index forecasting. *Decision Support Systems*, 2013, 54(3): 1228-1244.

[27] Zhang L, Wu X, Ji W, Abourizk S M. Intelligent approach to estimation of tunnel-induced ground settlement using wavelet packet and support vector machines. *Journal of Computing in Civil Engineering*, 2016, 31(2): Article No. 04016053.

[28] Wei L Y. A hybrid ANFIS model based on empirical mode decomposition for stock time series forecasting. *Applied Soft Computing*, 2016, 42: 368-376.

[29] Zhou F, Zhou H, Yang Z, Yang L. EMD2FNN: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction. *Expert Systems with Applications*, 2019, 115: 136-151.

[30] Thompson W R, Weil C S. On the construction of tables for moving-average interpolation. *Biometrics*, 1952, 8(1): 51-54.

[31] Watson D F. A refinement of inverse distance weighted interpolation. *GeoProcessing*, 1985, 2(4): 315-327.

[32] Liu G R, Zhang G Y. A novel scheme of strain-constructed point interpolation method for static and dynamic mechanics problems. *International Journal of Applied Mechanics*, 2009, 1(1): 233-258.

[33] Schoenberg I J. Contributions to the problem of approximation of equidistant data by analytic functions (part A). *Quarterly of Applied Mathematics*, 1946, 4: 3-57.

[34] Schoenberg I J. Cardinal Spline Interpolation. Society for Industrial and Applied Mathematics, 1973.

[35] Lin L, Wang Y, Zhou H. Iterative filtering as an alternative algorithm for empirical mode decomposition. *Advances in Adaptive Data Analysis*, 2009, 1(4): 543-560.

[36] Cicone A, Liu J, Zhou H. Adaptive local iterative filtering for signal decomposition and instantaneous frequency analysis. *Applied and Computational Harmonic Analysis*, 2016, 41(2): 384-411.

[37] Cicone A, Zhou H. Multidimensional iterative filtering method for the decomposition of high-dimensional non-stationary signals. *Numerical Mathematics: Theory, Methods and Applications*, 2017, 10(2): 278-298.

[38] Huang N E, Shen Z, Long S R, Wu M C, Shih H H, Zheng Q, Yen N C, Chi C T, Liu H H. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society A: Mathematical Physical and Engineering Sciences*, 1998, 454(1971): 903-995.

[39] Holt C C. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 2004, 20(1): 5-10.

[40] Winters P R. Forecasting sales by exponentially weighted moving averages. *Management Science*, 1960, 6(3): 231-362.

[41] Flandrin P, Rilling G, Goncalves P. Empirical mode decomposition as a filter bank. *IEEE Signal Processing Letters*, 2004, 11(2): 112-114.

[42] Zhou F, Yang L, Zhou H, Yang L. Optimal averages for nonlinear signal decompositions — Another alternative for empirical mode decomposition. *Signal Processing*, 2016, 121: 17-29.

[43] Huang N E, Shen Z, Long S R. A new view of nonlinear water waves: The Hilbert spectrum. *Annual Review of Fluid Mechanics*, 1999, 31(1): 417-457.

[44] Huang W, Shen Z, Huang N E, Yuan C F. Engineering analysis of biological variables: An example of blood pressure over 1 day. *Proceedings of the National Academy of Sciences of the United States of America*, 1998, 95(9): 4816-4821.

[45] Yang Z, Qi D, Yang L. Signal period analysis based on Hilbert-Huang transform and its application to texture analysis. In *Proc. the 3rd International Conference on Image and Graphics*, April 2005, pp.430-433.

[46] Smith J S. The local mean decomposition and its application to EEG perception data. *Journal of the Royal Society Interface*, 2005, 2(5): 443-454.

[47] Delechelle E, Lemoine J, Niang O. Empirical mode decomposition: An analytical approach for sifting process. *IEEE Signal Processing Letters*, 2005, 12(11): 764-767.

[48] Diop E H S, Alexandre R, Boudraa A O. Analysis of intrinsic mode functions: A PDE approach. *IEEE Signal Processing Letters*, 2010, 17(4): 398-401.

[49] Hong H, Wang X, Tao Z. Local integral mean-based sifting for empirical mode decomposition. *IEEE Signal Processing Letters*, 2009, 16(10): 841-844.

[50] Peng S, Hwang W L. Null space pursuit: An operator-based approach to adaptive signal separation. *IEEE Transactions on Signal Processing*, 2010, 58(5): 2475-2483.

[51] Daubechies I, Lu J, Wu H T. Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool. *Applied and Computational Harmonic Analysis*, 2011, 30(2): 243-261.

[52] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 39(6): 1137-1149.

[53] Shelhamer E, Long J, Darrell T. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 39(4): 640-651.

[54] Hinton G, Deng L, Yu D, Dahl G E, Mohamed A, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath T N. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 2012, 29(6): 82-97.

[55] Chen C H. Handbook of Pattern Recognition and Computer Vision (5th edition). World Scientific Publishing, 2016.

[56] Goldberg Y. Neural Network Methods for Natural Language Processing. Morgan and Claypool Publishers, 2017.

[57] Rendle S. Factorization machines. In *Proc. the 10th IEEE International Conference on Data Mining*, December 2010, pp.995-1000.

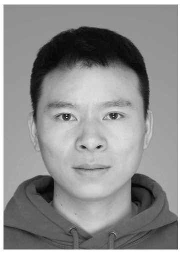
[58] Han J, Moraga C. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *Proc. International Workshop on Artificial Neural Networks: From Natural to Artificial Neural Computation*, June 1995, pp.195-201.

[59] Lecun Y, Bengio Y, Hinton G. Deep learning. *Nature*, 2015, 521(7553): 436-444.

[60] He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proc. the 2015 IEEE International Conference on Computer Vision*, December 2015, pp.1026-1034.

[61] Clevert D A, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (ELUs). arXiv:1511.07289, 2015. https://arxiv.org/pdf/1511.07289.pdf, November 2018.

[62] Makridakis S. Accuracy measures: Theoretical and practical concerns. *International Journal of Forecasting*, 1993, 9(4): 527-529.



Feng Zhou received his B.S. degree in information computing science from Minnan Normal University, Zhangzhou, in 2010, and Ph.D. degree in computational mathematics from Sun Yat-sen University, Guangzhou, in 2015. As a visiting student, he studied in the School of Mathematics, Georgia Institute of Technology, Atlanta, from 2013 to 2014, supported by the China Scholarship Council (CSC). From 2015 to 2017, he worked as a sensor algorithm engineer in Baidu and Tencent successively in the research of recommendation system. Now he is an assistant professor at the School of Information Science, Guangdong University of Finance and Economics, Guangzhou. His research interests include signal analysis, machine learning and ensemble learning.

Hao-Min Zhou is a professor in School of Mathematics at the Georgia Institute of Technology, Atlanta. He received his B.S. degree in pure mathematics from Peking University, Beijing, in 1991, his M.S. degree in scientific computing from Peking University, Beijing, and M.Phil degree in applied mathematics from the Chinese University of Hong Kong, Hong Kong, in 1994 and 1996 respectively, and his Ph.D. degree in applied mathematics from University of California, Los Angeles, in 2000. He had spent three years in California Institute of Technology as a postdoctoral scholar and von Karman instructor before he joined Georgia Institute of Technology in 2003. His research interests are on numerical analysis and scientific computing, specialized in digital image processing, wavelets and nonlinear signal processing, diffusion on graphs, multi-scale PDE computations, stochastic differential equations and their applications in engineering such as optics and optimal control.

Zhi-Hua Yang received his M.S. degree in automation from Hunan University, Changsha, and his Ph.D. degree in computer science from Sun Yat-sen University, Guangzhou, in 1995 and 2005, respectively. He was a visiting scholar with the Center of Pattern Recognition and Machine Intelligence, Concordia University, Montreal, in 2009. Since 2009, he has been a professor with the Information Science School, Guangdong University of Finance and Economics, Guangzhou. His current research interests include signal analysis, pattern recognition, and image processing. He has authored over 40 articles.

Li-Hua Yang received his B.S., M.S. and Ph.D. degrees in mathematics from Hunan Normal University, Changsha, Beijing Normal University, Beijing, and Sun Yat-sen University, Guangzhou, in 1984, 1987, and 1995 respectively. He is now a professor at School of Mathematics, Sun Yat-sen University, Guangzhou. From 1996 to 1998, he worked as a postdoctoral fellow in the Institute of Mathematics, Academia Sinica, Beijing. Since 1997, he was worked in short terms as a visiting scholar in Hong Kong Baptist University, CENPARMI at Concordia University, Syracuse University, Michigan State University, and City University of Hong Kong. He has been actively involved in many international conferences as an organizing committee and/or technical program committee member. His current research interests include signal processing and machine learning. He has published more than 100 papers, two books and three translations. He was the supervisor of 18 Ph.D. candidates.