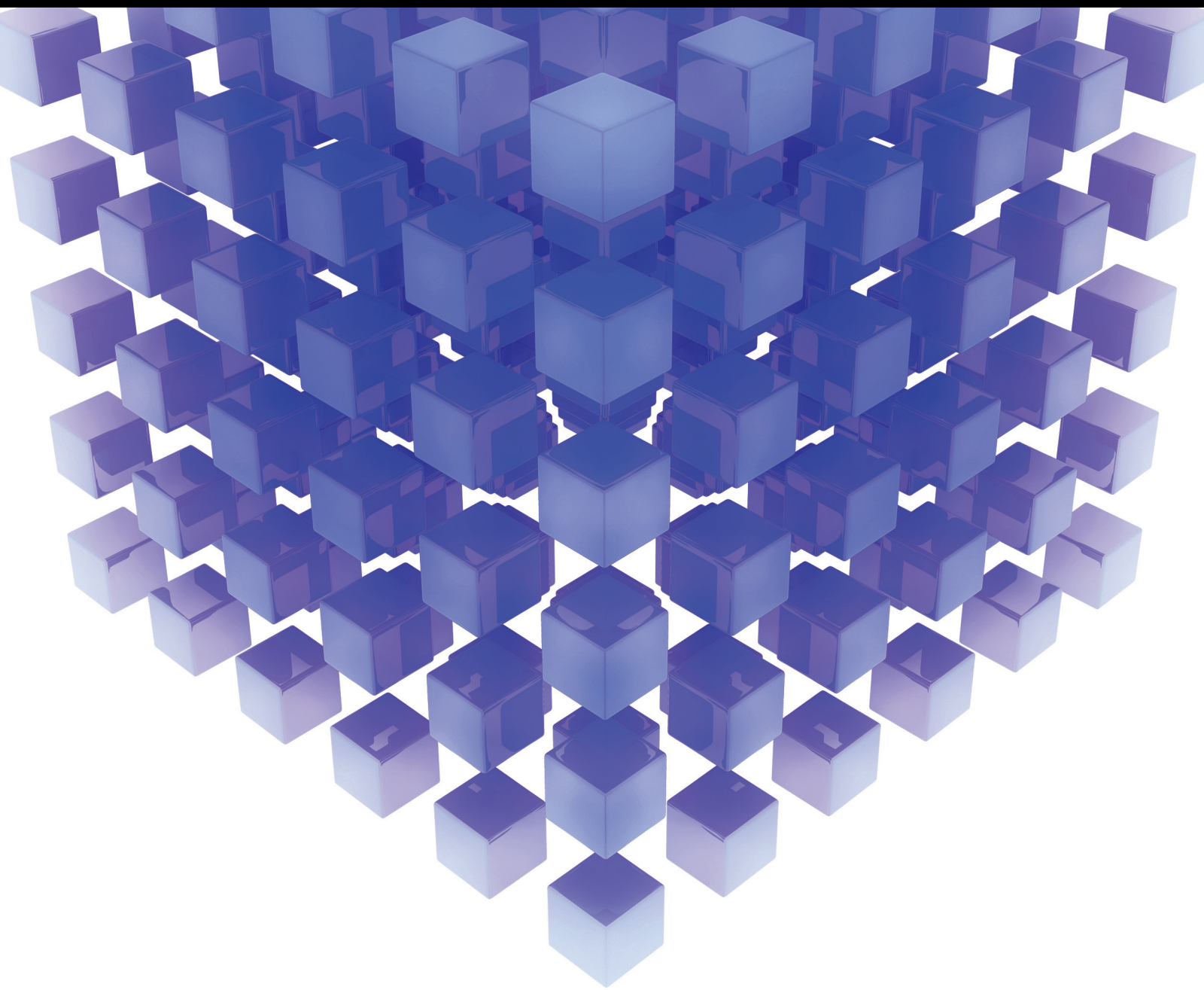
Mathematical Problems in Engineering

Mathematical Theories in the Era of Big Data

Lead Guest Editor: Ester Zumpano
Guest Editors: Luciano Caroprese, Florin Radulescu, Andrea Calì, and Pierangelo Veltri



Mathematical Theories in the Era of Big Data

Mathematical Problems in Engineering

Mathematical Theories in the Era of Big Data

Lead Guest Editor: Ester Zumpano
Guest Editors: Luciano Caroprese, Florin Radulescu, Andrea Calì, and Pierangelo Veltri

Editorial Board

Contents

Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 9231923, 2 pages
https://doi.org/10.1155/2019/9231923

Editorial

Mathematical Theories in the Era of Big Data

Ester Zumpano,[1] **Luciano Caroprese**,[1] **Pierangelo Veltri**,[2] **Andrea Calì**,[3] **and Florin Radulescu**[4]

[1]*DIMES, University of Calabria, Rende, Italy*
[2]*DSMC, University "Magna Graecia" of Catanzaro, Catanzaro, Italy*
[3]*Birkbeck, University of London, London, UK*
[4]*University of Rome Tor Vergata, Rome, Italy*

Correspondence should be addressed to Ester Zumpano; e.zumpano@dimes.unical.it

Received 2 April 2019; Accepted 3 April 2019; Published 11 April 2019

Data integration concerns the process of acquiring and managing heterogeneous data to be used by means of a unified view. Data can be merged in a unique data structure and can reside on different data sources and can be reconciled in the user view. Data is growing and huge increasing volume of data is available in different information sources; thus that furnishing uniquely available user interface is always more interesting challenge. To address this, data integration has become, over the last decades, the focus of extensive computer science theoretical works focusing on schema alignment and data fusion. Nevertheless, many issues are still open problems and thus unsolved.

The recent years have seen an impressive growth in the volume, speed, and heterogeneity of the generated data as well as in the variety and quality of the data. We are in the era of big data! Data is generated, collected, and processed at an unprecedented scale and data-driven decisions influence many aspects of modern society. Data integration contributes to rapid and efficient decisions and is required in social and life related areas such as emergency management, life quality, and health related data management. As a consequence, there is a growing interest in applying mathematical theories and methods to model, integrate, and manage massive and fast changing data and in retrieving the valid and valuable knowledge they imply.

The target of this special issue was to disseminate recent research results on data integration and to promote the *integration* between data management and knowledge representation communities. The aim was to merge articles describing novel theoretical as well as applied works regarding methodologies for big data modeling, integration, and management.

In the paper "Big Data Validity Evaluation Based on MMTD" by N. Zhou et al., medium mathematics systems are introduced for the evaluation of big data validity. A medium logic-based data validity evaluation method is proposed. The contributions of the paper are as follows: based on the 3V properties of big data, dimensions that have a major influence on data validity are determined; data completeness, correctness, and compatibility are defined; a medium truth degree-based model is proposed to measure each dimension of data validity; a medium truth degree-based multidimensional model is proposed to measure the integrated value of data validity.

In the paper "A Compound Structure for Wind Speed Forecasting Using MKLSSVM with Feature Selection and Parameter Optimization" by S. Sun et al., a compound MKLSSVM model optimized by HGSA algorithm integrated with signal decomposition technique EEMD, namely, EEMD-HGSA-MKLSSVM, is proposed for short-term wind speed forecasting. Four sets of mean half-hour wind speed, selected randomly from the historical wind speed data in 2015 and collected from a wind farm located in Anhui of China, are utilized as case studies to evaluate the forecasting performance of EEMD-HGSA-MKLSSVM model.

In the paper "A Negotiation Optimization Strategy of Collaborative Procurement with Supply Chain Based on Multi-Agent System" by C. Chen and C. Xu, the process

of collaborative procurement in which buyers and suppliers are prone to conflict in cooperation due to differences in needs and preferences is investigated. The paper provides a novel perspective for the analysis of intelligent supply chain managements; it constructs a negotiation model based on multi-agent system and proposes a negotiation optimization strategy combined with machine learning.

In the paper "High-Order Degree and Combined Degree in Complex Networks" by S. Wang et al., several novel centrality metrics are defined: the high-order degree and combined degree of undirected network, the high-order out-degree and in-degree and combined out out-degree and in-degree of directed network. Those are the measurement of node importance with respect to the number of the node neighbors. Centrality metrics are explored in the context of several best-known networks and it is proved that both the degree centrality and eigenvector centrality are special cases of the high-order degree of undirected network, and both the in-degree and PageRank algorithm without damping factor are special cases of the high-order in-degree of directed network.

Conflicts of Interest

The editors declare that they have no conflicts of interest regarding the publication of this special issue.

Acknowledgments

The guest editorial team would like to thank all authors for their contributions and the reviewer for their insightful comments.

Ester Zumpano
Luciano Caroprese
Pierangelo Veltri
Andrea Calì
Florin Radulescu

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 9287097, 21 pages
https://doi.org/10.1155/2018/9287097

Research Article

A Compound Structure for Wind Speed Forecasting Using MKLSSVM with Feature Selection and Parameter Optimization

Sizhou Sun,[1,2] **Jingqi Fu**,[1,3] **Feng Zhu**,[1,3] **and Nan Xiong**[1,3]

[1]*School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, 200072, China*
[2]*College of Electrical Engineering, Anhui Polytechnic University, Wuhu, 241000, China*
[3]*Shanghai Key Laboratory of Power Station Automation Technology, Shanghai University, Shanghai, 200072, China*

Correspondence should be addressed to Jingqi Fu; jqfu@staff.shu.edu.cn

Received 8 February 2018; Revised 14 July 2018; Accepted 30 August 2018; Published 14 November 2018

Academic Editor: Luciano Caroprese

The aims of this study contribute to a new hybrid model by combining ensemble empirical mode decomposition (EEMD) with multikernel function least square support vector machine (MKLSSVM) optimized by hybrid gravitation search algorithm (HGSA) for short-term wind speed prediction. In the forecasting process, EEMD is adopted to make the original wind speed data decomposed into intrinsic mode functions (IMFs) and one residual firstly. Then, partial autocorrelation function (PACF) is applied to identify the correlation between the corresponding decomposed components. Subsequently, the MKLSSVM using multikernel function of radial basis function (RBF) and polynomial (Poly) kernel function by weight coefficient is exploited as core forecasting engine to make the short-term wind speed prediction. To improve the regression performance, the binary-value GSA (BGSA) in HGSA is utilized as feature selection approach to remove the ineffective candidates and reconstruct the most relevant feature input-matrix for the forecasting engine, while real-value GSA (RGSA) makes the parameter combination optimization of MKLSSVM model. In the end, these respective decomposed subseries forecasting results are combined into the final forecasting values by aggregate calculation. Numerical results and comparable analysis illustrate the excellent performance of the EEMD-HGSA-MKLSSVM model when applied in the short-term wind speed forecasting.

1. Introduction

Owing to the abundant, renewable, and economical characteristics, the exploitation and utilization technique of renewable wind energy have attracted extensive attention of the scientific researchers. Wind energy has been considered as an effective way to address the global energy demands and eliminate green-house gas emissions [1]. In the past few years, wind energy has experienced fast growth worldwide. *World Wind Energy Association* reports that the total installed wind turbine capacity of the top 10 countries by the end of 2016 has approximately amounted to 410.613 *GW* and all the wind turbines worldwide by mid-2016 can generate about 4.7% of the global electricity demand [2]. However, for the high fluctuation and nonlinear and uncontrollable nature of wind speed, the integration of power system with large capacity of wind power has brought new challenges to the operation security and reliability of power system and the management of wind farms. Accurate short-term wind power output forecasting has been considered as one of the most economical and effective approaches to eliminate these problems; therefore, wind speed forecasting is a fundamental task in the routine operation management of wind farms [2–4].

Over the past decades, many methods and models, mainly including physical model, statistical model, and artificial intelligent method, are widely applied to predict the short-term wind speed [5, 6]. Physical model is generally applied in the large-term wind speed forecasting by usage of detailed meteorological data and environmental information, while statistical models are constructed commonly for short-term wind speed forecasting by revealing explicitly the linear relationship among the wind speed time series [7, 8]. Different from physical models and statistical methods, the artificial intelligent methods can tackle nonlinear problems better, thus, they are the most

popular and extensive approaches to apply in the short-term wind speed forecasting. To name a few here, backpropagation neural network (BPNN) [9], artificial neural networks (ANN) [10], fuzzy neural network (FNN) [11], support vector machine (SVM) [12–14], extreme learning machine (ELM) [2, 15], and least square support vector machine (LSSVM) [5] are mainly artificial intelligent methods for wind speed prediction.

As stated in [5], the single artificial intelligent model cannot work well when applied in wind speed forecasting in that wind speed exhibits high nonlinearity. Wind speed forecasting by the single model using directly the raw wind speed data without disposal is easily subjected from large errors; hence, multiscale decomposition or denoising processing techniques are utilized to preprocess wind data, and intelligent algorithms are used to tune the parameters in the forecasting engine. For example, Liu at al. [13] developed a hybrid forecasting model combining Wavelet Transform (WT) with SVM tuned by general algorithm (GA). Meng et al. [16] applied the signal analysis method WPD to realize the decomposition of the original wind speed data into several different subseries; then, each decomposed component with different frequency was submitted to ANN tuned by crisscross optimization algorithm for the multistep wind speed forecasting. Wang et al. [17] developed a hybrid prediction method using EEMD and BPNN. Abdoos [18] took advantages of the combination of VMD with ELM for short-term wind power prediction. These hybrid forecasting models discussed above improve the prediction performance mainly by integration of the individual advantages of signal preprocessing technique and optimization algorithm and artificial intelligent model.

Among these data preprocessing-based techniques as discussed and analyzed above, WT has sensitivity in the choice of threshold and the figuration of its wavelet basis should be determined beforehand, while EMD is sensitive to noise and suffers from mode mixing problems [6, 19]. EEMD method can eliminate the drawbacks of the decomposition approaches to some extent. EEMD is an empirical and self-adaptive signal processing approach which is widely used to analyze the nonlinear and nonstationary signal so that we use EEMD to decompose and analyze the original wind speed data in this study. LSSVM, proposed by Suykens [20], is an improved version of SVM, which lowers calculation complexity by translating convex quadratic programming problems into solving linear equations [21]. LSSVM can exhibit some advantages in solving small samples, nonlinearity, and pattern recognition with excellent generalization ability [22] and has been successfully applied in time series-based wind speed forecasting [10, 21, 23], and therefore LSSVM algorithm is adopted as the core forecasting engine for short-term wind speed forecasting.

Even though these signal decomposition based models have obtained good forecasting results, Wang et al. [24] pointed out that not all decomposed subseries are a benefit for the final wind speed forecasting. To address this problem, the feature selection method is utilized widely [25, 26]. In [8], Kullback-Leibler divergence-based and energy-based feature selections were exploited to identify the illusive components caused by the decomposed method EEMD. In [27], Salcedo-Sanz developed a hybrid model of physical model and ELM, where coral reefs optimization algorithm (CRO) was utilized as feature selection to select the useful meteorological predictive information from the output of the physical approach. In the hybrid model, removing the ineffective input candidate and decreasing the dimension of the input-matrix by feature selection, ELM model can better train and regress, thus improving the forecasting performance. In [28], a hybrid GSA integrating the binary-value GSA and real-value GSA is introduced for feature selection and optimization of the weights and biases in the ELM to diagnose fault of rolling element bearings. In these hybrid algorithms, feature selection removes the ineffective variables and determines the useful input candidate to construct the input-matrix for the forecasting engine while the optimization algorithm tunes the parameters in the forecasting engine other than random initialization, and therefore the hybrid forecasting model can obtain better forecasting results.

Inspired by these forecasting mechanisms, a novel hybrid model EEMD-HGSA-MKLSSVM is proposed for short-term wind speed prediction. Four sets of actual historical wind speed data from a wind farm located in Anhui of China are utilized as training and test samples to evaluate the proposed forecasting model. Considering the previous studies in the same research fields, the main works and contributions of this paper are summarized as

(i) A novel proposed forecasting model takes individual advantage of EEMD and HGSA and MKLSSVM to enhance wind speed prediction accuracy.

(ii) To improve the regression performance, radial basis function (RBF) with local exploitation capacity and polynomial (poly) kernel function with global exploration capacity are employed to construct multikernel function by weight coefficient for LSSVM, namely, MKLSSVM.

(iii) The forecasting accuracy and stability of the forecasting engine are enhanced by identification of the useful input variables and determination of optimal parameters through HGSA algorithm simultaneously.

(iv) To examine the performance of the proposed combined architecture, a number of simulation experiments are carried out and compared using four sets of wind speed data.

The remaining of this study are structured as follows. In Section 2, the individual models, including EEMD and HGSA and MKLSSVM, are introduced. Section 3 illustrates the detailed working principle of the proposed EEMD-HGSA-MKLSSVM model and performance evaluation indices. Case studies are implemented to evaluate the proposed hybrid model for short-term wind speed forecasting in Section 4. Conclusions are drawn in the final section.

2. Methodology

2.1. Wind Speed Decomposition Method. EMD, a self-adaptive signal analysis technique, is developed to analyze and decompose nonlinear signals by sifting process [6]. However, EMD easily suffers from the mode mixing problem that defined as a single IMF containing signals with dramatically disparate scales or a component of a similar scale residing in different IMFs, which causes easily intermittency in analyzing signals [6]. To eliminate the mode mixing problems caused by the EMD, a novel nonlinear signal analysis method EEMD was developed by adding white noises with finite amplitude to the original signals and offsetting themselves through ensemble averaging [17]. The analysis process of signals by EEMD algorithm can be described as in the following steps:

(i) *Step1.* Add Gaussian distribution white noise $n(t)$ with finite amplitude to the original signal data $x(t)$ to obtain a new signal $s(t)$ expressed as

$$s(t) = x(t) + n(t). \tag{1}$$

(ii) *Step2.* Decompose $s(t)$ into a series of IMF components c_i and residual component r_n using a standard EMD method. After decomposition, $s(t)$ can be mathematically expressed as

$$s(t) = \sum_{i=1}^{N} c_i + r_N. \tag{2}$$

(iii) *Step3.* Repeating step from 1 to 2 and add random white noise each time, these N groups of different white noise have characteristics of uncorrelated relationship and its statistical mean is zero.

(iv) *Step4.* Offsetting the impact of the Gaussian white noise by the final mean of the corresponding IMFs as (3), by the same way, the final residue can be obtained as

$$c_j = \frac{1}{N}\sum_{i=1}^{N} c_{ij}. \tag{3}$$

(v) *Step5.* In the end, the noise-free signal data are obtained by reconstruction of the IMF components c_j and the residual component r_N.

The effects of the added Gaussian distribution white noise are ensured by the statistical rule proved by Wu and Huang [29],

$$\varepsilon_{ne} = \frac{\varepsilon}{\sqrt{N_e}}, \tag{4}$$

where N_e is the ensemble members, ε is the amplitude of the added noise, and ε_{ne} is defined as the difference between the input signal and the corresponding IMFs.

To better illustrate the effectiveness of EEMD in overcoming the mode mixing problem, a given synthetic test signal $y(t)$ expressed as (5) consisting of a sinusoid signal $y_1(t)$ expressed as (6) and an intermittent signal $y_2(t)$ expressed as (7), which are displayed in Figure 1, is utilized to test EMD and EEMD.

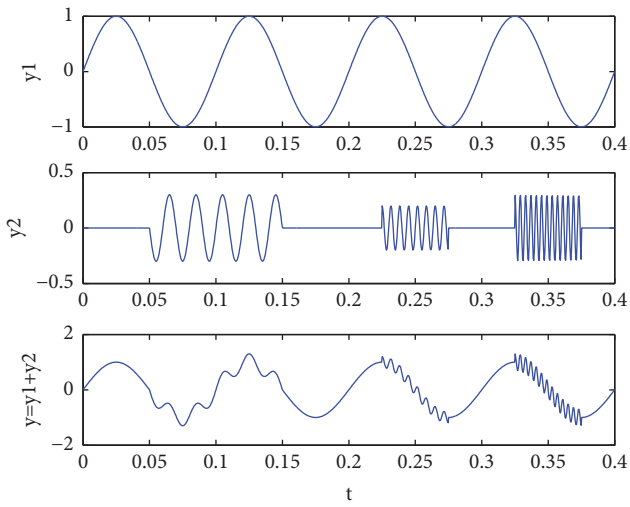


Figure 1: Synthesized signal y.

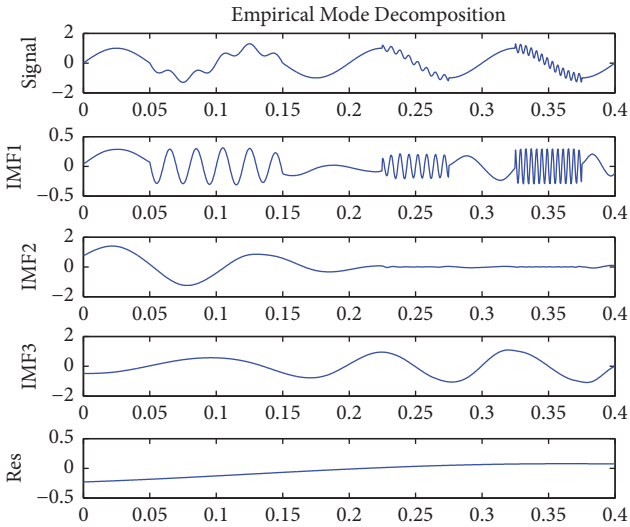


Figure 2: Decomposed results of the synthesized signal y by EMD.

$$y = y_1 + y_2, \tag{5}$$

$$y_1 = \sin(20\pi t) \quad 0 \le t \le 0.4, \tag{6}$$

$$y_2 = \begin{cases} 0.3\sin(100\pi t) & 0.05 \le t \le 0.15, \\ 0.2\sin(300\pi t) & 0.325 \le t \le 0.375, \\ 0.3\sin(500\pi t) & 0.225 \le t \le 0.275, \\ 0 & . \end{cases} \tag{7}$$

The decomposed results of the synthesized signal y by EMD and EEMD are shown in the Figures 2 and 3, respectively. It is obviously seen from Figures 2 and 3 that the mode mixing problems exist in the different components decomposed by EMD, while the mode mixing problems have been eliminated by EEMD and the intermittent signals embedded in the synthesized signal have been extracted successfully.

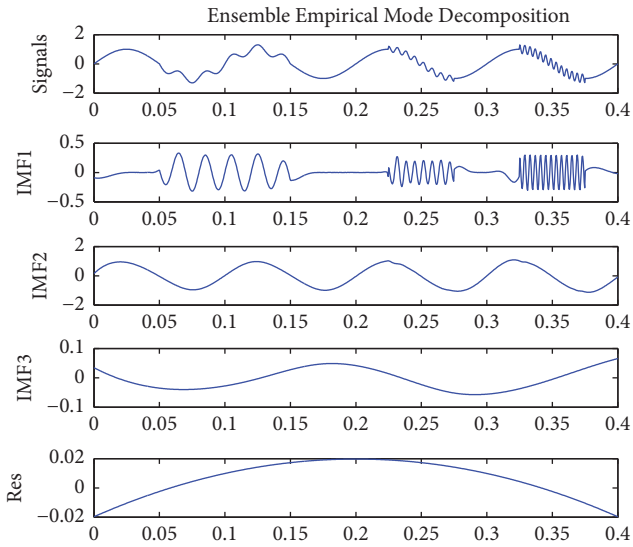


FIGURE 3: Decomposed results of the synthesized signal y by EEMD.

2.2. Multikernel LSSVM(MKLSSVM). In this study, LSSVM is adopted in that it is expert in addressing small sample problems [21] and has high generalization performance [30]. LSSVM is a type of powerful artificial intelligence technology based on the structural risk minimization. The basic principle of single output LSSVM regression is shown as follows.

Assume that the training sample set $\{x_i, y_i\}$, where $i = 1 \cdots N$, N is the total number of training samples; x_i is input training sample and y_i is its corresponding output. The regression function can be expressed as

$$y = w^T \varphi(x) + b, \tag{8}$$

where w and b denote the weight vector and the bias term, respectively. $\varphi(\cdot)$ is a nonlinear mapping function which maps the training samples into a high-dimension feature space where regression is carried out. The regression can be calculated by minimizing a cost function expressed as

$$\min C = \min\left(\frac{1}{2}\omega^T\omega + \frac{1}{2}\gamma\sum_{i=1}^{N} e_i^2\right), \tag{9}$$

subject to the equality constraints as

$$y_i = \omega^T\varphi(x_i) + b + e_i, \quad i = 1 \cdots N. \tag{10}$$

The first part of the cost function in (9) is utilized to regularize the weight sizes and penalize weights and γ is a regularization parameter which is optimized by user to control the trade-off between the bias and variance of LSSVM. The Lagrange function is constructed as follows to solve the convex optimization problem:

$$L(\omega, b, e, \alpha) = C - \sum_{i=1}^{N}\beta_i\left(\omega^T\varphi(x) + b - y_i + e_i\right), \tag{11}$$

where β_i are Lagrange multipliers. By partially differentiating with respect to ω, b, e, and β_i, eliminating ω and e, the solution of (11) can be obtained by the following linear regression function:

$$y = \sum_{i=1}^{N}\beta_i k(x, x_i) + b, \tag{12}$$

where $k(x, x_i)$ is a positive definite kernel function which meets Mercer's condition. The different type of kernel function influences the regressive performance of LSSVM model. In this paper, to improve the generalization capacity, a weighted multikernel LSSVM combining RBF kernel function and Poly kernel function is constructed and expressed as

$$k_{mix}(x_i, x_j) = \mu k_{RBF}(x_i, x_j) + (1-\mu)\, k_{Poly}(x_i, x_j), \tag{13}$$

where μ represents the weight coefficient within [0 1] and RBF kernel function and Poly kernel function are expressed as (14) and (15), respectively.

$$k_{RBF}(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\delta^2}\right), \tag{14}$$

$$k_{poly}(x_i, x_j) = \left[(x_i \cdot x_j) + r\right]^d. \tag{15}$$

When $\mu = 1$, the weighted multiple-kernel function exhibits the characteristic of RBF function and $\mu = 0$; this shows the characteristic of Poly function. By adjusting the μ value, the weighted multiple-kernel function can be suitable for different input samples.

2.3. Hybrid Gravitational Search Algorithm (HGSA). GSA, a novel heuristic optimization algorithm, was firstly developed by Rashedi [31], and the standard GSA has been successfully used to solve the engineering optimization problems in real-value parameters domain. However, in the actual engineering application, there exist many binary-encoded optimization problems, such as feature selection, which need to be solved. In this study, the optimization objectives including input variables binary feature selection as well as the real-value kernel parameters and weighted coefficients in MKLSSVM are dealt simultaneously; thus, we develop a hybrid GSA combing BGSA for feature selection with standard GSA for parameter optimization.

2.3.1. Gravitational Search Algorithm (GSA). In GSA algorithm, the performance of the agent is evaluated by their masses, namely, the heavier agent represents the optimal solution; thus the agent's mass is considered as the objective. All the agents move towards the agents with heavier masses by the gravitational force between them. As a result, after many iterations, the heavier masses with the higher fitness are obtained.

Assume that there are N random agents in the GSA. Firstly, the speed and position of each agent should be initialized and the position of ith agent is $\{(X_i = x_i^1, x_i^2, \cdots, x_i^d, \ldots, x_i^D) \mid i = 1, 2, \ldots, N\}$ and D denotes the dimension of

the search space. At the tth iteration, the mass of the ith agent is mathematically expressed as

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)},$$
$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)}, \tag{16}$$

where $fit(t)$, $M(t)$, $worst(t)$, and $best(t)$ are fitness value, mass, worst, and best value, respectively. For the wind speed forecasting error minimization problem, the $worst(t)$ and $best(t)$ are expressed as

$$best(t) = \min_{j\in\{1,\cdots N\}} \left(fit_j(t)\right),$$
$$worst(t) = \max_{j\in\{1,\cdots N\}} \left(fit_j(t)\right), \tag{17}$$

The gravitational force that the ith mass $x_i^d(t)$ acts on the jth mass $x_j^d(t)$ is expressed as follows according to the Newton gravitation theory:

$$F_{ij}^d(t) = G(t)\frac{M_i(t)\times M_j(t)}{\left\|X_i(t), X_j(t)\right\|_2 + \varepsilon}\left(x_j^d(t) - x_i^d(t)\right),$$
$$G(t) = G_0 \times \exp\left(-\alpha\frac{t}{iter_{\max}}\right), \tag{18}$$

where G_0, α, k, and $iter_{max}$ are the initial gravitational constant, attenuation factor, the current iteration number, and the maximum iteration number, respectively.

The resultant force that other masses act on the ith mass is calculated as (19) by a randomly weighted sum of $F_{ij}^d(t)$. Afterwards, the acceleration $a_i^d(t)$, velocity $v_i^d(t)$, and position $x_i^d(t)$ of an agent are calculated as (20):

$$F_i^d(t) = \sum_{i\neq j} rand_j \times F_{ij}^d(t) \tag{19}$$

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t)$$
$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{20}$$

where $rand_i$ and $rand_j$ are random value within [0 1] and $a_i^d(t) = F_i^d(t)/M_i(t)$.

2.3.2. Binary GSA(BGSA). The real-value GSA is exploited to solve the optimization problems in the continuous space, which can be not directly employed to solve the binary problems. The binary GSA (BGSA), developed by Rashedi et al. [32], is used to deal with discrete binary problems, and its working mechanism is that the velocity of each agent is converted by the Hyperbolic tangent function into a probability value which is expressed as

$$S\left(v_i^d(t)\right) = \left|\tanh\left(v_i^d(t)\right)\right|, \tag{21}$$

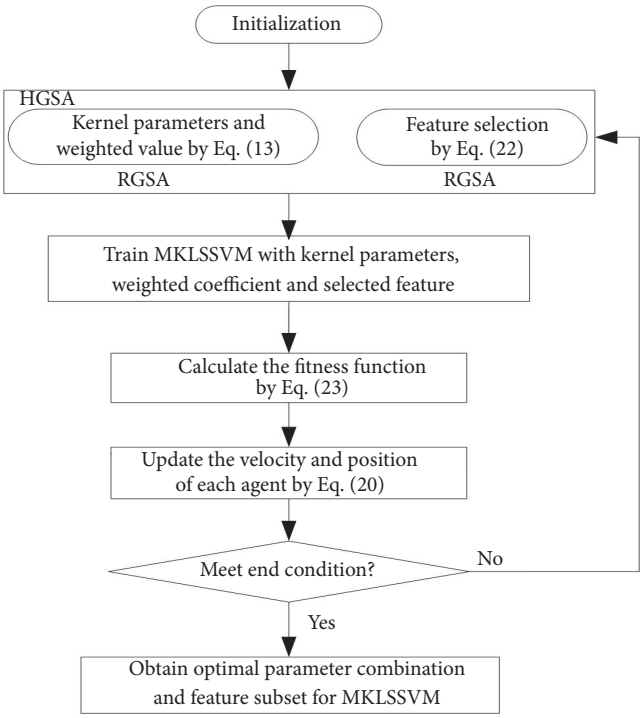


Figure 4: Feature selection and parameter optimization using HGSA.

where tanh(·) denotes the Hyperbolic tangent function. Based on the above (21), each dimension in the discrete binary space takes on "0" or "1" binary value by

$$if\left(rand < S\left(v_i^d(t)\right)\right)$$
$$x_i^d(t+1) = complement\left(x_i^d(t)\right) \tag{22}$$
$$else\ x_i^d(t+1) = x_i^d(t)$$

where $complement(\cdot)$ stands for the logical negation.

3. The Proposed Forecasting Strategy

3.1. Feature Selection and Parameter Optimization by HGSA. To eliminate the uncorrelated and ineffective variables in the input samples and avoid the drawbacks of trapping in local optima or overfitting of MKLSSVM, HGSA is used as feature selection and parameter optimization to solve this problems simultaneously. As seen from Figure 4, a hybrid optimization algorithm HGSA is exploited for the optimization of the kernel parameters and weighted coefficient in MKLSSVM by GSA and feature selection of input variables by BGSA to improve the forecasting performance of the forecasting strategy. During the optimization procedure, the root mean square error (RMSE), expressed in (23), between the training results of MKLSSVM and the measured wind speed data is used as fitness function.

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}|s(t) - \hat{s}(t)|^2}, \tag{23}$$

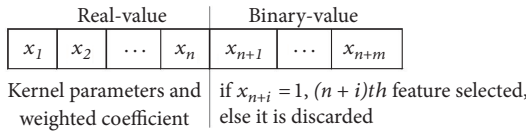


Figure 5: Solution representation.

where $s(t)$ and $\hat{s}(t)$ denote the measured wind speed and the forecasting wind speed value, respectively. N stands for the total number of samples.

With this in mind, the hybrid optimization problem can be illustrated in Figure 5, and there are n real-valued kernel parameters and weighted coefficient and m binary-valued parameters. The binary value "1" in BGSA algorithm means that the input variable is selected while "0" means that the input variable is not considered. Thus, the initialization encoding dimension of an agent in HGSA is set as $(n + m)$-length vector.

3.2. Specific Steps of EEMD-HGSA-MKLSSVM Model. In this study, LSSVM, based on weighted multikernel function, is utilized as the core forecasting engine in the forecasting strategy. The single LSSVM model has many advantages in solving small sample and nonlinearity. However, owing to the irregularity and randomness of wind speed time series, it cannot obtain the favorable prediction performance when independently applied in wind speed forecasting. To better catch the characteristics of wind speed, a hybrid approach combining EEMD and MKLSSVM with HGSA together is proposed as shown in Figure 6 and its working process is as follows:

(i) Use EEMD method to break down the empirical wind speed into IMFs and *Res* with different frequency.

(ii) Prior to forecasting by MKLSSVM model, the PACF that is widely used as a lag identification approach in Auto Regression (AR) (p) is applied to determine the correlation coefficients of the inputs. When the PACF values at lags bigger than p are approximately independent $N(0, 1/n)$ random variables, the lag of the sample can be determined as p.

(iii) To lower the forecasting difficulties of the MKLSSVM model, the inputs are normalized linearly to interval $[0, 1]$.

(iv) Train the MKLSSVMs optimized by the HGSA algorithm using the different frequency subseries (IMFs and *Res*). The $1st$-$480th$ wind speed series in Figure 7 are adopted as the training dataset.

(v) Apply the well-trained HGSA-MKLSSVM models to do the multistep ahead wind speed forecasting using each subseries. The subsequent $481th$-$576th$ sampling points in Figure 7 are used as the test dataset. Additionally, a rolling forecasting mechanism is adopted in the prediction processes.

(vi) Obtain the final forecasting results by aggregating the calculation after denormalization. In the end, compare the forecasting performance between EEMD-HGSA-MKLSSVM and other wind speed forecasting methods.

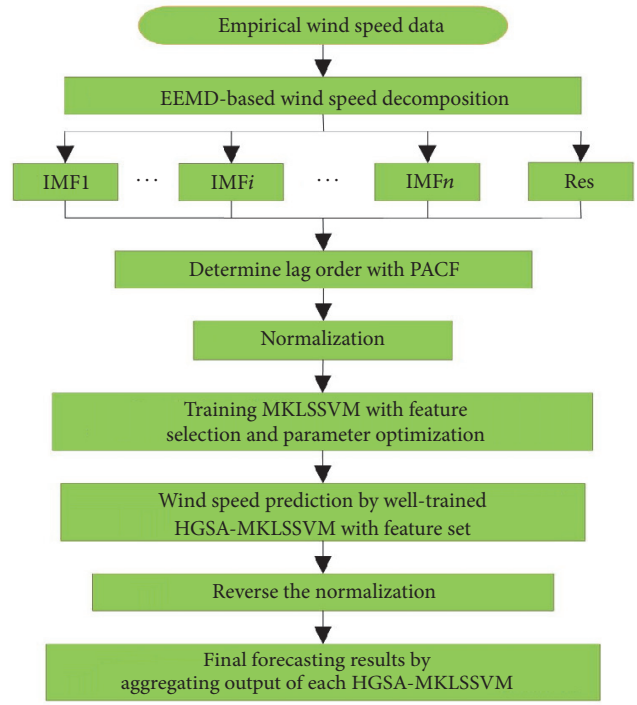


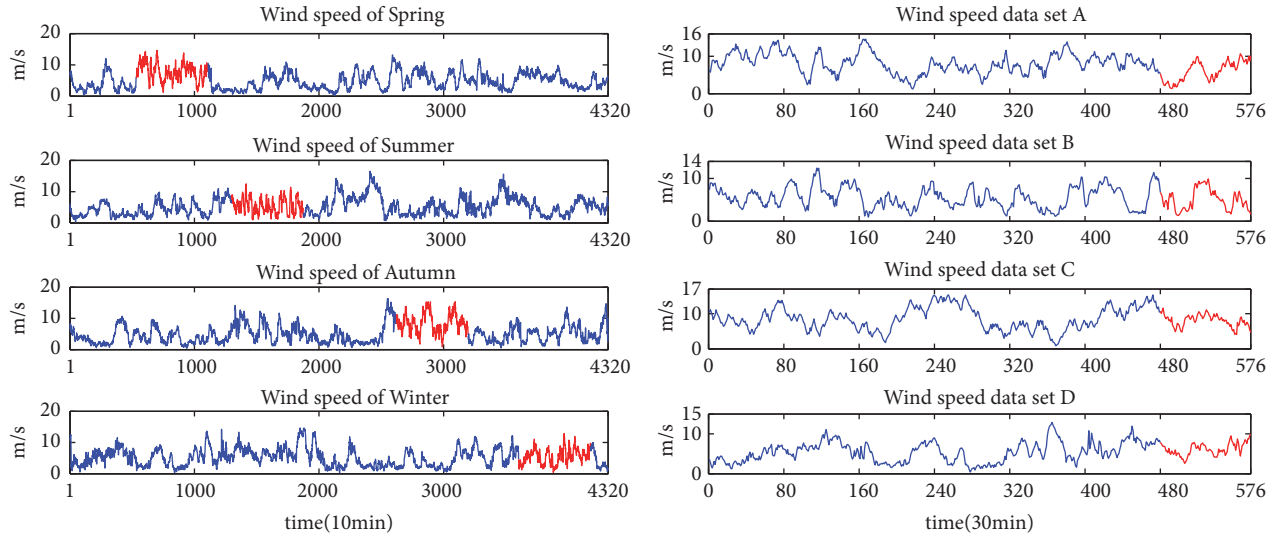
Figure 6: The flowchart of the hybrid EEMD-HGSA-MLLSSVM model.

3.3. Forecasting Performance Evaluation Indices. To verify the prediction performance of the hybrid EEMD-HGSA-MKLSSVM model, three statical indices, namely RMSE, MAE, and MAPE, are utilized to measure the prediction accuracy, and these indices are expressed as (23), (24), and (25).

$$MAE = \frac{1}{N}\sum_{t=1}^{N} |s(t) - \hat{s}(t)|, \tag{24}$$

$$MAPE = \frac{1}{N}\sum_{t=1}^{N} \frac{|s(t) - \hat{s}(t)|}{s(t)} \times 100\%, \tag{25}$$

where $s(i)$, $\hat{s}(t)$, and N are the same meaning as that in (23). It is well known that small statistical index values indicate high forecast accuracy. Among the statistical indices, the MAE reveals the similarity between the predicted and observed



(a) Wind speed of different seasons

(b) Empirical wind speed time series for training and testing

FIGURE 7: Four datasets of empirical wind speed time series (*red* and *blue* lines stand for testing and training data, respectively).

wind speed value, MAPE indicates forecasting percent error at every observation spot, whereas the RMSE measures the overall deviation; therefore, RMSE is adopted as the fitness function in this paper.

To further illustrate the improvement of model I over model II, three improved percentage indices based on the RMSE, MAPE and MAE, are introduced to describe the improvement degree, and these three indices P_{RMSE}, P_{MAE} and P_{MAPE} are expressed as follows [2]:

$$P_{RMSE} = \left(\frac{RMSE1 - RMSE2}{RMSE1}\right) \times 100\%, \quad (26)$$

$$P_{MAE} = \left(\frac{MAE1 - MAE2}{MAE1}\right) \times 100\%, \quad (27)$$

$$P_{MAPE} = \left(\frac{MAPE1 - MAPE2}{MAPE1}\right) \times 100\%. \quad (28)$$

4. Case Study and Forecasting Results

4.1. Wind Speed Data Statistical Description. In this study, the empirical wind speed data were collected and stored in a wind farm located in Anhui province of China to verify the proposed model. The wind speed time series were measured every 10 min from the anemometer at the top of a nacelle, and the wind turbines with 88m are installed at the top of the 300 m mountain (32°28′N, 118°26′E). Accurate half-an-hour to several days ahead wind speed forecasting are very beneficial to routine management of wind farms, and thus the original 10 m wind speed time series are transformed to 30 min by averaging. The developed models in the most studies are tested and evaluated with no more than two sets of wind speed data, and the conclusion that the models are better than the other approaches is generally not convincing enough [26]. Therefore, four sets of wind speed data in 2015 are randomly selected to evaluate the proposed model. The historical wind speed time series of four seasons in 2015 are displayed in the Figure 7(a). The red mark variables in Figure 7(a), which are clearly shown in Figure 7(b), are utilized to train and test the proposed models. The blue variables and red ones in Figure 7(b) are employed to train and test the models, respectively.

The quantity of input samples of LSSVM model affects the wind speed forecasting performance. However, there are not any unified standard and clear definition for section of the input sample quantity [5]. As for the LSSVM which is expert in solving small samples and the suggestion made by Wang et al. [6], four sets of 576 half-hourly wind speed samples in 2015 are selected randomly from Spring, Summer, Autumn, and Winter, respectively. Thereinto, the 1st-480th sampling points are employed to train the proposed model while the subsequent 481th-576th data are applied to test the well-trained model.

The descriptive statistics of the original wind speed times series list in Table 1. It can be obviously seen that the empirical wind speed time series presents the characters of fluctuations and stochastic volatility, fluctuating around 7.80 m/s, 5.48 m/s, 8.47 m/s, and 5.55 m/s, respectively. There is no apparent regularity in the wind speed series.

4.2. Wind Speed Preprocessing Using EEMD. Prior to submitting the original wind speed to the forecasting engine, the signal decomposition technique EEMD is applied to break down the empirical wind speed into several relatively stable IMFs and one *Res* for reducing the forecasting difficulties of HGSA-MKLSSVM. According to [8], the amplitude of the added white noise and the ensemble number in this study are set 0.2 and 100, respectively. As shown in Figure 8, the empirical wind speed time series are broken down into six IMFs and one *Res*. It is obviously observed from the figures that all the components present distinct respective characteristic, frequencies decrease gradually from IMF1 to

Table 1: Statistical description of empirical wind speed (m/s).

data set	NO.	Max	Min	Mean	St.dev.	Median
Data set A						
All samples	576	14.59	1.22	7.80	2.79	7.91
Training data	480	14.59	1.22	8.16	2.68	7.91
Testing data	96	10.68	1.32	6.01	2.64	6.01
Data set B						
All samples	576	12.44	0.94	5.48	2.46	6.69
Training data	480	12.44	0.94	5.58	2.46	6.69
Testing data	96	9.91	1.34	4.99	2.38	5.62
Data set C						
All samples	576	15.34	0.87	8.47	3.08	8.10
Training data	480	15.34	0.87	8.56	3.28	8.10
Testing data	96	11.57	4.08	7.99	1.69	7.82
Data set D						
All samples	576	12.87	0.41	5.55	2.40	6.64
Training data	480	12.87	0.41	5.42	2.52	6.64
Testing data	96	9.91	2.57	6.17	1.56	6.25

Table 2: Lag order of the autoregressive process.

Samples	Original	IMF1	IMF2	IMF3	IMF4	IMF5	IMF6	*Res*
A	7	11	10	15	8	12	7	8
B	8	10	13	14	11	8	7	9
C	7	11	9	10	7	13	9	6
D	6	11	10	8	8	10	9	10

Res, and high-frequency IMF1 and IMF2 reflect the stochastic characteristic of wind speed data, while the signals IMF3 to IMF6 with periodic trend features represent the periodic components of the wind speed data and *Res* are named the trend components.

4.3. Input-Matrix Construction by PACF and BGSA. Before the forecasting processes are carried out by the HGSA-MKLSSVM model, the input variables matrix requires to be determined. The PACF technique is employed to identify the correlations among each corresponding decomposed component for determination of the input combination. The PACF of four sets of the original wind speed and their decomposed components are illustrated in Figure 9. As seen in Figure 9, the PACF values of all wind speed and decomposed components from lags 1 to 30 are calculated. For original wind speed data A, PACF values at lags bigger than 7 are between the 95% confidence lines (the upper and lower lines); thus, the time lag is determined as 7 which can be used to identify the input variables dimension, and the 7 previous continuous wind speed time series contribute the most correlative information to forecast the subsequent wind speed value. The lag order of the original wind speed data and their decomposed components are shown in Table 2. For the original wind speed data A, the input vector for the MKLSSVM model can be described as $X = x_{t-1}, x_{t-2}, \cdots, x_{t-7}$ and the corresponding forecasting value is $Y = x_{t-k}$, where $k(\geq 7)$ is the forecasting horizontal. The other input variables combination for the HGSA-MKLSSVM model determined by PACF values is expressed in Table 3. The feature selection results obtained by BGSA for different subseries are shown in Table 4.

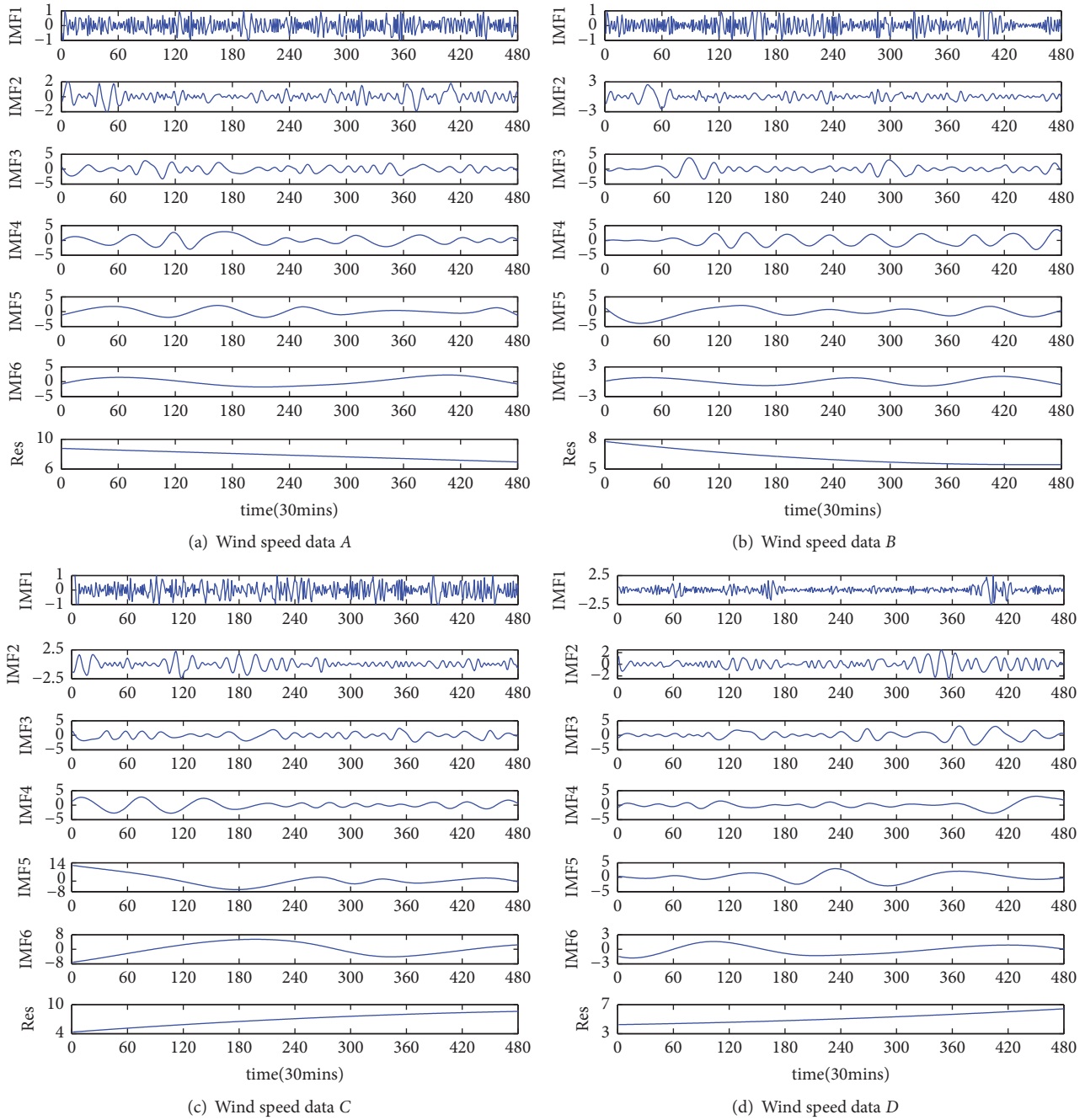
In order to provide better training conditions for MKLSSVM model to enhance wind speed forecasting accuracy, the inputs of each MKLSSVM model are normalized linearly according to (29) into the interval [0 1]. The prediction outcomes of each MKLSSVM model are denormalized through (30) by the contrary process of the corresponding normalization approach.

$$s_i' = \frac{s_i - s_{min}}{s_{max} - s_{min}}, \tag{29}$$

$$\widehat{s}_i = \widehat{s}_i' \left(s_{max} - s_{min}\right) + s_{min}, \tag{30}$$

where s_i, s_{max}, and s_{min} are wind speed at time i and the maximum and the minimum wind speed, respectively. $\widehat{s}_i'$ is the output of MKLSSVM.

4.4. Parameter Settings. After the lag values of each input sample are determined by PACF technique, the LSSVM model based on the weighted multikernel function is ready for wind speed forecasting by the parameter optimization and feature selection. As described in the aforementioned section, the regression performance of MKLSSVM model is greatly affected by the weighted coefficient, penalty parameter, and kernel parameters. These parameters are tuned by the GSA algorithm according to the fitness function with the training wind speed data and they are set in Table 5.



(a) Wind speed data *A*

(b) Wind speed data *B*

(c) Wind speed data *C*

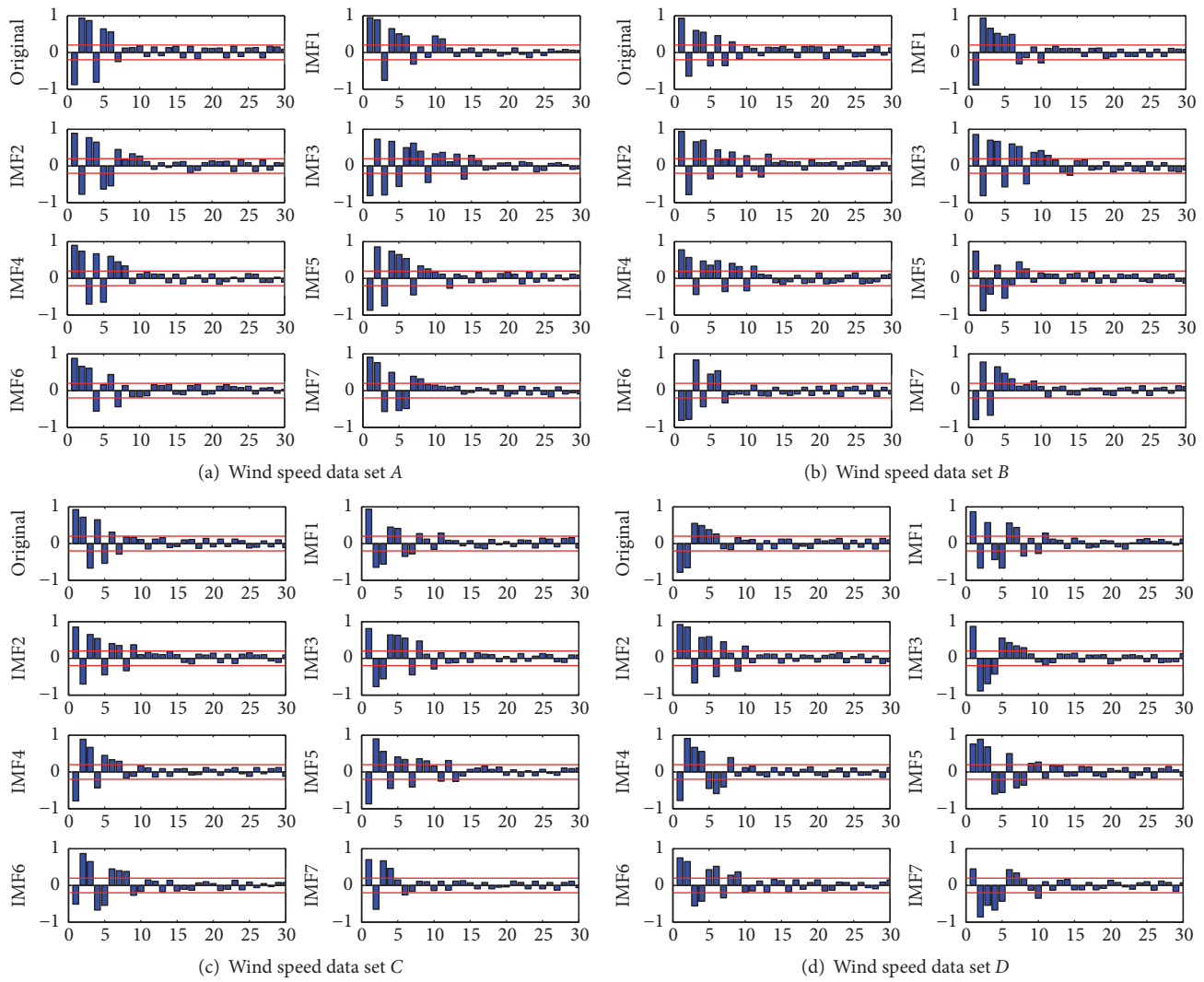
(d) Wind speed data *D*

Figure 8: Wind speed decomposition using EEMD.

4.5. Forecasting Results, Comparison, and Analysis

4.5.1. Forecasting Results and Comparison for LSSVM-Based Model. In an effort to evaluate comprehensively the proposed combined model, the comparisons and analysis of EEMD-HGSA-LSSVM based on RBF, Poly and multikernel function, EMD-HGSA-MKLSSVM, WT-HGSA-MKLSSVM, HGSA-MKLSSVM, and EEMD-MKLSSVM are given and these comparisons are divided into three parts, namely, experiments I, II, and III. The tests with the empirical wind speed samples are carried out in Matlab 2014a environment on windows 7 with 2.4GHz Intel Core i5-4210U and 64bit 8G RAM. The statistical indices and the improved percentage indices are applied to evaluate the forecasting performance.

Experiment I. In this part, the proposed EEMD-HGSA-LSSVM model based on multikernel function is compared with the model based on RBF and Poly kernel functions, respectively, to show the superiority of the multikernel function. Table 6 illustrates the forecasting results obtained by EEMD-HGSA-LSSVM based on multikernel, RBF and Poly functions and the detailed improved percentage indices



(a) Wind speed data set *A*

(b) Wind speed data set *B*

(c) Wind speed data set *C*

(d) Wind speed data set *D*

Figure 9: PACF values of original wind speed and decomposed components (95% confidence interval line).

between the proposed model and the compared models are listed in Table 7 and the forecasting curves are shown in Figure 10. As seen from the tables, the RMSE errors of the model based on the weighted multikernel function are 0.2787 m/s, 0.2856 m/s, 0.2691 m/s, and 0.2903 m/s for wind speed data *A*, *B*, *C*, and *D*, respectively, which are smallest. The improved percentage P_{RMSE} between the models based on the weighted multikernel function *vs*. based on the RBF function and the Poly function are 2.48% and 5.75% for data *A*, 2.79% and 5.59% for data *B*, 3.93% and 7.43% for data *C*, and 3.04% and 6.59% for data *D*, respectively. In the scatter diagram of the forecasting values versus the empirical measured wind speed time series displayed in Figure 10, the dashed straight red lines represent the actual original wind speed data which are the same as the forecasting results, which means that the forecasting errors are bigger if the forecasting points get further away from the line. The whole forecasting points obtained by EEMD-HGSA-MKLSSVM model are much closer to the line. From the statistical indices in Table 6 and the improved percentage in Table 7, the EEMD-HGSA-LSSVM based on the weighted multikernel function performs best although the models based on RBF function and Poly function perform slightly worse than that based on the weighted multikernel function in terms of performance indices for all wind speed data sets.

Remark. Overall, LSSVM with multikernel function exhibits the highest forecasting accuracy whereas LSSVM with RBF kernel function obtains the worst forecasting performance in that there are high fluctuation and randomness in wind speed. The combination of individual advantages of the RBF kernel function and Poly kernel function by weight coefficient can catch the nonlinear character in the wind speed.

Experiment II. In this part, comparisons involving EEMD-HGSA-MKLSSVM, EMD-HGSA-MKLSSVM, WT-HGSA-MKLSSVM, and HGSA-MKLSSVM without signal preprocessing method are employed to illustrate the indispensability of the wind speed preprocessing technique in the forecasting model and superiority of EEMD approach. The

Table 3: Inputs combinations of wind speed and their decomposed components for the HGSA-MKLSSVM model.

Data set	Time series	input variable combination
A	Original	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}$
	IMF1	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}$
	IMF2	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}$
	IMF3	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}, x_{t-12}, x_{t-13}, x_{t-14}, x_{t-15}$
	IMF4	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8},$
	IMF5	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}, x_{t-12}$
	IMF6	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}$
	Res	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}$
B	Original	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}$
	IMF1	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}$
	IMF2	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}, x_{t-12}, x_{t-13}$
	IMF3	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}, x_{t-12}, x_{t-13}, x_{t-14}$
	IMF4	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}$
	IMF5	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}$
	IMF6	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}$
	Res	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}$
C	Original	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}$
	IMF1	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}$
	IMF2	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}$
	IMF3	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}$
	IMF4	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}$
	IMF5	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}, x_{t-12}, x_{t-13}$
	IMF6	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-8}, x_{t-9}$
	Res	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}$
D	Original	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}$
	IMF1	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}$
	IMF2	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}$
	IMF3	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}$
	IMF4	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}$
	IMF5	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}$
	IMF6	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}$
	Res	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}$

Table 4: Feature selection results obtained by HGSA algorithm.

Data set	Time series	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	Original	1	1	1	0	1	0	1								
	IMF1	1	1	0	1	1	1	0	1	1	0	1				
	IMF2	1	1	1	1	0	0	1	1	1	1					
	IMF3	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1
	IMF4	1	1	1	0	1	1	0	1							
	IMF5	1	0	1	0	1	1	1	0	1	1	1	1			
	IMF6	1	1	0	1	1	0	1								
	Res	1	1	0	1	1	1	1	1							
B	Original	1	0	1	1	1	1	1	1							
	IMF1	1	0	1	1	1	1	1	1	0	1					
	IMF2	1	1	0	1	0	1	1	1	1	1	1	1	0		
	IMF3	1	0	1	1	1	1	1	1	0	0	1	1	1	1	
	IMF4	1	1	0	1	1	0	1	1	1	1	1				
	IMF5	1	1	1	0	1	1	0	1							
	IMF6	1	1	0	1	1	1	1								
	Res	1	0	1	1	1	1	0	1	1						
C	Original	1	1	1	0	1	1	1								
	IMF1	1	0	1	1	0	1	1	0	1	1	1				
	IMF2	1	0	1	1	1	1	0	1	1						
	IMF3	1	1	1	0	1	1	0	1	1	1					
	IMF4	1	1	0	1	1	1	1								
	IMF5	1	1	1	1	0	1	1	1	1	1	0	0	1		
	IMF6	1	1	0	1	1	1	1	0	1						
	Res	1	0	1	1	1	1									
D	Original	1	1	1	0	1	1									
	IMF1	1	0	1	1	1	1	0	1	1	0	1				
	IMF2	1	0	1	1	0	1	1	0	1	1					
	IMF3	1	0	1	1	1	0	1	1							
	IMF4	1	1	1	0	1	1	1	1							
	IMF5	1	1	0	1	1	1	1	1	1	1					
	IMF6	1	1	0	1	0	1	1	1	1						
	Res	1	0	1	1	1	1	1	1	0	1					

[1] 1 stands for *selection*, 0 represents *discard*.

Table 5: Parameters setup in HGSA-MKLSSVM model.

Parameters	Value
Maximum iteration number	100
Particle number	30
Dimension number	5
Gravitational constant G_0	2
Kernel parameter γ	$(0, 200)$
Kernel parameter δ	$(0, 20)$
Kernel parameter r	$(0, 20)$
Kernel parameter d	$(0, 20)$
Weighted coefficient μ	$[0, 1]$

optimal parameters in the HGSA-MKLSSVM without signal processing are illustrated in Table 8. The forecasting curves, forecasting indices and the detailed improved percentage indices, are shown in Figure 11 and Tables 9 and 10, respectively. As seen from the figures and tables, the proposed EEMD-based model has better forecasting performance than the corresponding WT-HGSA-MKLSSVM and EMD-HGSA-MKLSSVM models and HGSA-MKLSSVM model. In these models, HGSA-MKLSSVM without wind speed decomposition performs the worst from the statistical indices viewpoint. For example, compared with WT-based, EMD-based, and without wind speed data decomposition based HGSA-MKLSSVM, RMSE of the proposed EEMD-based model are cut by 0.1192 m/s, 0.1056 m/s, and 0.2332 m/s for data set *A*, 0.1312 m/s, 0.1173 m/s, and 0.255 m/s for data set *B*, 0.1399 m/s, 0.1191 m/s, and 0.249 m/s for data set *C*, 0.1122 m/s, 0.1005 m/s, and 0.2503 m/s for data set D.

Remark. It can be obviously seen from the histograms that the forecasting accuracy of these models from the highest to the lowest are HGSA-MKLSSVM with EEMD, EMD, and

Table 6: Forecasting results by EEMD-HGSA-LSSVM based on different kernel function.

Data set	Models	RMSE (m/s)	MAPE (%)	MAE (m/s)
A	Model 1	0.2858	5.6439	0.2597
	Model 2	0.2957	5.8311	0.2673
	Model 3	0.2787	5.457	0.2558
B	Model 1	0.2938	6.8854	0.274
	Model 2	0.3025	7.4624	0.2786
	Model 3	0.2856	6.4077	0.2538
C	Model 1	0.2801	3.433	0.2607
	Model 2	0.2907	3.382	0.2546
	Model 3	0.2691	3.232	0.2392
D	Model 1	0.2994	4.9216	0.2819
	Model 2	0.3108	4.9586	0.2865
	Model 3	0.2903	4.5088	0.2607

[1] Models 1, 2, and 3 denote EEMD-HGSA-LSSVM based on RBF, Poly, and multikernel function, respectively.

Table 7: Improved percentage obtained by EEMD-HGSA-MKLSSVM over that based on RBF or Poly kernel function (%).

Data set	Contrast	P_{RMSE}	P_{MAPE}	P_{MAE}
A	Model 1 *Vs*. Model 3	2.48	3.31	1.51
	Model 2 *Vs*. Model 3	5.75	6.42	4.30
B	Model 1 *Vs*. Model 3	2.79	6.94	7.37
	Model 2 *Vs*. Model 3	5.59	14.13	8.91
C	Model 1 *Vs*. Model 3	3.93	5.85	8.25
	Model 2 *Vs*. Model 3	7.43	4.43	6.05
D	Model 1 *Vs*. Model 3	3.04	8.39	7.52
	Model 2 *Vs*. Model 3	6.59	9.07	9.01

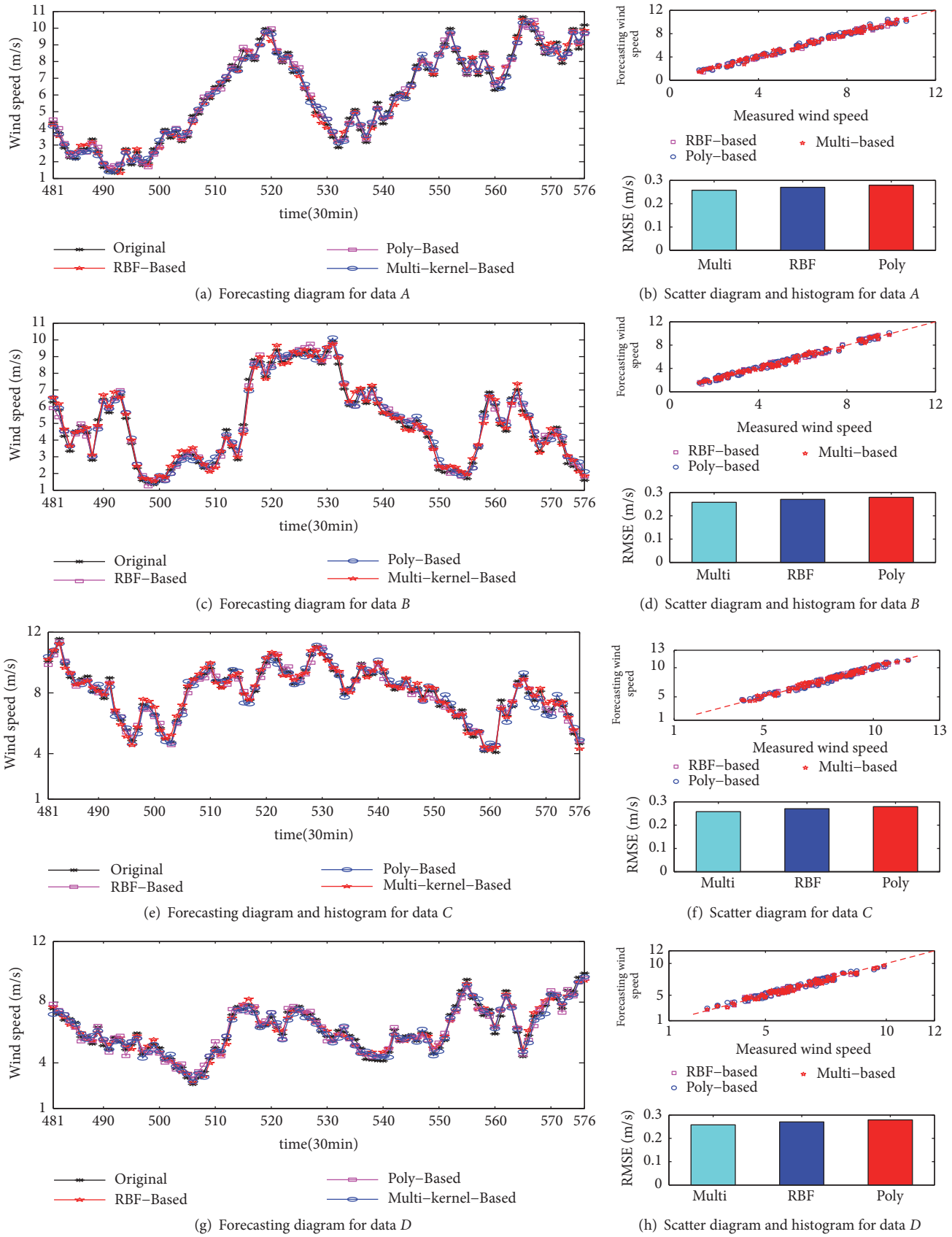
[1] Models 1, 2, and 3 denote EEMD-HGSA-LSSVM based on RBF, Poly and multikernel function, respectively.

WT and without signal decomposition. Compared with other models, the scatter points of the forecasting approach EEMD-HGSA-MKLSSVM distribute closest to the regression line. The preprocessing technique EEMD is more effective than WT, EMD signal decomposition when applied in the wind speed decomposition. Besides, the forecasting accuracy of the HGSA-MKLSSVM model can be improved greatly through the signal decomposition technique for all wind speed data sets; therefore, wind speed decomposition technique is indispensable in the application wind speed forecasting in that wind speed time series exhibit random and highly fluctuant, and wind speed decomposition technique makes wind speed data decomposed into relatively stable components which reduce the prediction difficulties of the forecasting engine.

Experiment III. In this part, comparisons between the proposed model with EEMD-GSA-MKLSSVM and EEMD-MKLSSVM are carried out to illustrate the necessity of feature selection and parameter optimization in the forecasting model. In the EEMD-MKLSSVM model, $\mu = 0.5$ and the other kernel parameters are set according to [21]. The forecasting curves, forecasting indices and the detailed improved percentage indices, are shown in Figure 12 and Tables 11 and 12, respectively. As seen from the figures and tables, compared with EEMD-GSA-MKLSSVM and EEMD-MKLSSVM, the RMSE errors of EEMD-HGSA-MKLSSVM are reduced by 0.1155 m/s and 0.2012 m/s for data set *A*, 0.1113 m/s and 0.2295 m/s for data set *B*, 0.1227 m/s and 0.2163 m/s for data set *C*, 0.1026 m/s and 0.2192 m/s for data set *D*, respectively. From the histogram in the subplot of Figure 12, the forecasting accuracy of the models ranks from low to high as EEMD-MKLSSVM, EEMD-GSA-MKLSSVM, and EEMD-HGSA-MKLSSVM for all wind speed data sets. The most forecasting points of EEMD-MKLSSVM locate farthest from the regression line, while those of EEMD-HGSA-MKLSSVM are closest to the line.

Remark. It illustrates that the application of feature selection and parameter optimization in the forecasting model is helpful to wind speed prediction accuracy; the EEMD-MKLSSVM model works worse than both EEMD-GSA-MKLSSVM model and EEMD-HGSA-MKLSSVM model, while EEMD-GSA-MKLSSVM model without feature selection performs also worse than EEMD-HGSA-MKLSSVM model in that the direct application of MKLSSVM in wind speed forecasting without parameters optimization by GSA algorithm may result in overfitting or trapping into local optima, and the redundant and illusive components within each subseries are not identified by feature selection through BGSA algorithm.

4.5.2. Compared with Other Forecasting Models. In this section, Persistence method, generally adopted as a benchmark approach to validate a new developed forecasting method



(a) Forecasting diagram for data *A*

(b) Scatter diagram and histogram for data *A*

(c) Forecasting diagram for data *B*

(d) Scatter diagram and histogram for data *B*

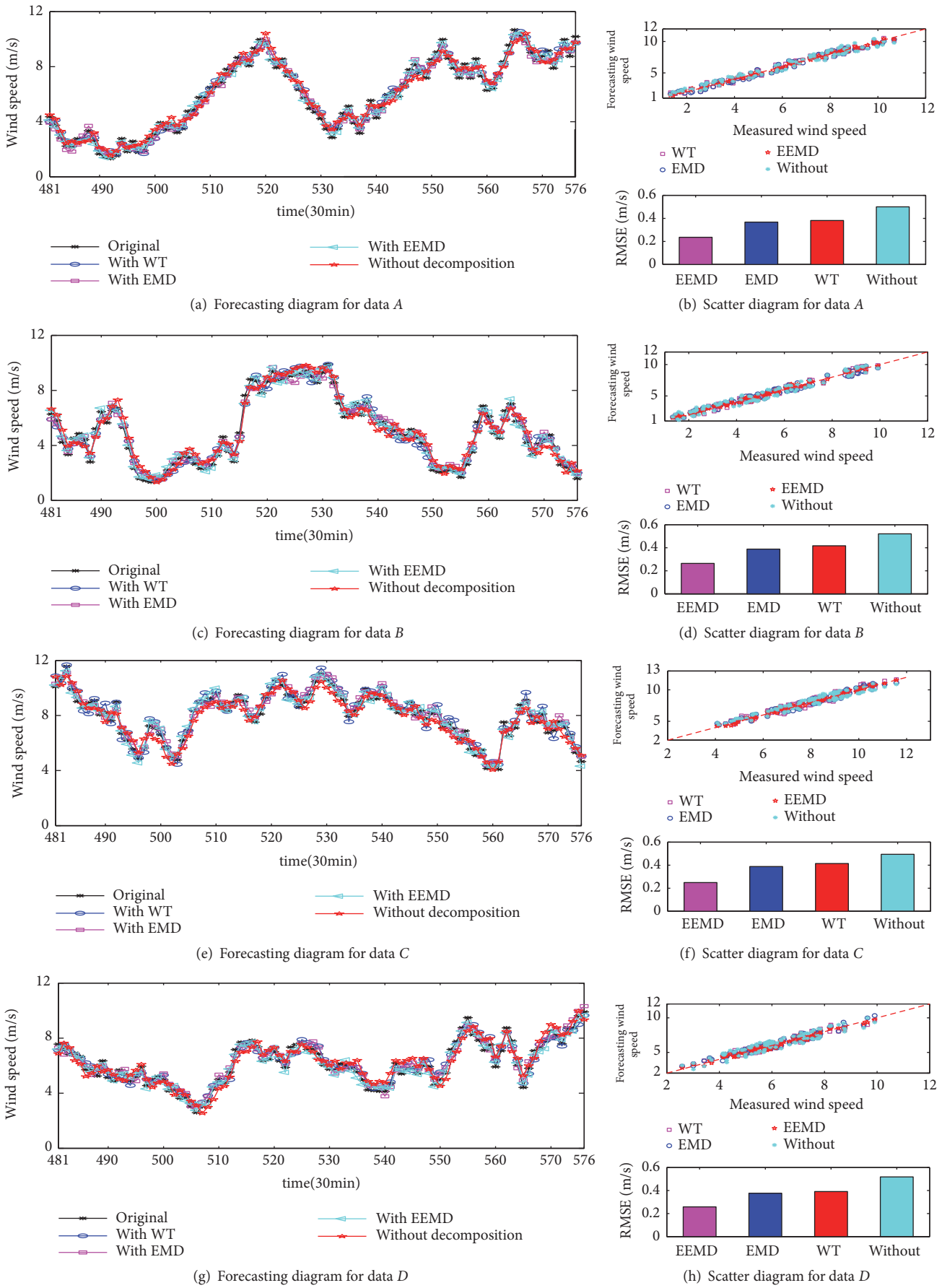
(e) Forecasting diagram and histogram for data *C*

(f) Scatter diagram for data *C*

(g) Forecasting diagram for data *D*

(h) Scatter diagram and histogram for data *D*

Figure 10: Wind speed forecasting results obtained by EEMD-based HGSA-LSSVM with different kernel function.



(a) Forecasting diagram for data *A*

(b) Scatter diagram for data *A*

(c) Forecasting diagram for data *B*

(d) Scatter diagram for data *B*

(e) Forecasting diagram for data *C*

(f) Scatter diagram for data *C*

(g) Forecasting diagram for data *D*

(h) Scatter diagram for data *D*

Figure 11: Wind speed forecasting results obtained by HGSA-LSSVM with different signal preprocessing.

Table 8: Tuned parameters of MKLSSVM by HGSA directly for original wind speed.

Data set	γ	δ	μ	r	d
A	8.0192	0.6309	0.2935	0.9763	4.0871
B	123.7445	3.4637	0.5853	1.1434	8.3528
C	42.4292	2.3582	0.7329	1.8739	9.3567
D	33.6749	1.1526	0.4323	2.1457	7.4737

Table 9: Forecasting results by EEMD-HGSA-LSSVM based on different decomposition method.

Data set	Models	RMSE (m/s)	MAPE (%)	MAE (m/s)
A	Model 4	0.5119	9.9989	0.4785
	Model 5	0.3979	7.8679	0.3514
	Model 6	0.3843	7.883	0.3596
	Model 7	0.2787	5.457	0.2558
B	Model 4	0.5406	13.3218	0.5138
	Model 5	0.4168	9.9403	0.3927
	Model 6	0.4029	9.658	0.3824
	Model 7	0.2856	6.4077	0.2538
C	Model 4	0.5181	6.5663	0.4985
	Model 5	0.409	4.5731	0.3456
	Model 6	0.3882	4.8105	0.3642
	Model 7	0.2691	3.232	0.2392
D	Model 4	0.5406	8.7821	0.5064
	Model 5	0.4025	6.3819	0.3701
	Model 6	0.3908	6.3498	0.3687
	Model 7	0.2903	4.5088	0.2607

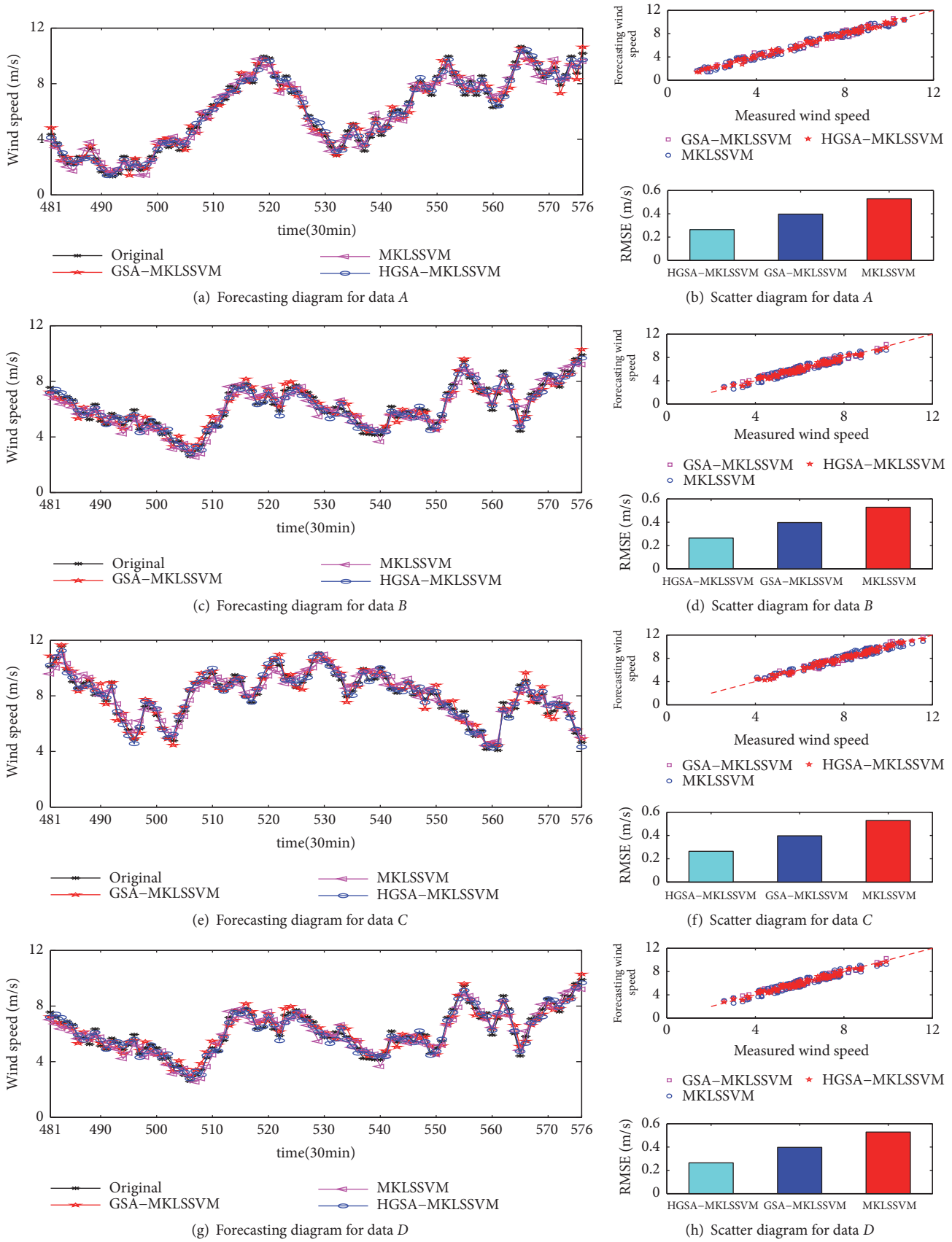
[1] Model 4 denotes HGSA-MKLSSVM.
[2] Models 5, 6, and 7 denote HGSA-MKLSSVM with WT, EMD, and EEMD, respectively.

[1], is also taken as the benchmark approach to see how much the EEMD-HGSA-MKLSSVM approach improves the prediction performance. Moreover, the forecasting performances of the EEMD-based ELM and SVM with parameter optimization and feature selection by HGSA algorithm are compared to further evaluate the effectiveness of the proposed forecasting model in terms of the three statistical indices. In the test, the hidden nodes number in ELM is determined by the grid search(GS) method whose searching range and grid step are set as [1 200] and 1, respectively. Tables 13 and 14 list forecasting statistical indices and improved percentage, respectively. As seen from the tables, compared with Persistence, EEMD-HGSA-ELM, and EEMD-HGSA-SVM, the RMSE values of the proposed model are cut by 0.4517 m/s, 0.0712 m/s, and 0.0827 m/s for data set *A*, 0.4664 m/s, 0.1042 m/s, and 0.1238 m/s for data set *B*, 0.4925 m/s, 0.0984 m/s, and 0.1148 m/s for data set *C*, 0.4506 m/s, 0.0844 m/s, and 0.0929 m/s for data set *D*, respectively.

Remark. The proposed EEMD-HGSA-MKLSSVM model not only performs better than EEMD-HSA-ELM and EEMD-HGSA-SVM models, but also it obtains remarkably higher forecasting accuracy than benchmark method Persistence. In Persistence approach, the current sample at time t is utilized to predict the future time $t + \Delta t$, then the $t + \Delta t$ value as the current observation is employed to forecast the next data; therefore, it can be easily established in the wind speed forecasting. Compared with Persistence approach, the developed EEMD-HGSA-MKLSSVM model is more complicated to accomplish the total prediction process; however, thanks to the advance of computer technology, this is acceptable. Moreover, the superiority of the proposed model over EEMD-HGSA-SVM can explain that the basic idea of SVM is to map the input samples into high-dimensional space through nonlinear function, while the basic working mechanism of LSSVM model is that the quadratic programming problems are converted to solve linear equations, thus enhancing its regressive performance. Therefore, the discussion and analysis by comparison with other forecasting models can provide sufficient evidence that the proposed hybrid model with feature selection and parameter optimization is an excellent approach for short-term wind speed prediction.

5. Conclusion

In this article, a compound MKLSSVM model optimized by HGSA algorithm integrated with signal decomposition technique EEMD, namely, EEMD-HGSA-MKLSSVM, is proposed for short-term wind speed forecasting. Four sets of mean half-hour wind speed selected randomly from the historical wind speed data in 2015 collected from a wind farm located in Anhui of China are utilized as case studies to evaluate the forecasting performance of EEMD-HGSA-MKLSSVM model. From the comparison and analysis



(a) Forecasting diagram for data *A*

(b) Scatter diagram for data *A*

(c) Forecasting diagram for data *B*

(d) Scatter diagram for data *B*

(e) Forecasting diagram for data *C*

(f) Scatter diagram for data *C*

(g) Forecasting diagram for data *D*

(h) Scatter diagram for data *D*

Figure 12: Wind speed forecasting results obtained by EEMD-based models.

Table 10: Improved percentage obtained by EEMD-HGSA-MKLSSVM over other models with different decomposition method (%).

Data set	Contrast	P_{RMSE}	P_{MAPE}	P_{MAE}
A	Model 4 *Vs.* Model 7	45.56	45.42	46.54
	Model 5 *Vs.* Model 7	29.96	30.64	27.21
	Model 6 *Vs.* Model 7	27.48	30.78	28.87
B	Model 4 *Vs.* Model 7	47.17	51.90	50.60
	Model 5 *Vs.* Model 7	31.48	35.54	35.37
	Model 6 *Vs.* Model 7	29.11	33.65	33.63
C	Model 4 *Vs.* Model 7	48.06	50.78	52.02
	Model 5 *Vs.* Model 7	34.21	29.33	30.79
	Model 6 *Vs.* Model 7	30.68	32.81	34.32
D	Model 4 *Vs.* Model 7	46.30	48.66	48.52
	Model 5 *Vs.* Model 7	27.88	29.35	29.56
	Model 6 *Vs.* Model 7	25.72	28.99	29.29

[1] Model 4 denotes HGSA-LSSVM.
[2] Models 5, 6, and 7 denote HGSA-MKLSSVM with WT, EMD, and EEMD, respectively.

Table 11: Forecasting results by EEMD-MKLSSVM based model.

Data set	Models	RMSE (m/s)	MAPE (%)	MAE (m/s)
A	Model 8	0.4799	9.9012	0.4557
	Model 9	0.3942	7.0633	0.3397
	Model 10	0.2787	5.457	0.2558
B	Model 8	0.5151	12.4774	0.4844
	Model 9	0.3969	9.4711	0.3527
	Model 10	0.2856	6.4077	0.2538
C	Model 8	0.4854	5.9519	0.4591
	Model 9	0.3918	4.365	0.329
	Model 10	0.2691	3.232	0.2392
D	Model 8	0.5095	8.5347	0.4884
	Model 9	0.3929	6.479	0.3673
	Model 10	0.2903	4.5088	0.2607

[1] Model 8 denotes EEMD-MKLSSVM.
[2] Model 9 denotes EEMD-GSA-MKLSSVM.
[3] Model 10 denotes EEMD-HGSA-MKLSSVM.

carried out in the previous sections, some conclusions can be drawn as follows:

(i) Considering that EEMD is an effective approach to decompose and analyze the nonlinear and nonstationary signal, we adopt it as the wind speed data preprocessing tool in the hybrid models. Correspondingly, the EEMD-based forecasting model is trained and tested with the four sets of empirical wind speed data. Compared with HGSA-MKLSSVM model without signal preprocessing, the EEMD-based HGSA-MKLSSVM model has obvious improvement in the forecasting results, thus, wind speed decomposition is indispensable in the wind speed forecasting. The forecasting results show that the EEMD-based HGSA-MKLSSVM model yields better forecasting accuracy than the corresponding EMD-based and WT-based models; thus signal decomposition technique EEMD is suitable in this hybrid forecasting model.

(ii) The EEMD-HGSA-LSSVM based on multikernel function has better forecasting results than that based on RBF kernel function or Poly kernel function in that the multikernel function takes advantages of individual merits of RBF and Poly kernel functions by optimal weighted coefficient.

(iii) The hybrid algorithm HGSA utilizes the respective advantages of BGSA and RGSA to realize the feature selection and parameter optimization simultaneously. Compared with the EEMD-GSA-MKLSSVM without feature selection, the proposed model obtains smaller RMSE for all the data sets, which means that BGSA selects the useful candidates for the forecasting engine.

(iv) The proposed model outperforms the EEMD-HGSA-ELM and EEMD-HGSA-SVM. Especially, the improvements obtained by EEMD-based HGSA-MKLSSVM model over the benchmark Persistence model in terms of improved percentage RMSE are

Table 12: Improved percentage obtained by EEMD-HGSA-MKLSSVM over other EEMD-MKLSSVM based models (%).

Data set	Contrast	P_{RMSE}	P_{MAPE}	P_{MAE}
A	Model 8 *Vs.* Model 10	41.93	44.89	43.87
	Model 9 *Vs.* Model 10	29.29	22.74	24.69
B	Model 8 *Vs.* Model 10	44.55	48.64	47.60
	Model 9 *Vs.* Model 10	28.04	32.34	28.04
C	Model 8 *Vs.* Model 10	44.56	45.69	47.89
	Model 9 *Vs.* Model 10	31.32	25.96	27.29
D	Model 8 *Vs.* Model 10	43.05	47.17	46.62
	Model 9 *Vs.* Model 10	26.11	30.41	29.02

[1] Model 8 denotes EEMD-MKLSSVM.
[2] Model 9 denotes EEMD-GSA-MKLSSVM.
[3] Model 10 denotes EEMD-HGSA-MKLSSVM.

Table 13: Forecasting results by different forecasting model.

Data set	Models	RMSE (m/s)	MAPE (%)	MAE (m/s)
A	Model 11	0.7304	15.3973	0.717
	Model 12	0.3499	6.2955	0.2929
	Model 13	0.3614	7.5089	0.3353
	Model 14	0.2787	5.457	0.2558
B	Model 11	0.752	18.7638	0.7344
	Model 12	0.3898	9.2387	0.3638
	Model 13	0.4094	10.4421	0.3869
	Model 14	0.2856	6.4077	0.2538
C	Model 11	0.7616	9.9466	0.7445
	Model 12	0.3675	4.642	0.3472
	Model 13	0.3839	4.6887	0.355
	Model 14	0.2691	3.232	0.2392
D	Model 11	0.7409	12.5176	0.7193
	Model 12	0.3747	6.1382	0.353
	Model 13	0.3832	6.3973	0.3623
	Model 14	0.2903	4.5088	0.2607

[1] Model 11 denotes Persistence.
[2] Model 12 denotes EEMD-HGSA-ELM.
[3] Model 13 denotes EEMD-HGSA-SVM.
[4] Model 14 denotes EEMD-HGSA-MKLSSVM.

Table 14: Improved percentage obtained by EEMD-HGSA-MKLSSVM over other models (%).

Data set	Contrast	P_{RMSE}	P_{MAPE}	P_{MAE}
A	Model 11 *Vs.* Model 14	61.84	64.56	64.33
	Model 12 *Vs.* Model 14	20.34	13.32	12.67
	Model 13 *Vs.* Model 14	22.88	27.33	23.71
B	Model 11 *Vs.* Model 14	62.02	65.85	65.44
	Model 12 *Vs.* Model 14	26.73	30.64	30.24
	Model 13 *Vs.* Model 14	30.24	38.64	34.40
C	Model 11 *Vs.* Model 14	64.67	67.51	67.87
	Model 12 *Vs.* Model 14	26.78	30.37	31.11
	Model 13 *Vs.* Model 14	29.91	31.07	32.62
D	Model 11 *Vs.* Model 14	60.82	63.98	63.76
	Model 12 *Vs.* Model 14	22.52	26.55	26.15
	Model 13 *Vs.* Model 14	24.24	29.52	28.04

[1] Model 11 denotes Persistence.
[2] Model 12 denotes EEMD-HGSA-ELM.
[3] Model 13 denotes EEMD-HGSA-SVM.
[4] Model 14 denotes EEMD-HGSA-MKLSSVM.

about 61.84%, 62.02%, 64.67%, and 60.82% for the corresponding wind speed data sets *A*, *B*, *C*, and *D*, respectively.

Therefore, the proposed model EEMD-HGSA-MKLSSVM is an effective short-term wind speed prediction approach. For further studies, this hybrid model will be utilized for other wind farms, and some environmental and climate information should be taken into consideration as potential input samples.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

All the authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Projects of Science and Technology Commission of Shanghai Municipality under Grant nos. 15JC1401900 and 17511107002, Natural Capital Project of Anhui Province under 1501021015, National Natural Science Foundation of China under 61803001, the Open Research Fund of Wanjiang Collaborative Innovation Center for High-End Manufacturing Equipment, and Anhui Polytechnic University under Grant no. GCKJ2018010.

References

[1] P. Du, J. Wang, Z. Guo, and W. Yang, "Research and application of a novel hybrid forecasting system based on multi-objective optimization for wind speed forecasting," *Energy Conversion and Management*, vol. 150, pp. 90–107, 2017.

[2] C. Zhang, J. Zhou, C. Li, W. Fu, and T. Peng, "A compound structure of ELM based on feature selection and parameter optimization using hybrid backtracking search algorithm for wind speed forecasting," *Energy Conversion and Management*, vol. 143, pp. 360–376, 2017.

[3] S. Salcedo-Sanz, A. Pastor-Sánchez, L. Prieto, A. Blanco-Aguilera, and R. García-Herrera, "Feature selection in wind speed prediction systems based on a hybrid coral reefs optimization - extreme learning machine approach," *Energy Conversion and Management*, vol. 87, no. 7, pp. 10–18, 2014.

[4] L. Xiao, F. Qian, and W. Shao, "Multi-step wind speed forecasting based on a hybrid forecasting architecture and an improved bat algorithm," *Energy Conversion and Management*, vol. 143, pp. 410–430, 2017.

[5] J. M. Hu, J. Z. Wang, and K. L. Ma, "A hybrid technique for short-term wind speed prediction," *Energy*, vol. 81, no. 1, pp. 563–574, 2015.

[6] J. Wang and J. Hu, "A robust combination approach for short-term wind speed forecasting and analysis—combination of the ARIMA (Autoregressive Integrated Moving Average), ELM (Extreme Learning Machine), SVM (Support Vector Machine) and LSSVM (Least Square SVM) forecasts using a GPR (Gaussian Process Regression) model," *Energy*, vol. 93, pp. 41–56, 2015.

[7] D. Wang, H. Luo, O. Grunder, and Y. Lin, "Multi-step ahead wind speed forecasting using an improved wavelet neural network combining variational mode decomposition and phase space reconstruction," *Journal of Renewable Energy*, vol. 113, pp. 1345–1358, 2017.

[8] Y. Jiang and G. Huang, "Short-term wind speed prediction: Hybrid of ensemble empirical mode decomposition, feature selection and error correction," *Energy Conversion and Management*, vol. 144, pp. 340–350, 2017.

[9] C. Ren, N. An, J. Wang, L. Li, B. Hu, and D. Shang, "Optimal parameters selection for BP neural network based on particle swarm optimization: a case study of wind speed forecasting," *Knowledge-Based Systems*, vol. 56, pp. 226–239, 2014.

[10] Z. H. Guo, J. Zhao, W. Y. Zhang, and J. Z. Wang, "A corrected hybrid approach for wind speed prediction in Hexi Corridor of China," *Energy*, vol. 36, no. 3, pp. 1668–1679, 2011.

[11] J. Wang, H. Jiang, B. Han, and Q. Zhou, "An Experimental Investigation of FNN Model for Wind Speed Forecasting Using EEMD and CS," *Mathematical Problems in Engineering*, vol. 2015, Article ID 464153, 13 pages, 2015.

[12] S. Salcedo-Sanz, E. G. Ortiz-García, Á. M. Pérez-Bellido, A. Portilla-Figueras, and L. Prieto, "Short term wind speed prediction based on evolutionary support vector regression algorithms," *Expert Systems with Applications*, vol. 38, no. 9, pp. 4052–4057, 2011.

[13] D. Liu, D. X. Niu, H. Wang, and L. L. Fan, "Short-term wind speed forecasting using wavelet transform and support vector machines optimized by genetic algorithm," *Journal of Renewable Energy*, vol. 62, no. 9, pp. 592–597, 2014.

[14] M. A. Mohandes, T. O. Halawani, S. Rehman, and A. A. Hussain, "Support vector machines for wind speed prediction," *Journal of Renewable Energy*, vol. 29, no. 11, pp. 939–947, 2004.

[15] J. Z. Wang, J. M. Hu, K. L. Ma, and Y. X. Zhang, "A self-adaptive hybrid approach for wind speed forecasting," *Journal of Renewable Energy*, vol. 78, no. 1, pp. 374–385, 2015.

[16] A. Meng, J. Ge, H. Yin, and S. Chen, "Wind speed forecasting based on wavelet packet decomposition and artificial neural networks trained by crisscross optimization algorithm," *Energy Conversion and Management*, vol. 114, no. 2, pp. 75–88, 2016.

[17] S. Wang, N. Zhang, L. Wu, and Y. Wang, "Wind speed forecasting based on the hybrid ensemble empirical mode decomposition and GA-BP neural network method," *Journal of Renewable Energy*, vol. 94, no. 4, pp. 629–636, 2016.

[18] A. A. Abdoos, "A new intelligent method based on combination of VMD and ELM for short term wind power forecasting," *Neurocomputing*, vol. 203, no. 5, pp. 111–120, 2016.

[19] J. M. Hu, J. Z. Wang, and G. W. Zeng, "A hybrid forecasting approach applied to wind speed time series," *Journal of Renewable Energy*, vol. 60, no. 6, pp. 185–194, 2013.

[20] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[21] X. H. Yuan, C. Chen, Y. B. Yuan, Y. H. Huang, and Q. X. Tan, "Short-term wind power prediction based on LSSVM-GSA model," *Energy Conversion and Management*, vol. 101, pp. 393–401, 2015.

[22] Chen Wang, Jie Wu, Jianzhou Wang, and Zhongjin Hu, "Short-Term Wind Speed Forecasting Using the Data Processing Approach and the Support Vector Machine Model Optimized

by the Improved Cuckoo Search Parameter Estimation Algorithm," *Mathematical Problems in Engineering*, vol. 2016, Article ID 4896854, 17 pages, 2016.

[23] H. Shayeghi and H. Ghasemi, "Day-ahead electricity prices forecasting by a modified CGSA technique and hybrid WT in LSSVM based scheme," *Energy Conversion and Management*, vol. 174, no. 7, pp. 482–491, 2015.

[24] Y. Wang and L. Wu, "On practical challenges of decomposition-based hybrid forecasting algorithms for wind speed and solar irradiation," *Energy*, vol. 112, no. 6, pp. 208–220, 2016.

[25] Z. Yang, K. Li, Q. Niu, and Y. Xue, "A novel parallel-series hybrid meta-heuristic method for solving a hybrid unit commitment problem," *Knowledge-Based Systems*, vol. 134, no. 7, pp. 13–30, 2017.

[26] C. Feng, M. Cui, B.-M. Hodge, and J. Zhang, "A data-driven multi-model methodology with deep feature selection for short-term wind forecasting," *Applied Energy*, vol. 190, pp. 1245–1257, 2017.

[27] S. Salcedo-Sanz, A. Pastor-Sánchez, L. Prieto, A. Blanco-Aguilera, and R. García-Herrera, "Feature selection in wind speed prediction systems based on a hybrid coral reefs optimization - extreme learning machine approach," *Energy Conversion and Management*, vol. 87, no. 6, pp. 10–18, 2014.

[28] M. Luo, C. Li, X. Zhang, R. Li, and X. An, "Compound feature selection and parameter optimization of ELM for fault diagnosis of rolling element bearings," *ISA Transactions*, vol. 65, no. 8, pp. 556–566, 2016.

[29] G. Zhang, Y. Wu, and Y. Liu, "An advanced wind speed multi-step ahead forecasting approach with characteristic component analysis," *Journal of Renewable and Sustainable Energy*, vol. 6, no. 7, pp. 1–14, 2014.

[30] A. Zendehboudi, "Implementation of GA-LSSVM modelling approach for estimating the performance of solid desiccant wheels," *Energy Conversion and Management*, vol. 127, no. 9, pp. 245–255, 2016.

[31] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 1, pp. 2232–2248, 2009.

[32] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "BGSA: binary gravitational search algorithm," *Natural Computing*, vol. 9, no. 9, pp. 727–745, 2010.

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 4653648, 8 pages
https://doi.org/10.1155/2018/4653648

Research Article

A Negotiation Optimization Strategy of Collaborative Procurement with Supply Chain Based on Multi-Agent System

Chouyong Chen and Chao Xu

Management School, Hangzhou Dianzi University, Hangzhou, Zhejiang 310018, China

Correspondence should be addressed to Chao Xu; chaoxu_hdu@163.com

Received 13 March 2018; Revised 29 July 2018; Accepted 5 August 2018; Published 26 August 2018

Academic Editor: Luciano Caroprese

In the process of collaborative procurement, buyers and suppliers are prone to conflict in cooperation due to differences in needs and preferences. Negotiation is a crucial way to resolve the conflict. Aimed at ameliorating the situations of underdeveloped self-adaptive learning effect of current collaborative procurement negotiation, this paper constructs a negotiation model based on multi-agent system and proposes a negotiation optimization strategy combined with machine learning. It provides a novel perspective for the analysis of intelligent SCM. The experimental results suggest that the proposed strategy improves the success rate of self-adaptive learning and joint utility of agents compared with the strategy of single learning machine, and it achieves win-win cooperation between purchasing enterprise and supplier.

1. Instruction

Information technology has enabled, and in some cases forced, enterprises to reorient their internal capabilities and to redefine their business models to develop e-commerce techniques. In order to attain timely responsiveness and to proffer higher service level, constructive cooperation among partners in supply chain is critical in any endeavor to ameliorate disruptions and mitigate risks [1]. A small number of successful contemporary associations have transformed from an opportunistic doctrine of cooperation to a synergistic ethos and integrated their supply chain procedures.

The synergism of Cluster Supply Chain (CSC), comprising collaborative manufacture, collaborative procurement, collaborative logistics, and collaborative inventory, is the coupling organizing form between industrial cluster and supply chain. And it helps small and medium-sized enterprise (SME) shorten the transaction cycles and reduce costs. Procurement directly affecting the production and operation is the key link in the development of the whole enterprise. In the fierce market competition, therefore, the purchasing mode gradually shifts from traditional independent purchasing that faces the problems of small quantity discount, low bargaining power, and slow response to customer demand to collaborative procurement.

In this article, we consider a distributed supply chain (SC) in that each member seeks to optimize personal performance and independently plans his business. A large measure of supply chain managements (SCM) have to communicate and negotiate effectively with SC members. In the process of collaborative purchasing, buyers and suppliers are prone to conflict in cooperation due to differences in needs and preferences. Negotiation is a crucial way to resolve the conflict and an effective mechanism for supply chain coordination and cooperation. It has been demonstrated that the information sharing between the buyers and suppliers ensures effective supplier participation and enhances mutual understanding, which contributes to more excellent performance over the rivals [2]. Negotiation is reckoned to be a sound approach for participators to exchange messages, understand other perspectives, and identify new order alternatives based on the information and knowledge learned in the process. And it allows enterprises in the CSC to prevent both self-interest and local optimization of finicky partner, to proceed to the optimization of objectives of all participators and to achieve a win-win situation in SCM.

Former researchers have paid much attention to negotiation problem over the past decade and proposed some salient models. The majority of models primarily use either methods of the improved algorithms or game-theoretic techniques

as a basis to formulate autonomous negotiation. However, those approaches are considered to be complicated to spread to widespread problem fields due to the uncertainty and complexity in real-world negotiation. This paper ameliorates the negotiation model combining multi-agent system (MAS) with machine learning for further tackling the conflict in cooperation. New model provides a buyer with a method for purchasing a product systematically. And it helps in achieving a win-win cooperation between two sides during the process of collaborative procurement with supply chain as far as feasible.

The remainder of the paper is structured as follows. Section 2 shows literature review. Section 3 recalls some general concepts of key techniques. Section 4 is devoted to a negotiation model of collaborative procurement based on MAS. Section 5 describes a self-adaptive negotiation optimization strategy combined with dynamic selective ensemble learning. Finally, the experiments design, results, and concluding remarks are presented in Sections 6 and 7, respectively.

2. Literature Review

Given that research on negotiation of collaborative procurement is new and largely fragmented, it is practically paramount to arouse individuals' attention. Previous studies have, nevertheless, proposed a basic model of supply chains and a negotiation strategy for solving conflicts in consideration of efficiency and cost. A multi-objective cooperative production–distribution planning model was formulated by Jolai et al. [3] applying the fuzzy goal programming approach to maximize the gains of all participators. To discover the optimal solutions of resource allocation, Lin et al. [4] recommended a collaborative negotiation mechanism that was built on price schedules decomposition algorithm. But the popular methods to research the negotiation's conundrum in SCM involve Game Theory and artificial intelligence (AI). Game theorists deem the negotiation as an incomplete, dynamic information game, and attempt to settle the game by offering some predictions on certain conditions [5]. Primary methodological tools of Game Theory are Nash game [6] and Stackelberg game [7], which concentrate on the sequential and simultaneous decision-making of multiple players, respectively. For those relevant analytic modeling studies, the problem is analyzed mostly from a theoretical perspective. Despite being extremely successful in a quantity of situations, the game theoretical approach is considered to be difficult to spread to universal problem fields owing to the uncertainty and complexity in real-world negotiation.

Compared to Game Theory, participants that bargain with consideration of human preference and thoughts could be considerably represented by the agent technology which is a branch of AI [8]. The use of information and communication technology tools, offering the capacities of customer sensitivity, information sharing, and process integration, is observed as the uppermost enabler for this collaborative perception's realization [9]. In computer science, an agent is generally considered as a software entity, which is autonomous to communicate and coordinate with other agents to accomplish its design objectives. Consequently, multi-agent simulation modeling, which originated from AI, is suitable for the conduction of distributed system and has certain advantages in being testable, quantifiable, and efficient. It is superior in expansibility, is easy to configure, and has been widely used in the SCM. Kwon et al. [10] constructed an integrated framework that was based on multi-agent cooperation and case-based reasoning to help address emerging uncertainties. Lin et al. [11] demonstrated a supply chain coordination model of multi-agent and put forward a conflict solution method built on constraint satisfaction algorithm due to the different form of demand. Considering the conflict between businesses caused by the difference of information asymmetry and goals, Behdani et al. [12] developed a negotiation method based on multi-agent in the condition that demand is uncertain. The significance of addressing negotiation mechanisms for collaborative matters is shown by the discussed literatures. The combination of negotiation model and optimization technology is requisite to help negotiators achieve optimal selections.

In order to better promote the agent's self-adaptive negotiation ability, an army of scholars have begun to introduce machine learning into the negotiation. Bayesian Learning estimates the probability distribution of opponent negotiation parameters and preferences and adaptively adjusts the concession strategy [13]. Q-Learning generates the optimal negotiation strategy by calculating the utility cumulative value [14]. Radial Basis Function (RBF) neural network is capable of optimizing the Actor-Critic learning algorithm to predict and amend the concession magnitude of agents [15]. Unfortunately, previous self-adaptive negotiation is built on a single or integrated learning machine to draw the final result [16]. Selective ensemble learning improves the efficiency of general integrated learning machine by eliminating the less accurate ones in sublearning models [17].

This paper is built on our previous work in the field of automated negotiation. In particular, it lays the foundation for accomplishing an experiment to investigate the performance of agent which is operating in the supply chain system and equipped with our negotiation model. The main contribution consists of constructing a negotiation model concerning collaborative procurement based on MAS by analyzing the characteristics of multilateral transact and proposing a negotiation strategy founded on dynamic selective ensemble learning. We exploited supply chain analysis detailedly that was based on agent technology, which detects novel patterns through the improved data mining techniques and provides a new perspective for the analysis of intelligent SCM. Moreover, agent job was led by this association between intelligent agents and machine learning to do faster and better. And the negotiation strategy has also potential for big data decrement and compression.

3. Methods

3.1. Machine Learning. Machine learning gradually becomes an irreplaceable method for processing data in the big data era. As an embranchment of AI, it has entered foreland of the mainstream computer science's research that often uses statistical techniques to give agents the ability to learn with data, without being explicitly programmed. Machine learning

has substantial connections with mathematical optimization, which delivers theory, application domains, and methods to the field. Moreover, it is a popular method practiced to devise complicated models and algorithms for prediction. These analytical models permit researchers to find results and authentic decisions and reveal hidden insights via learning from historical relationships and tendencies in the data.

3.2. K-Means Clustering. K-means clustering, an unsupervised learning, is fundamentally a partitioning method that is utilized to analyze data and treat the data's observations as objects on the basis of locations and distance between diverse input data points. It helps to partition the undisposed objects into mutually exclusive clusters (K) so that objects remain as close as possible to each other within individual cluster but as far as possible from other clusters' objects.

3.3. Support Vector Machine. Support Vector Machine (SVM), introduced by Vapnik, is originated from the theory of structural risk minimization belonging to statistical learning theory. The essential idea of SVM is to map input vectors into a high dimensional feature space and construct the optimal separating hyperplane in this space. SVM tries to minimize an upper bound of the generalization error by maximizing the margin between the test data and the separating hyperplane [18]. It has several merits: (1) A unique hyperplane maximizing the margin of separation between the classes can be discovered by SVM, so it has a good ability of robustness. (2) SVM's power is to use kernel function to transform data from the low dimension space to the high dimension space and create a linear binary classifier. (3) The solving of SVM is a convex programming problem, and its local optimum is selected as the global optimum. In the field of machine learning, models combined with learning algorithms for analyzing and classifying data are represented by SVM.

4. Negotiation Model of Collaborative Procurement Based on MAS

One of the most distinguishing advantages of using MAS for SCM is the dynamic supply chain construction via automated negotiation between agents. In the MAS, the coordinator agent is introduced to regulate multiple buyer and seller agents. A distributed negotiation model based on MAS is demonstrated in Figure 1. The model assists enterprises in choosing the most suitable suppliers quickly, efficiently, and economically. The system consists of 3 mutually coordinated agents: CA represents the supplier agent, PA the purchasing enterprise agent of industrial cluster, and MA the broker agent of collaborative purchasing service. Agents participating in the negotiation must register with MA (such as an e-commerce platform) in advance and configure a unique ID. The MA manages various information in the negotiation process and coordinates the communication between the agents. The selection of the supplier is done with the assistance of the MA and repeated negotiation between the PA and the CA (the types of messages used by the agents in the negotiation process are shown in Table 1).

MA: (i) It promptly registers, verifies, and updates information about registered agents. (ii) It duly publishes, forwards, and organizes messages. (iii) It comprehensively utilize real-time environment and enterprise data to evaluate the operation of businesses.

PA: If Reply is received, PA will compare the property values of the products given by the participating CA with accredited ones, and then send Improve to the nonoptimal CA. Subsequently, it selects CA whose values are no less than the threshold as a candidate supplier. If there is no qualified supplier, purchasing enterprise will modify the relevant threshold and renegotiate with all suppliers. Finally, the result opted for is sent to the MA with Selection. After receiving Confirm, if the CA is found to have objected to the negotiation result, check the modification and resend Improve until no objection occurs.

CA: After monitoring Announce published by the MA, if the requirements of order are met, deliver the Bid to participate in the negotiation. In the event of corresponding values suggested by the PA being acceptable, during Adjust, CA sends a new Bid, or else emits Reject. Eventually, when receiving Result, the selected CA checks the content of the protocol, and if there is no objection, the Accept is fed back. Otherwise, the Refuse is transmitted to point out the problem.

The specific negotiation process is showed in Figure 2.

5. Self-Adaptive Negotiation Optimization Strategy

5.1. Negotiation Parameter. Negotiation parameters consisted of four elements which are proposed and explained in Table 2.

$$\mathrm{NM} = \{\mathrm{A}, \mathrm{P}, \mathrm{w}, \mathrm{U}\} \tag{1}$$

5.2. Concessional Learning Based on Dynamic Selective Ensemble of SVM. According to the current negotiation issues, the nearest neighbor sample set is used as the training sample to evaluate the performance of each submodel and select the better ones. In the negotiation, K-means algorithm is adopted for each negotiation issue, and the k sample subsets are found as the training datasets. And the Support Vector Machine (SVM) is used to learn the concession amplitude in each evaluation sample. Taking root-mean-square error (RMSE) as the evaluation criterion, we eliminate some submodels with poor performance. The combination weight is calculated and the final dynamic selective SVM model is established.

(1) K-means algorithm generates evaluation datasets. P_q is negotiation sequence to be predicted and its number of the nearest neighbor sample in the data set P_L is k, and the first k samples P_{k} can be got by calculating the Euclidean Distance P_D between P_q and the sample points P_{i}.

$$P_D\left(P_q, P_i\right) = \sqrt{\sum_{i \in L}\left(P_q - P_i\right)^2} \tag{2}$$

(2) Input sample set P_{k}, and estimate concession amplitude with SVM. Assume that negotiation values of A_C and A_P in round t and issue j are denoted as P_t^C and P_t^P, respectively,

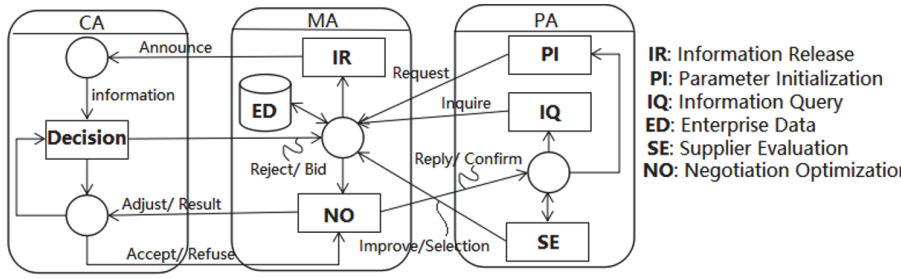


Figure 1: MAS negotiation model of collaborative procurement.

Table 1: Instructions of related messages.

Agent	Message	Description of Message
MA	Announce	Publish PA's requirement information to registered Agents
	Adjust	Forward PA's improved requests to CA
	Result	Inform of results and send the agreed protocol contents to CA
	Reply	Send the product and enterprise information to PA
	Confirm	Transform the confirmation messages or protocol modification information
PA	Request	Ask MA to release massage to corresponding CA
	Inquire	Consult MA for information about the supplier
	Improve	Request PA to improve the relevant attributes on MA
	Selection	Post results and agreements to selected PA
CA	Reject	Notify the MA not to participate in the consultation
	Bid	Give information of the product and the enterprise or submit the improved relevant attributes
	Accept	Acquaint MA the accepted agreements
	Refuse	Object the protocol or request for verification

Table 2: Instructions of negotiation parameters.

Parameters	Instructions
A	A_C represents Supplier, A_P Industrial cluster buyer
P	Issue value of negotiation
w	Weight vector of the issue
U	Utility value of the issue

and ΔD_t is the negotiation difference between A_C and A_P obtained by (3). The average concession amplitudes of A_C, A_P for the first t rounds are $\overline{C}_t^C$, $\overline{C}_t^P$. As inputs to the SVM, t, ΔD_t, $\overline{C}_t^C$, and $\overline{C}_t^P$ are mapped to the high dimensional space using the Radial Basis Function $H_t = (\varphi(t), \varphi(\Delta D_t), \varphi(\overline{C}_t^C), \varphi(\overline{C}_t^P))$. C_{t+1}^P is the output variable of the linear regression function obtained by (5).

$$\Delta D_t = \left| P_t^C - P_t^P \right| \tag{3}$$

$$\overline{C}_t^P = \sum_{i=2}^{t} \frac{\Delta D_i}{P_{i-1}^P} \tag{4}$$

$$C_{t+1}^P = w^T * \left(\varphi(t), \varphi(\Delta D_t), \varphi\left(\overline{C}_t^P\right), \varphi\left(\overline{C}_t^C\right) \right) + b \tag{5}$$

where w^T is the weight vector of 4 input variables and b is a offset value.

The error ε between predicted value y and function value C_{t+1}^Pcould be calculated by (6). If the error ε is regarded as an error-free fitting, then we can get the nonlinear regression function as (7) of the concession amplitude $C_{t+1}^P$of the opponent in round t+1. After the equivalent substitution, we can get the final regression function as (8).

$$\max\left\{0, \left| y - C_{t+1}^P \right| - \varepsilon \right\} \tag{6}$$

$$C_{t+1}^P = \sum_{j=1}^{n} \left(a_j - a_j' \right) K\left(H_t, H_{t-1} \right) + b \tag{7}$$

$$C_{t+1}^P = \sum_{j=1}^{n} a_j \exp\left\{ -\frac{\left\| H_t - H_{t-1} \right\|^2}{\sigma^2} \right\} + b \tag{8}$$

where a_j $(a_j > 0)$ is a Lagrange multiplier, identified by SVM training. Similarly, C_{t+1}^C is the predictive concession amplitude value of A_C in round t+1.

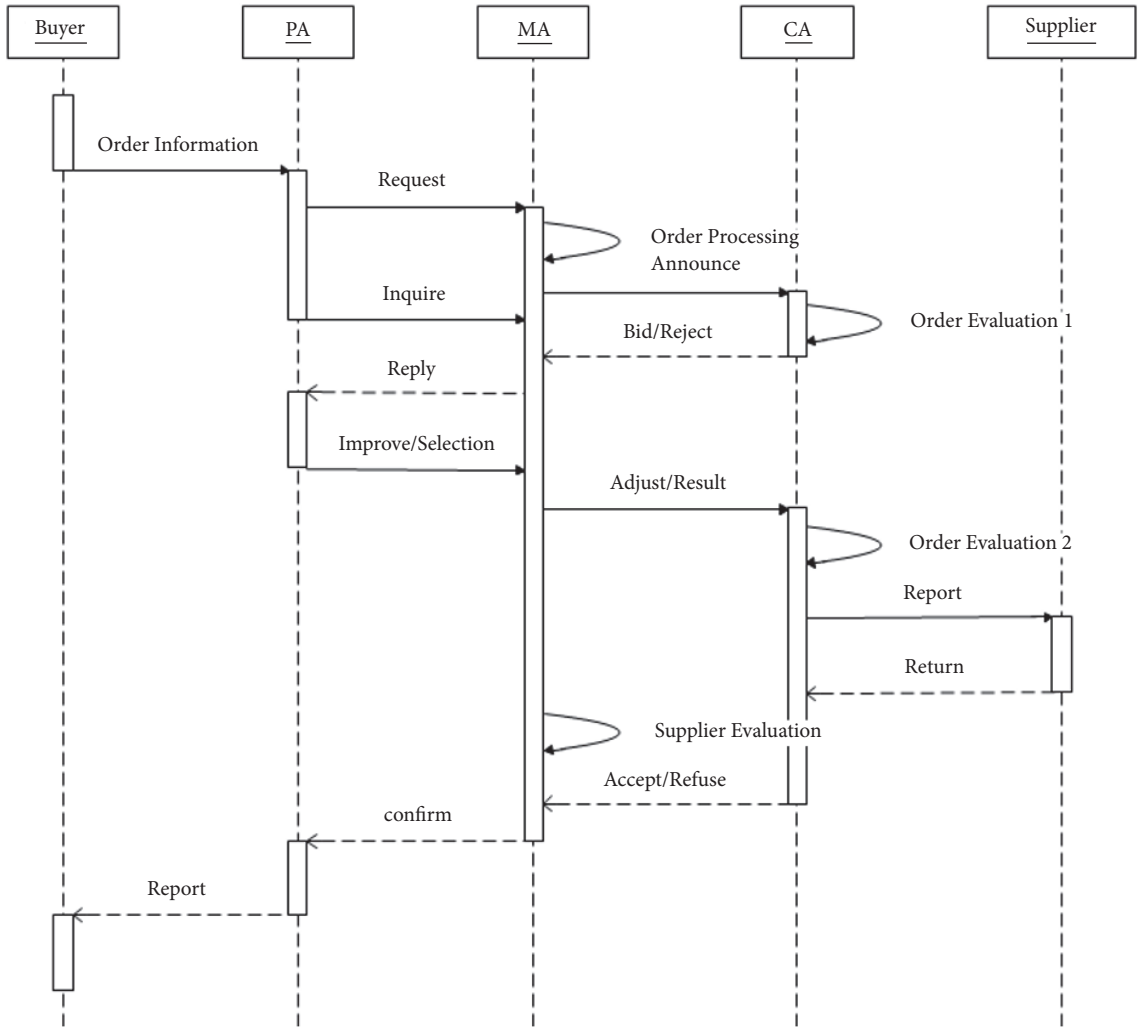


Figure 2: MAS sequence diagram of negotiation model.

(3) Using the RMSE as a filter criterion as (9), we select the corresponding first $\overline{k}$ sublearning machines.

$$E_{ij} = \sqrt{\frac{\sum_{i=1}^{k} \left(\widetilde{c}_{ij} - C_{ij}\right)^2}{k}} \tag{9}$$

where $\widetilde{c}_{ij}$ is the next predictive concession value in issue j of sublearning machine i and C_{ij} means the actual concession amplitude.

(4) Calculate the combined weight of each submodel. According to the RMSE value E_{ij} of the i-th submodel, the weight of the submodel is obtained.

$$\alpha_i = \frac{\left(1/E_{ij}^2\right)}{\left(\sum_{i=1}^{\overline{k}} \left(1/E_{ij}^2\right)\right)} \tag{10}$$

When all the k sublearning machines are successfully trained, select the $\overline{k}$ sublearning models with the smallest error. Input the actual concession C_{ij}, and then get the output of ultimate concession about issue j in the round t+1.

$$C_{t+1,j}^{C/P} = \sum_{i=1}^{\overline{k}} \alpha_i C_{ij} \tag{11}$$

5.3. Utility Optimization. Taking A_P as an example, the utility difference of sequential negotiations is used to decide whether to stop the current consultation. $C_{t+1,j}^{P}$ means a predictive concession value about issue j in round t+1. $P_{t,j}^{P}$ is an actual value of buyer A_P about issue j in round t.

$$U_{\mathrm{t}} = \sum_{j=1}^{n} w_j P_{t,j}^{P} \tag{12}$$

$$P_{t+1,j}^{P} = P_{t,j}^{P} + C_{t+1,j}^{P} \tag{13}$$

The error between the predictive utility value in round t+1 and actual utility value in round t can be calculated by coordinating equations (12) and (13). While $\Delta U_{t+1,t} > 0$, the

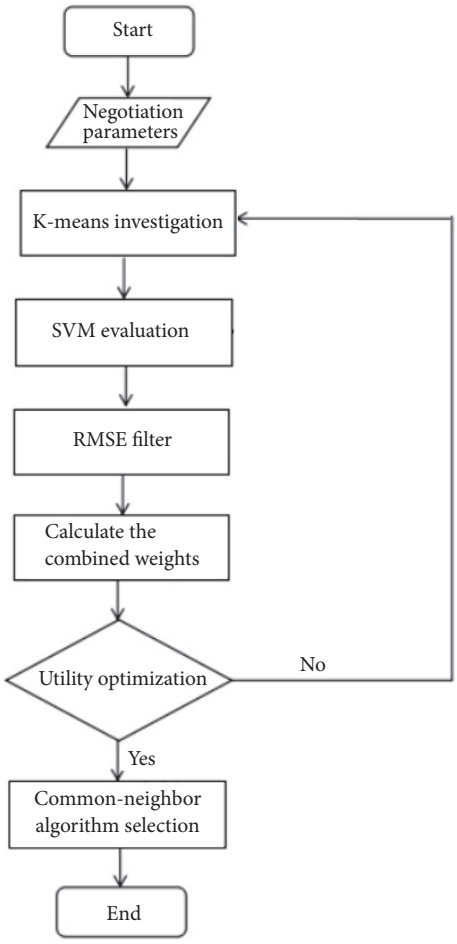


Figure 3: Flowchart of Self-adaptive negotiation optimization strategy.

utility of concession has not been maximized; it will increase. Conversely, end the concession.

5.4. Selection of the Most Appropriate Partner. After the negotiation, the common-neighbor algorithm [19] is applied to compute the similarity of the issues, and A_P choose more suitable partners according to the similarity.

$$S_{P,C} = \left(1 + e^{-D_{P,C}/2}\right) * \left\| I_P \cap I_C \right\| \tag{14}$$

$D_{P,C}$ means the total issue difference between A_P and A_C. $\|I_P \cap I_C\|$ is the quantity of accredited issue after the negotiation.

Procedures are as follows (see Figure 3). First, K-means search was adopted to generate sample sets. Second, the SVM was used to learn the concession amplitude in each evaluation sample and then eliminated the poor performance of sublearning model with RMSE and calculated the combined weight and the final dynamic selective SVM model was established. Third, the utility function was used to decide whether to terminate the negotiation. Finally, the most appropriate partner was selected on the basis of issues' similarity calculated with common-neighbor algorithm.

Furthermore, the self-adaptive negotiation optimization strategy is also suitable for complicated problems of big data in massively parallel environments. The complexity of big data could be decreased by data processing algorithms' application.

6. Simulation Example

Relying on modern logistics network system, Yiwu has become the largest small commodity distribution center in the world. The merchandise is sold to Europe, America, the Middle East, and South Asia and other regions. Yiwu market now has more than 4.3 million square meters of business area, 63 thousand operators, and more than 400 thousand kinds of products. In 2016, the trading volume of commodity markets reached 373 billion RMB and the total export-import volume extended to 223 billion RMB (Yiwu China Commodities City Group Official Website 2017). Yiwu Global Purchasing (www.yiwuok.com) as an e-commerce platform contributed 60% of the first value. The key link of supply chain synergism is to utilize e-commerce platform services to develop a healthy relationship of trust among partners and establish an effective mechanism for information collaboration. This paper takes Yiwu Small Commodity Industry Cluster (SCIC) as an instance and grabs five main parameters: product price, quantity, delivery time, warranty time, and defective rate as the negotiation issue. The effectiveness of self-adaptive Integrated Optimization Strategy (IOS) is verified by using Matlab R2014a, which is compared with the General Learning Strategies (GLS) based on single SVM.

According to the historical data analysis of electric appliances industry in Yiwu SCIC, the supplier cares more about price, quantity, and delivery time, while concentrating less on warranty time and defective rate. The purchasing enterprise is a little bit different; they focus on defective rate rather than warranty time, demonstrated detailedly in Table 3. Initial experimental datasets could be extracted from Dataverse repository. The whole examinations were performed on a laptop (4 GB of RAM that operated under Windows 10 desktop, Intel core i3 CPU @ 2.54 GHz). In addition, we selected the open source libraries, VLFeat for K-means clustering and LIBSVM for SVM algorithm, with excellent interfaces in Matlab for ease of use. To get the generation of optimal solutions, the experimental time is limited to 2 minutes.

A separating hyperplane of datasets illustrated by the IOS is exhibited in Figure 4. In place of the smaller margin, the hyperplane creates sheltered subregions to make most examples with identical class label drop on the same side of the decision boundary. And subregions are produced by decision boundary with diverse piecewise shapes, such as jutting out as peninsulas that are virtually surrounded by the antagonists. The misclassifications might comprise some stray examples submerged in the opponents. As the crucial target of sustaining the native class' membership, the IOS eliminates the stray examples—those characterized as black solid symbols—from the hyperplane. As mentioned above, we are working on the assumption that the margin shrinkage is a price to trade off with the misclassification decrease in the practice stage.

Table 3: Intervals and weights of negotiation issue.

Parameters	Intervals of supplier's issue	Intervals of purchaser's issue	Weight vector of supplier	Weight vector of purchaser
Price/Yuan	[100, 150]	[100, 130]	0.40	0.35
Quantity	[800, 1000]	[850, 1200]	0.25	0.30
Delivery time/Month	[1.5, 2]	[1, 2]	0.20	0.20
Warranty time/Month	[12, 18]	[15, 24]	0.10	0.05
Defective rate/%	[80, 95]	[90,95]	0.05	0.10

Table 4: Error rate (%) comparison of experimental results.

Strategy	Min error	Max error	Median error	Average error	Standard Deviation
GLS	2.8	32.1	14.2	15.86	8.37
IOS	3.2	26.1	10.1	11.97	6.05

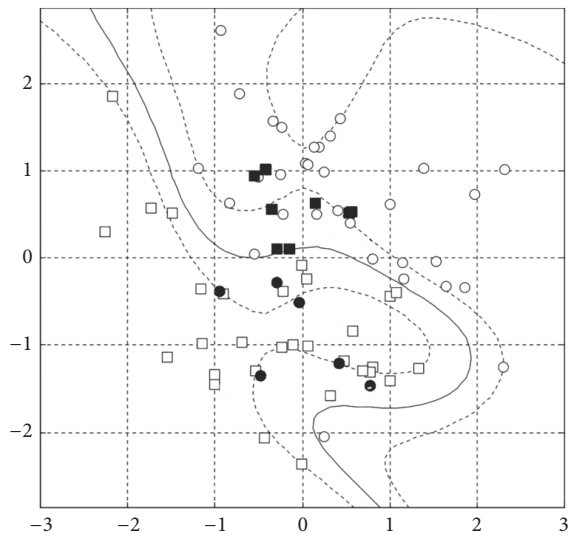


Figure 4: A separating hyperplane depicted by the IOS.

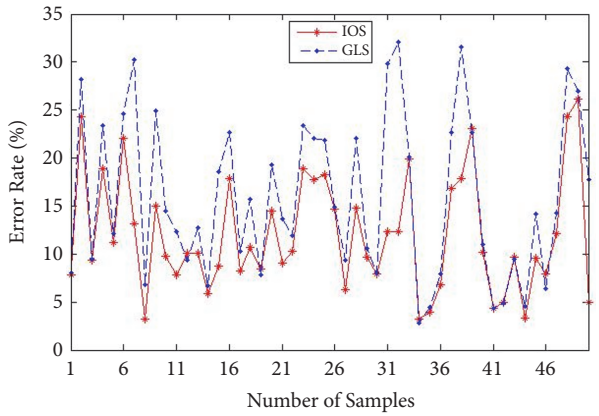


Figure 5: Simulation results of error rate of 2 strategies.

50 couples are selected in the experiment to predict the margin of opponent concessions by comparing two strategies. According to Figure 5, we can draw the conclusion that in most cases this IOS infers lower error rate than the ordinary single SVM. Additionally, the basic descriptive statistics of the data is provided in Table 4. The average error of IOS for all 50 objects is 11.97% with a standard deviation of 6.05%. It could be seen that the IOS outperforms the GLS in four vital error measures. The max error is 6.0% lower, the median error is 4.1% lower, the average error is 3.89% lower, and standard deviation is 2.32% lower than the GLS, respectively.

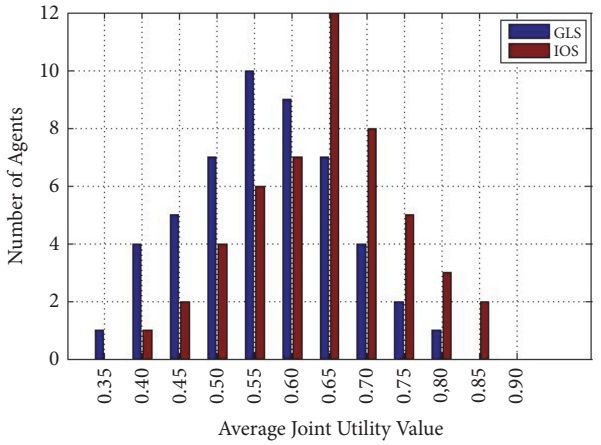


Figure 6: The comparison of joint utility and successful agent.

In Figure 6, the average joint utility value founded by IOS is mainly concentrated in [0.50, 0.75], while another value is mainly concentrated in [0.40, 0.70]. The total average joint utility of the former is 0.641, and 60% of agents are higher than that value. Nevertheless, the numbers of the latter calculated severally are 0.565 and 46%. Distinctly, the strategy proposed by this paper is superior to GLS in both the amounts of successful agents and joint utility value.

7. Conclusions

Previous studies proposed a number of basic supply chain models which are difficult to spread to universal problem fields owing to the uncertainty and complexity in real-world negotiation. The most fascinating modern application of ensemble systems lies in processing high dimensional, complex, and big data that cannot be analyzed efficiently

by single-model methods. To better solve the conflict in negotiation, this paper has discussed the negotiation problem of collaborative procurement operating on MAS model with a negotiation optimization strategy. We exploited supply chain analysis minutely based on agent technology and machine learning, which provides a new perspective for the analysis of intelligent SCM. Apparently, we perceive that the negotiation and learning are key aspects in the system performance by the simulation of the proposed MAS model for the procurement management of CSC. The agents have symmetric preferences, complicating the negotiation. However, the learning helped each one acquire the ultimate strategy choice. The experimental results show that the IOS based on dynamic selective ensemble SVM can reduce the error rate and elevate the joint utility, compared with GLS of the ordinary single learning machine. The test reveals that the model plays a key role in negotiation issue inside the intelligent SCM, and the agent negotiation performance and efficiency can be enhanced via the combination of the improved data mining techniques.

The procurement management of supply chain involves fabrication, inventory, distribution, and other issues, and the supply chain needs collaboration of upstream and downstream enterprises to achieve a synergistic, dynamic, and timely supply-production-marketing operation mode. Future research will focus on the resolution of conflict in self-adaptive negotiation to further improve the intelligent level of supply chain.

Data Availability

The datasets analyzed during the current study are available in Dataverse repository: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi%3A10.7910%2FDVN%2FVT2AQJ&version=DRAFT.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (U1509220).

References

[1] J. Hallikas, I. Karvonen, U. Pulkkinen, V.-M. Virolainen, and M. Tuominen, "Risk management processes in supplier networks," *International Journal of Production Economics*, vol. 90, no. 1, pp. 47–58, 2004.

[2] Y. Zhang, L. Wang, and J. Gao, "Supplier collaboration and speed-to-market of new products: the mediating and moderating effects," *Journal of Intelligent Manufacturing*, vol. 28, no. 3, pp. 805–818, 2017.

[3] F. Jolai, J. Razmi, and N. K. Rostami, "A fuzzy goal programming and meta heuristic algorithms for solving integrated production: distribution planning problem," *Central European Journal of Operations Research*, vol. 19, no. 4, pp. 547–569, 2011.

[4] Y.-I. Lin, Y.-W. Chou, J.-Y. Shiau, and C.-H. Chu, "Multi-agent negotiation based on price schedules algorithm for distributed collaborative design," *Journal of Intelligent Manufacturing*, vol. 24, no. 3, pp. 545–557, 2013.

[5] K. Govindan, A. Diabat, and M. N. Popiuc, "Contract analysis: a performance measures and profit evaluation within two-echelon supply chains," *Computers & Industrial Engineering*, vol. 63, no. 1, pp. 58–74, 2012.

[6] M. Leng and M. Parlar, "Game theoretic applications in supply chain management: a review," *Infor Information Systems & Operational Research*, vol. 43, no. 3, pp. 187–220, 2005.

[7] J.-C. Hennet and S. Mahjoub, "Toward the fair sharing of profit in a supply network formation," *International Journal of Production Economics*, vol. 127, no. 1, pp. 112–120, 2010.

[8] N. C. Karunatillake, N. R. Jennings, I. Rahwan, and P. McBurney, "Dialogue games that agents play within a society," *Artificial Intelligence*, vol. 173, no. 9-10, pp. 935–981, 2009.

[9] Y. Wu and J. Angelis, "Achieving agility of supply chain management through information technology applications," *International Federation for Information Processing*, vol. 246, pp. 245–253, 2007.

[10] O. Kwon, G. P. Im, and K. C. Lee, "MACE-SCM: a multi-agent and case-based reasoning collaboration mechanism for supply chain management under supply and demand uncertainties," *Expert Systems with Applications*, vol. 33, no. 3, pp. 690–705, 2007.

[11] F.-R. Lin and Y.-Y. Lin, "Integrating multi-agent negotiation to resolve constraints in fulfilling supply chain orders," *Electronic Commerce Research and Applications*, vol. 5, no. 4, pp. 313–322, 2006.

[12] B. Behdani, A. Adhitya, Z. Lukszo, and R. Srinivasan, "Negotiation-based approach for order acceptance in a multiplant specialty chemical manufacturing enterprise," *Industrial & Engineering Chemistry Research*, vol. 50, no. 9, pp. 5086–5098, 2011.

[13] J. Zhang, F. Ren, and M. Zhang, "Bayesian-based preference prediction in bilateral multi-issue negotiation between intelligent agents," *Knowledge-Based Systems*, vol. 84, pp. 108–120, 2015.

[14] L. Chen, H. Dong, and Y. Zhou, "A reinforcement learning optimized negotiation method based on mediator agent," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7630–7640, 2014.

[15] Z. Ma, C. Wang, Y. Niu, X. Wang, and L. Shen, "A saliency-based reinforcement learning approach for a UAV to avoid flying obstacles," *Robotics and Autonomous Systems*, vol. 100, pp. 108–118, 2018.

[16] J. Heinermann and O. Kramer, "Machine learning ensembles for wind power prediction," *Journal of Renewable Energy*, vol. 89, pp. 671–679, 2016.

[17] Y. Liu, B. He, D. Dong et al., "Particle swarm optimization based selective ensemble of online sequential extreme learning machine," *Mathematical Problems in Engineering*, vol. 2015, Article ID 504120, 10 pages, 2015.

[18] N. B. Peng, Y. X. Zhang, and Y. H. Zhao, "A SVM-kNN method for quasar-star classification," *Science China Physics, Mechanics & Astronomy*, vol. 56, no. 6, pp. 1227–1234, 2013.

[19] Y. H. He, D. B. Chen et al., "Similarity algorithm based on users common neighbors and grade information," *Computer Science*, vol. 37, no. 9, pp. 184–186, 2010.

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 4925841, 12 pages
https://doi.org/10.1155/2018/4925841

Research Article

High-Order Degree and Combined Degree in Complex Networks

Shudong Wang,[1] Xinzeng Wang ,[2] Qifang Song,[2] and Yuanyuan Zhang[3]

[1]*College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266580, China*
[2]*College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao 266510, China*
[3]*Computer Engineering Institute, Qingdao University of Technology, Qingdao 266520, China*

Correspondence should be addressed to Xinzeng Wang; wangelxz@126.com

Received 30 January 2018; Revised 22 May 2018; Accepted 3 June 2018; Published 27 June 2018

Academic Editor: Ester Zumpano

We define several novel centrality metrics: the high-order degree and combined degree of undirected network, the high-order out-degree and in-degree and combined out out-degree and in-degree of directed network. Those are the measurement of node importance with respect to the number of the node neighbors. We also explore those centrality metrics in the context of several best-known networks. We prove that both the degree centrality and eigenvector centrality are the special cases of the high-order degree of undirected network, and both the in-degree and PageRank algorithm without damping factor are the special cases of the high-order in-degree of directed network. Finally, we also discuss the significance of high-order out-degree of directed network. Our centrality metrics work better in distinguishing nodes than degree and reduce the computation load compared with either eigenvector centrality or PageRank algorithm.

1. Introduction

The theory of network has gone through rapid development since the late 1990s. One of the hottest points is the research on the attributes of network. Node degree has always been considered as one of the most important and fundamental attributes. Many relative researches have defined other attributes of network based on node degree [1, 2], such as degree distribution [3, 4], clustering coefficient [5, 6], the characteristic path length [6], and so on. As early as 1960s, Rapoport [3, 4] emphasized the importance of the degree distribution in all kinds of real networks. Wasserman and Faust [5] introduced fraction of transitive triples in social network in 1994. In order to describe cliquishness of a typical neighborhood, Watts and Strogatz [6] defined clustering coefficient of general complex network in 1998 based on the fraction of transitive triples. Watts and Strogatz [6] also defined the characteristic path length to measure the typical separation between two nodes in the network. In more recent years, numerous researches paid attention to degree distribution [7–10]. Early researches believed that degree distribution followed Poisson Distribution just as the random network theory has described [11, 12]. Recent researches have found out that the degree distribution for a large number of networks, such as the World Wide Web [13], the Internet [14], the metabolic networks [15], genome-wide disruption networks for yeast [16], and the network of interregional direct investment stocks across Europe [17], have a power-law tail. Such networks are called scale-free [18].

When a practical problem is transformed into a complex network model, people tend to use node centrality to describe the importance and influence of a node, or people need to sort these nodes [19]. Node degree (or degree centrality) is one of the basic methods of sorting the nodes [20–29]. Other common methods are also based on node degree [1, 2]. As long ago as 1948, Bavelas [30] studied the center of social network. Sabidussi et. al. [31] defined what it means for a network node to approach closeness. Freema [32] used node degree and betweenness to define two kinds of node centrality, and, furthermore, he used node centrality to define graph centrality. Network eigenvector centrality [33, 34] is often used to describe the importance of nodes in social network. In 1998, Brin and Page [35, 36] simplified the eigenvector centrality for undirected network into PageRank algorithm (*PR*), which is widely used in searching engine Google [36] and many other directed networks [37–42]. By

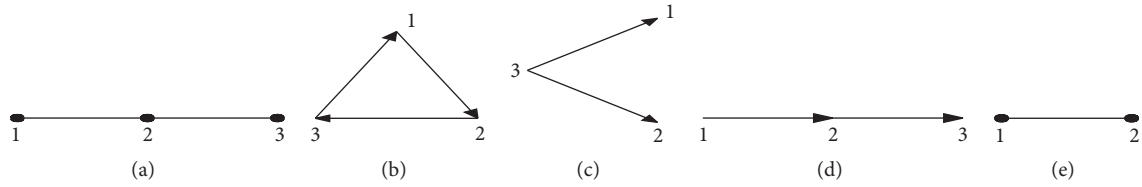


Figure 1: Diagram of random walk. Choose the origin node 1: (a) the process of splitting and random walking could be repeated an infinite number of times, because origin node 1 is connected to node 2; (b) the ball can walk randomly an infinite number of times without split since the origin node is one of the directed loop; (c) the ball split and walk randomly only once; (d) the ball can only walk twice without splitting; (e) the ball can walk an infinite number of times without split.

using the degree distribution of neighbor nodes, Ai [43] gave the definition of the neighbor vector centrality. Zeng [44] implemented the Mixed Degree Decomposition (*MDD*) procedure by using coreness centrality, and he defined the mixed degree of nodes. Bae [45] defined the kernel centrality of a node by using the kernel of the neighbor nodes.

The degree centrality used for sorting nodes has the advantages of simple calculation, but the results are not accurate enough. Therefore, it may require verification by other methods [20] or other network attributes [27, 28]. Closeness centrality and betweenness centrality are often given with degree centrality for comparison purpose [27, 28]. The computation of both closeness centrality and betweenness centrality are so complicated that big networks often need fast approximate algorithm [46–48]. Even though the eigenvector centrality on undirected network and the PageRank on directed work can give satisfying results in nodes sorting, those two methods often involve expensive computations, such as iterations [42]. In the following paragraphs, we will define several novel centrality metrics, which are cheaper in computation compared with closeness centrality, betweenness centrality, eigenvector centrality on undirected network, and PageRank on directed network. We will show that node degree, the eigenvector centrality on undirected network and in-degree, and PageRank on directed network are all special cases of (or equivalent to) one of our novel centrality metrics (see Sections 4.1 and 4.2). It should be pointed out that the combined degree defined in our paper is different from the mixed degree in Zeng [44], which is the Mixed Degree Decomposition of coreness centrality (see Section 4.4).

2. Methods

Random walk has always been one of the most important methods in the research of complex network [49–53]. Noh et. al. [49] derived the mean first passage time (*MFPT*) between any two nodes by using random walk of complex network. Tejedor [50] computed the *MFPT* of a network based on a broad class of random walk. Rosvall et. al. [51] used the probability flow of random walks on a network as a proxy for information flow in the real system. Saramäki et. al. [52] generated scale-free networks based on selecting parent nodes by using random walk. Weng [53] used the newly defined mean first traverse distance (*MFTD*) to describe anomalous random walks. Now let us think about an ideal random walk of a simple network (could be an undirected network or a directed network): choose a node of the network, which we call the origin node, put a ball at the origin node, and the ball obeys the following rules to split repeatedly and to walk randomly.

(i) Splitting: the ball splits up into $d_G(v)$ (or $d_G^+(v)$) balls at the node v, where $d_G(v)$ is the degree of v in undirected network and $d_G^+(v)$ is the out-degree of v in directed network.

(ii) Random walk: after every split, the balls move to the adjacent nodes along the $d_G(v)$ edges; in directed network they can only move along $d_G^+(v)$ outgoing edges.

(iii) Disappearance: balls that can no longer walk would disappear, for example, at the isolated nodes or dead nodes of directed network.

The number of repeats of the random walk would be different depending on the selection of network and/or the selection of origin node, such as in Figure 1.

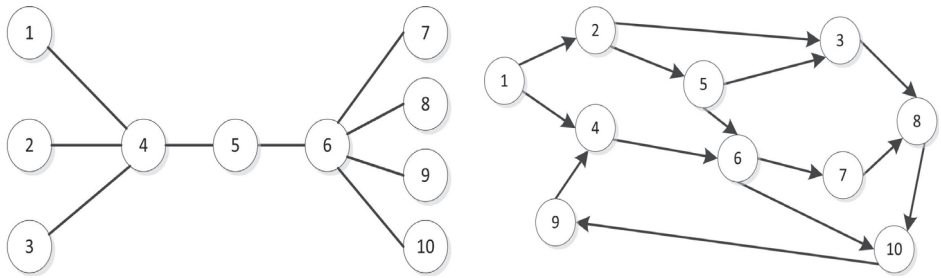
In the undirected network, the number of balls after s^{th} split is defined as the s-order degree of the origin node v, denoted by $d_G^s(v)$. Clearly, if $s = 1$, $d_G^1(v)$ is the degree $d_G(v)$ of v (see Theorem 2, Section 4.1). In the directed network, the number of balls after s^{th} split is called the s-order out-degree of the origin node v, denoted by $d_G^{+s}(v)$. Also, when $s = 1$, $d_G^{+1}(v)$ is the out-degree $d_G^+(v)$ of v (see Theorem 3, Section 4.2).

Consider a more complicated random walk of a directed network: select a network node v, which we call the sink node. Then we put balls at every node in the network (including the sink node v), and all of the balls follow the above rules to split repeatedly and to walk randomly, the number of balls at the sink node after the s^{th} random walk is defined as the s-order in-degree of the sink node v, denoted by $d_G^{-s}(v)$. Clearly, if $s = 1$, $d_G^{-1}(v)$ is the in-degree $d_G^-(v)$ of v (see Theorem 3, Section 4.2).

2.1. 2-Order Degree. As for the undirected network, according to the above definition, 2-order degree $d_G^2(v)$ of the node v is the number of two-edge paths connected to v. The two-edge paths may overlap. For example, Figure 1(e), both nodes have an overlap two-edge path, then the 2-order degree of both nodes is 1. Note that the 2-order degree is not necessarily equal to the number of neighbors of a node's neighbors,

Table 1: The 2-order out-/in-degree of Figure 2.

	Name of each node	1	2	3	4	5	6	7	8	9	10	Mean
Figure 2(a)	Degree	1	1	1	4	2	5	1	1	1	1	1.9
(Undirected network)	2-order degree	4	4	4	5	9	6	5	5	5	5	5.2
	Out-degree	2	2	1	1	2	2	1	1	1	1	1.4
Figure 2(b)	In-degree	0	1	2	2	1	2	1	2	1	2	1.8
(Directed network)	2-order out-degree	3	3	1	2	3	2	1	1	1	1	1.4
	2-order in-degree	0	0	2	1	1	3	2	3	2	4	1.8



(a) (b)

Figure 2: The degree and 2-order degree of a network.

since there may be more than one two-edge path between any two nodes (here the two nodes may be the same one). The computation of node 2-order degree $d_G^2(v)$ is relatively simple. Suppose matrix $A = (a_{ij})$ is the adjacency matrix of the undirected network G, we call $A_2 = A^2$ the 2-order adjacency matrix, and then the sum of elements in each row (or each column) of A_2 is the 2-order degree of the corresponding node. Use $\{A_2\}_i$ to denote the sum of the i-th row of matrix A_2, then the 2-order degree of the i-th node v_i of the undirected network G is $d_G^2(v_i) = \{A_2\}_i$.

A directed graph has both a 2-order in-degree and a 2-order out-degree for each node, which are the numbers of incoming and outgoing two-edge paths, respectively. Still denote the adjacency matrix as $A = (a_{ij})$, and denote $\{A_2\}_i$ and $\{A_2\}^i$ as the sum of the i-th row and i-th column of the 2-order adjacency matrix $A_2 = A^2$, respectively. Then the 2-order out-degree and the 2-order in-degree of the i-th node v_i in this directed network are $d_G^{+2}(v_i) = \{A_2\}_i$ and $d_G^{-2}(v_i) = \{A_2\}^i$, respectively.

If G has vertices $v_1, v_2, \cdots, v_n$, the sequence $(d(v_1), d(v_2), \cdots, d(v_n))^T$ is called the***degree sequence*** of G [54]. As we all know, the mean value and the distribution of degree sequence guide the definition of mean degree and degree distribution. We can obtain the mean 2-order degree and the 2-order degree distribution of the network by using the definition of 2-order degree.

In Figure 1, (a) the 2-order degree of the three nodes all is 2; (b) the 2-order out-degree and 2-order in-degree of the three nodes all are 1; (c) the 2-order out-degree and 2-order in-degree of the three nodes all are 0; (d) the 2-order out-degree of the three nodes is 1, 0, 0, and the 2-order in-degree of the three nodes is 0, 0, 1; (e) the 2-order degree of the two nodes is 1.

Figure 2 is a relatively complicated undirected network and a relatively complicated directed network. Table 1 shows the 2-order (in-/out-)degree of the all nodes and both mean 2-order degrees for the two networks in Figure 2. As we can see from Table 1, node 5 has the highest 2-order degree in Figure 2(a), which is the same as the common sense that node 5 is the most important node in Figure 2(a). However this conclusion cannot be obtained simply by calculating node degree. Of course we can obtain such conclusion by calculating node betweenness centrality or eigenvector centrality, and so forth. Another thing we can tell from Table 1 is that nodes having the highest 2-order out-degree are 1, 2, and 5, and the node having the highest 2-order in-degree is 10. Therefore, we could consider that nodes 1, 2, and 5 are the source nodes of (b), and the node 10 could be thought as the collection node of (b). In general, we need to run the PageRank algorithm of the entire network in order to get such conclusions.

2.2. s-Order Degree and Its Computation. Same as 2-order degree, suppose matrix A is an adjacency matrix of a network G, we denote $A_s = A^s (s \in N^*)$ as the s-order adjacency matrix of G. The matrix A_s was used to calculate the number of walk with length s between nodes [54–56]. Denote $\{A_s\}_i$ and $\{A_s\}^i$ as the sum of the i-th row and i-th column respectively in matrix A_s; then the s-order degree of node v_i in undirected network G is $\{A_s\}_i$ and $\{A_s\}_i = \{A_s\}^i$ (if G is a directed network, then the s-order out-degree is $\{A_s\}_i$, and the s-order in-degree is $\{A_s\}^i$). It is easy to define mean s-order (out-/in-)degree and the s-order (out-/in-)degree distribution, too.

The computation of s-order degree can make use of adjacency matrix, but this involves the exponentiation computation of matrix. The order of matrix A equals the number of nodes in the network, and there could be thousands and tens of thousands of nodes in a network. Therefore even though the computation of s-order adjacency matrix only involves the exponentiation computation of matrix and additions of integers, it would be a high requirement on the computer's CPU and memory. A simple method to tackle this problem is to firstly compute the lower order degree sequence of the network and then compute the higher-order degree sequence. We can prove Theorem 1 by the method of mathematical induction.

Theorem 1. *For s-order degree sequence, the following conclusions are true:*

(i) *If d^s is a s-order degree sequence of an undirected network G, A is the adjacency matrix, then the $(s+1)$-order degree sequence of G is $A * d^s$; particularly, if d^1 is the degree sequence of an undirected network G, then the $(s+1)$-order degree sequence of G is $A^s * d^1$.*

(ii) *Similarly, if d^{+s} and d^{-s} are the s-order out-degree sequence and s-order in-degree sequence of directed network respectively, A is the adjacency matrix, then the $(s+1)$-order out-degree sequence and the $(s+1)$-order in-degree sequence are $A * d^{+s}$ and $A^T * d^{-s}$ respectively; particularly, if d^{+1} and d^{-1} are the out-degree sequence and in-degree sequence of the directed network respectively, then the $(s+1)$-order out-degree sequence and the $(s+1)$-order in-degree sequence are $A^s * d^{+1}$ and $(A^T)^s * d^{-1}$, respectively.*

2.3. Combined Degree. Based on the s-order degree defined as above, the following gives the definition of combined degree of a node v_i of an undirected network:

$$d_G^{Com}(v_i) = \alpha_1 \cdot d_G^1(v_i) + \alpha_2 \cdot d_G^2(v_i) + \cdots + \alpha_s \cdot d_G^s(v_i) + \cdots, \quad (1)$$

where $d_G^m(v_i)$ $(m = 1, 2, \cdots, s, \cdots)$ denotes the various order degrees of node v_i, and constants α_m $(m = 1, 2, \cdots, s, \cdots)$ are all nonnegative real numbers, with a sum of 1. If $\alpha_1 = 1$ and $\alpha_m = 0$ $(m = 2, 3, \cdots, s, \cdots)$, the combined degree is the node degree in common sense, if $\alpha_2 = 1$ and $\alpha_m = 0$ $(m = 1, 3, \cdots, s, \cdots)$, the combined degree is the 2-order degree, and so forth. For the values of parameters, we usually consider the case $\alpha_1 \geq \alpha_2 \geq \alpha_3 \geq \cdots$, where $\alpha_1 \geq \alpha_2$ indicates that a single neighbor's influence to v_i is no less than a single neighbor's influence. Notice that the value of constant α_m $(m = 1, 2, \cdots, s, \cdots)$ may not be integers; therefore the combined degree $d_G^{Com}(v_i)$ usually not be integer. Since the combined degree is the combination of regular degree and various high-order degree, we need not discuss the mean combined degree and combined degree distribution. Same as above, we can define combined in-degree and combined out-degree of a directed network. According to Theorem 1, we could give the following formula of combined degree sequence. Here we omit the formulas of combined in-degree sequence and combined out-degree sequence of directed networks.

$$d_G^{Com} = \left(\alpha_1 \cdot E_n + \alpha_2 \cdot A + \cdots + \alpha_s \cdot A^{s-1} + \cdots\right) d^1, \quad (2)$$

where E_n is the identity matrix of order n.

3. Results

In order to compare the behavior of degree and high-order degree and/or combined degree, we apply these attributes to a couple of best-known networks. We believe that the attributes of network with better behavior should be better to discover the differences among different nodes. By the definition of the differences of a given set of data, standard deviation (std) is the most important metric parameter. The greater the standard deviation is, the more diversities the data has, and the better the discrimination is, the better the attributes behavior is. As a complement to std, we define a novel parameter and call it as overflow ratio (OR_α), which denotes the ratio of the number of elements outside $[\alpha \cdot mean, 1/\alpha \cdot mean]$ to the number of elements in the given set of data, where $mean$ denotes the mean value of the given set of data and $0 < \alpha < 1$. The greater the overflow ratio is, the less the possibility that data cluster around the mean value of the given set is, and the better the discrimination is, the better the attributes behavior is. Therefore, these two parameters both reflect the diversity of a given set of data to some extent. Clearly, if one of the network attributes shows more diversity, it is easier to distinguish or sort the nodes.

3.1. Random Network. Random network was put forward for the first time by Paul Erdös and Alfred Rényi in 1960 (which is called *ER* Random Network) [11]. Here we generate a random network with n nodes, and we investigate the influence of node connection probability p taking different values on degree and high-order degree. Figure 3 shows the simulation results of 1000 times for node degree distribution, 2-order degree distribution, 4-order degree distribution, and combined degree distribution with $n = 10000$ and $p = 0.05$. We can tell from the results that the two high-order degrees and combined degree of random network still preserve the property of degree distribution, which is similar to Poisson Distribution (or Normal Distribution) with mean degree/mean high-order degree as the peak value.

Table 2 gives the discrimination of some attributes with $n = 10000$ and $p = 0.05$. Since the node degree distribution and high-order degree/combined degree distribution are all similar to Poisson Distribution (or Normal Distribution), we take big values for the overflow ratios. We can also tell from Table 2 that high-order degree has better discrimination compared with regular node degree, which is also true when n and p change (see Supplementary Table 1).

3.2. Small World Model. Watts [57] gave the basic attributes of Small World Model in 1999. Watts and Stregatz [6] proposed the construction of small world network by edge redistribution, which is called the *WS* small world network. Monasson

Table 2: Random Network: the discrimination parameters of 10000 simulations for different stages in Random Network, with number of nodes $n = 1000$ and connection probability $p = 0.05$.

Node degree	Standard Deviation	overflow ratio $\alpha = 0.9$	$\alpha = 0.91$	$\alpha = 0.93$	$\alpha = 0.95$	$\alpha = 0.98$
degree	2.80 ∗ 10e1	1.5%	2.73%	9.1%	23.08%	63.01%
2-order degree	1.093 ∗ 10e4	1.55%	2.86%	9.1%	23.86%	64.48%
4-order degree	2.747 ∗ 10e9	1.57%	2.9%	9.21%	23.96%	64.54%
combined degree	3.440 ∗ 10e8	1.57%	2.9%	9.21%	23.96%	64.54%

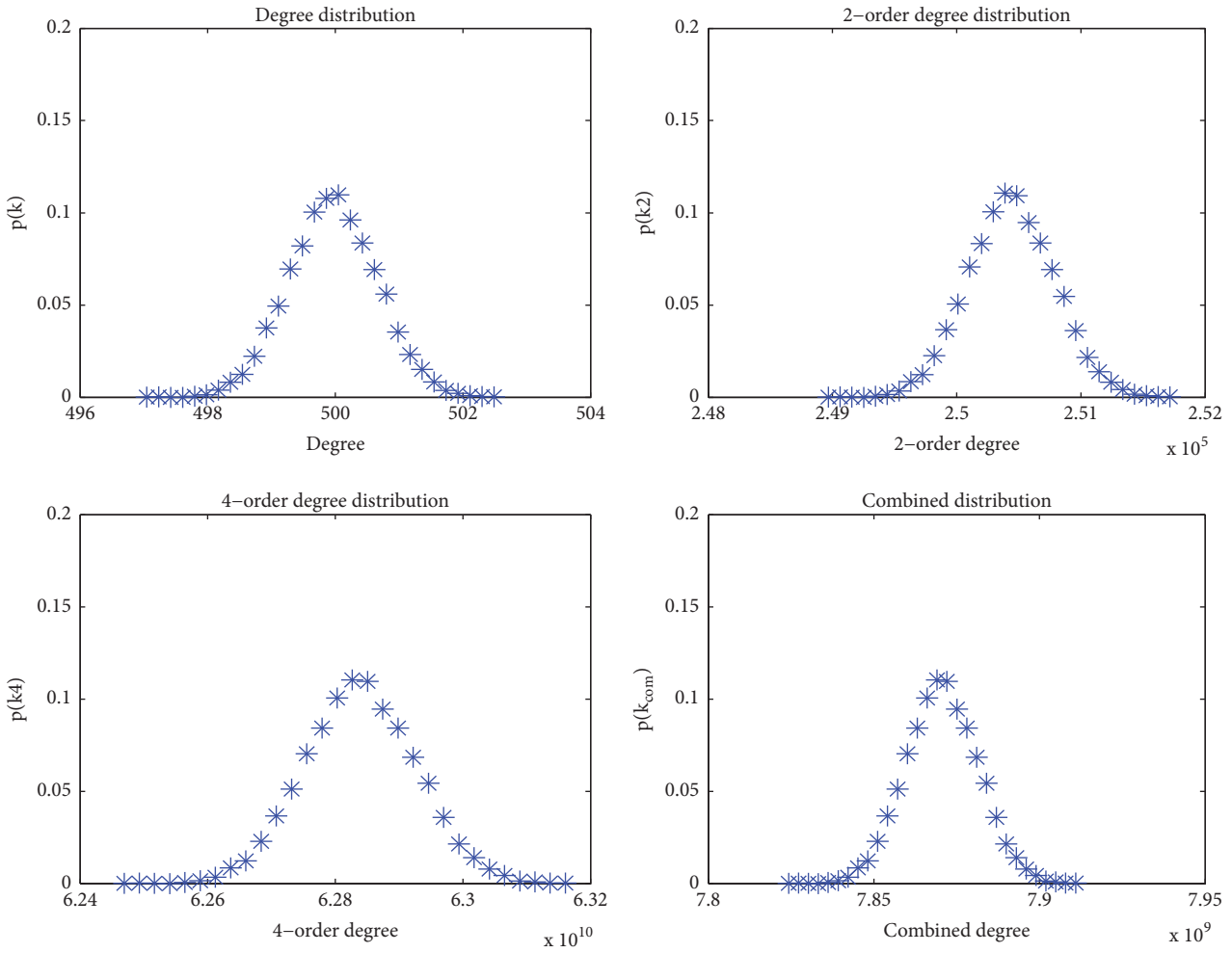


Figure 3: Random network: simulation results of 1000 times for node degree (mean) distributions with number of nodes $n = 10000$ and edge augment connection probability $p = 0.05$, taking $\alpha_1 = 1/2, \alpha_2 = 1/4, \alpha_3 = \alpha_4 = 1/8, \alpha_5 = \cdots = 0$ for combined degree distribution.

[58] and Newman et al. [59] initiated the construction of small world network through edge augment, which is called the *NW* small world network. In this subsection we construct a small world network by edge augment, and investigate the behaviors of high-order degrees in small world network. Firstly, we construct a regular network with n nodes, and with each node connected to the nearest d nodes (where $n \gg d \gg \ln n \gg 1$). For a given probability p, we stochastic add $pnd/2$ edges to produce the small world network. We can tell from Figure 4 that the scatter plot of 2-order degree distribution, 3-order degree distribution and 4-order degree distribution are all similar to Poisson Distribution (or Normal Distribution), which is the same as the degree distribution of the small world network.

For small world network, similar to random network, we are more concerned with the discrimination of small world network. Table 3 shows the mean discrimination of 1000 simulations for degree, 2-order degree, 4-order degree and 8-order degree, with $n = 800$, initial edge connection number $d = 30$, and connection probability $p = 0.05$. Of all

Table 3: Small world network: the discrimination parameters of 1000 simulations for different stages in small world network with $n = 5000$, $p = 0.05$, and $d = 20$.

order	Standard Deviation	overflow ratio			
		$\alpha = 0.85$	$\alpha = 0.9$	$\alpha = 0.93$	$\alpha = 0.95$
degree	1.2476	0.05%	0.6102%	7.2873%	7.2954%
2-order degree	41.321	0.15%	1.57%	5.9382%	5.9425%
4-order degree	45674	0.21%	1.8%	9.87%	9.76%
8-order degree	5.268e + 010	0.4%	4%	16.8%	32.875%

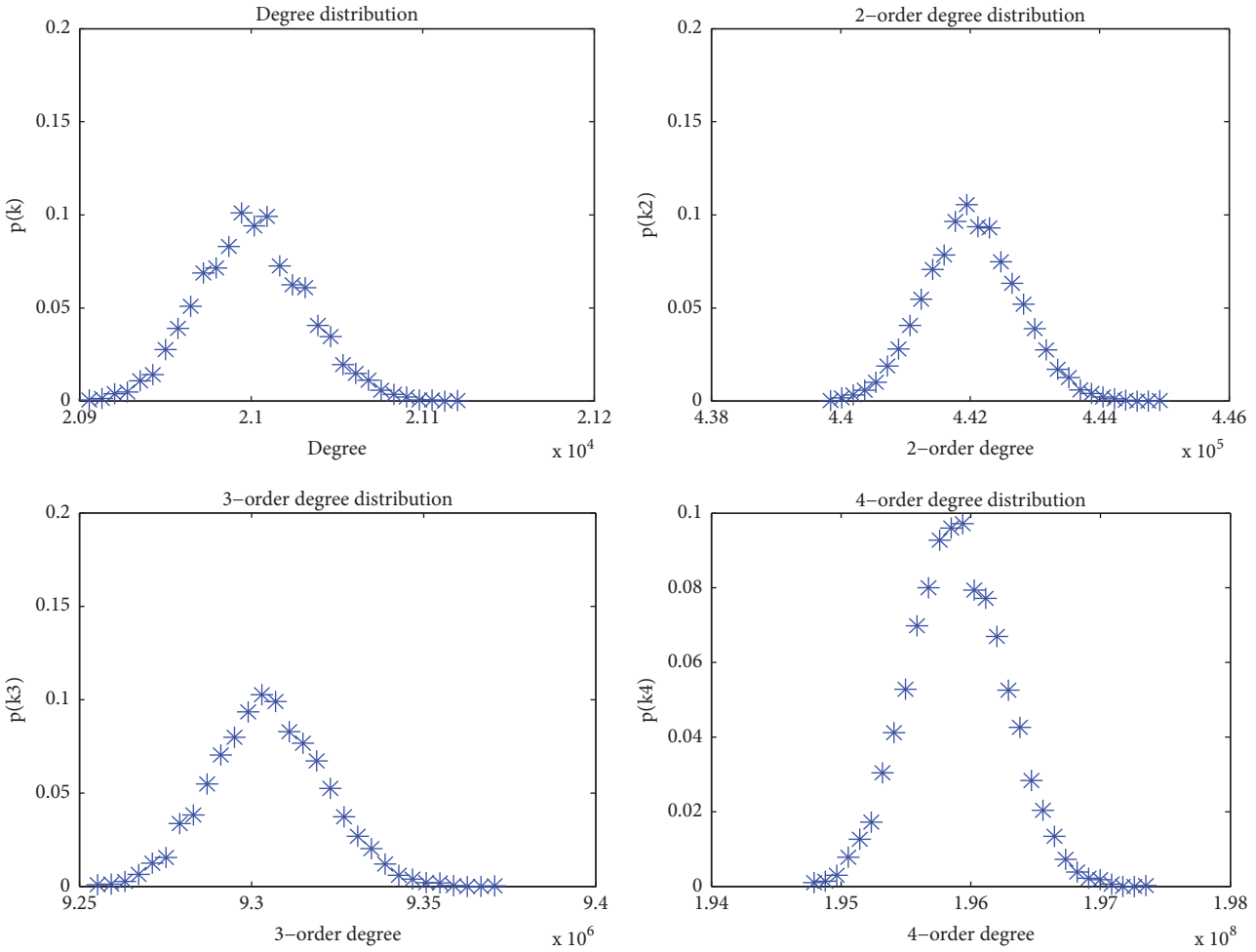


Figure 4: Small world network: simulation results of 1000 times for node degree distributions with number of nodes $n = 5000$ and given probability $p = 0.05$ and with each node connected to the nearest $d = 20$ nodes initially.

the situations, 4-order degree and 8-order degree show better discrimination results than the others. But sometimes the 2-order degree does not show good results in discrimination than degree, which is probably caused by the smallness of the value of parameter p. In Supplementary Table 2, we increase the value of p, and show the discrimination of degree, 2-order degree, and 4-order degree, with the number of nodes n and initial connection edges d change over 54 cases. Among all of the cases, the standard deviation discrimination gets bigger (better), 4-order degree shows smaller overflow ratio twice than that of node degree, and 2-order degree shows smaller overflow ratio four times than that of node degree, which suggests that 4-order degree and 2-order degree are better than node degree in the aspect of overflow ratio. Moreover, as the order of high-order degree increases, the overflow ratio shows better results in discriminating the nodes.

3.3. Undirected Scale-Free Network. Barabási and Albert [18, 60, 61] proposed scale-free network (which is called *BA* scale-free network) with node degree distribution following the power-law distribution. In this subsection, we construct an undirected, scale-free network to investigate the behavior

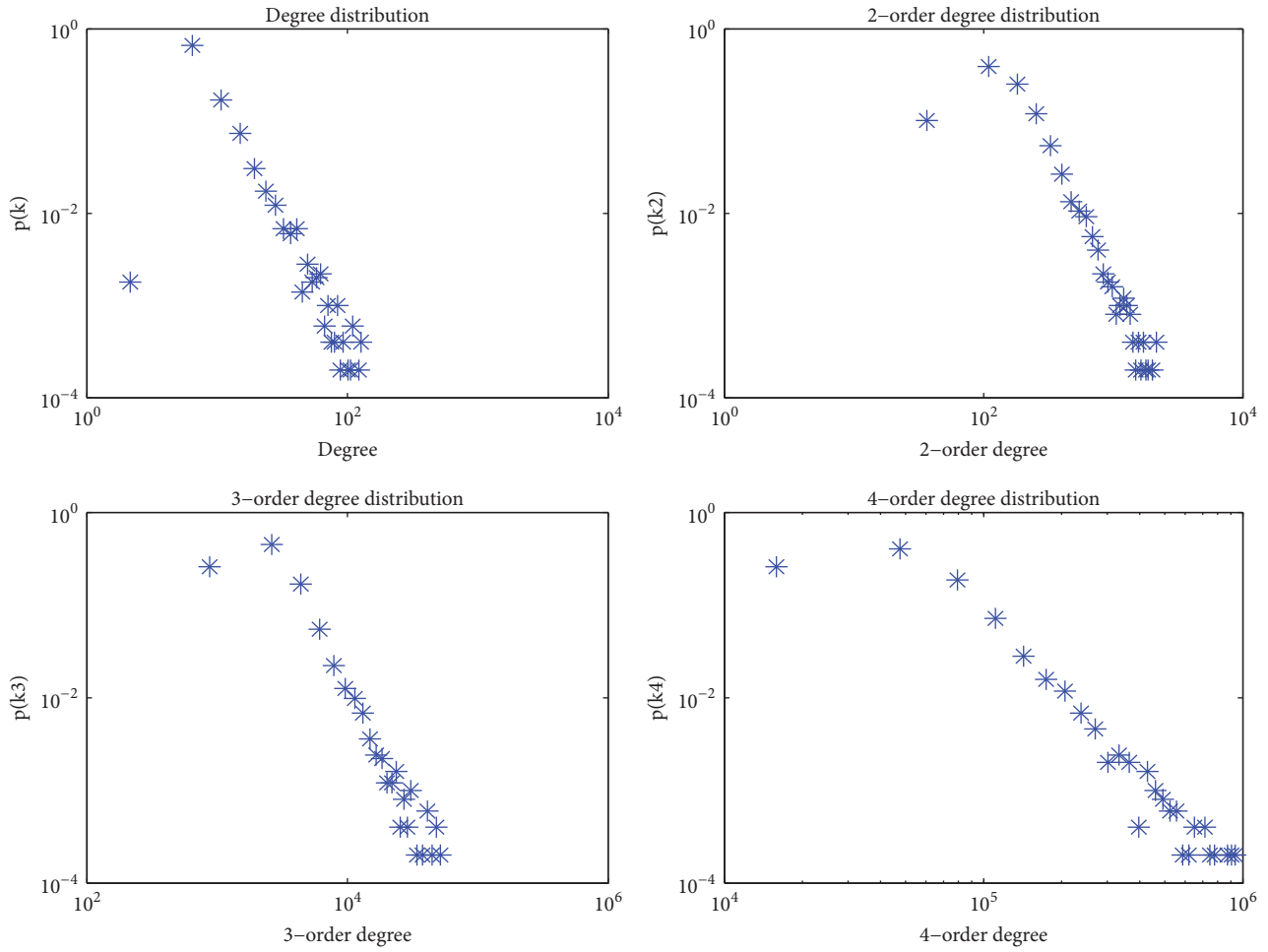


FIGURE 5: Undirected scale-free network: the node degree distribution and high-order degree distribution of a scale-free network, with $n = 100$, $p = 0.05$, $m = 5$, $t = 4900$.

of various orders of degree. Firstly, we produce a random network with the number of nodes n, stochastic connection probability p. Then we add one node each time whose degree equals m by the preferential attachment mechanism. Preferential attachment means that the more connected a node is, the more likely it is to receive new links. Nodes with higher degree have stronger ability to grab links added to the network. In our paper, the probability that new node connects to node i is proportional to the degree of i. Repeating this process for t times, we get an undirected scale-free network with $n + t$ nodes and about $np/2 + mt$ edges.

The high-order degree distribution and combined degree distribution are also power-law distribution; see Figure 5. Supplementary Table 3 gives the discrimination results of various orders of degree, with initial number of nodes n, initial node connection probability p, and running times t, and the new node added to the network each time has a degree of m. As a matter of fact, only if $0.3 \leq \alpha \leq 0.5$, the higher-order degree shows better discrimination results. The reason maybe that the range of node degree sequence is relatively large for scale-free network (compared with random network and small world network), which coincides with our knowledge [18, 60, 61].

3.4. Directed Scale-Free Network. In this subsection, we construct a directed scale-free network. Firstly, produce a random network with nodes number n and connection probability p. Secondly assign the direction of each existing edge at random (with equal probability) to establish a directed network. Thirdly, run this step t times, add a new node at each time, and add m_1 in-edges by out-degree priority mechanism and m_2 out-edges by in-degree priority mechanism; there we usually choose $m_2 \geq m_1 \geq 0$. Therefore we obtain a directed network with $n + t$ nodes and $np/2 + (m_1 + m_2)t$ edges. Figure 6 shows that both the in-degree distribution and the out-degree distribution in the directed network we developed follow power-law distribution; therefore it is a directed scale-free network [56, 57]. But there are variations at the beginning of the graphs for high-order in-degree and high-order out-degree (especially the out-degree), even though the general shapes of these graphs still resemble the power-rule distributions; see Figure 6. We suspect that this may be

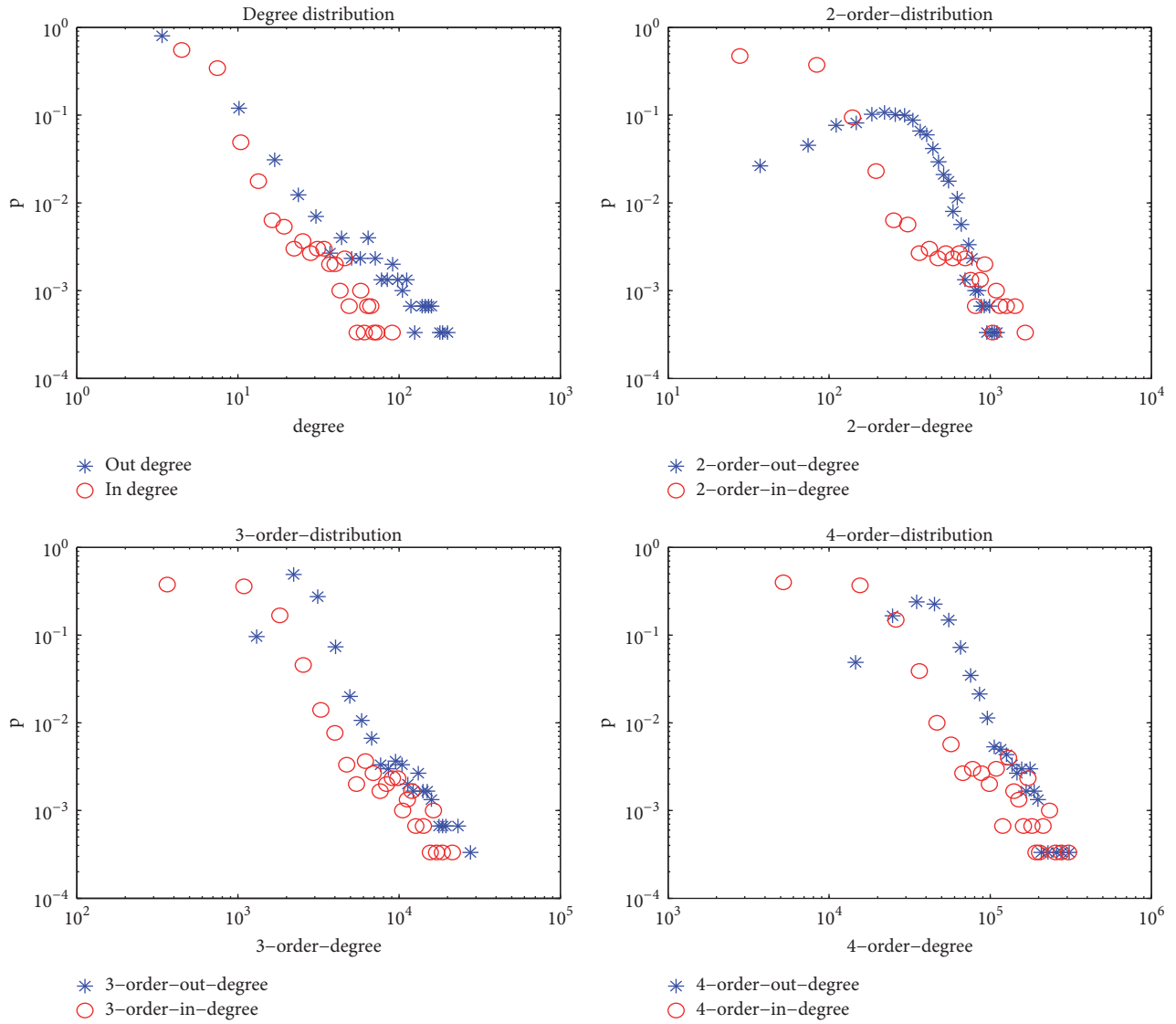


FIGURE 6: Directed scale-free network: node degree distribution and high-order distribution in the directed scale-free network with initial nodes $n = 100$, initial connection probability $p = 0.1$, the direction of an edge is assigned randomly, and $m_1 = 2$, $m_2 = 5$ in-edges/out-edges are added to network by the priority mechanism, running times $t = 2900$.

caused by the fact that we always add in-edges and out-edges at each time when we construct the network, and the numbers of edges maintain constants rather than random numbers. Since the discrimination comparison between high-order degree and node degree in directed scale-free network is similar to that in undireceted scale-free network, we will not demonstrate the results here.

4. Discussion

The high-order degree in a network considers the influence of different path distances on a node. In a social network, consider a node is a person, and an edge is a friendship. Node degree of node v_a indicates the influence of v_a's friends on v_a. The 2-order degree of v_a indicates the influence of v_a's friends' friends on v_a. An interesting question is that, if v_a and v_b are friends, there is an undirected edge connecting v_a and v_b, then v_a will influence v_b, which will in return have influence on v_a itself also. If v_a has many friends, v_a will have influence on all of its friends, which in return will affect v_a many times.

We know from Section 3 that high-order degrees and combined degree are superior to degree in discriminating nodes. However the computation of degree is simpler than high-order degrees and combined degree. Compared with the eigenvector centrality in undirected network, and PageRank in directed network, the high-order (out-/in-)degree and/or combined (out-/in-)degree we defined in this paper do not need iterations. The computation of our method only involves multiplications of matrices and vectors, which is clearly easier than eigenvector centrality or PageRank.

Table 4: Different centralities of Figure 2.

	Name of each node	1	2	3	4	5	6	7	8	9	10
Figure 2(a)	4-order degree	21	21	21	26	50	35	29	29	29	29
	Betweenness	0	0	0	21	20	26	0	0	0	0
	Eigenvector centrality	0.0547	0.0547	0.0547	0.1296	0.1431	0.2096	0.0884	0.0884	0.0884	0.0884
Figure 2(b)	3-order in-degree	0	0	1	2	0	2	3	4	4	6
	PageRank*	0	0	0	0.1795	0	0.1154	0.1795	0.1154	0.2308	0.1795
	PageRank**	0.0150	0.0318	0.0610	0.1255	0.0505	0.1134	0.1417	0.1283	0.1827	0.1501

* PageRank without damping factor.
** PageRank with damping factor equal to 0.85.

4.1. The High-Order Degree and Eigenvector Centrality/Betweenness in Undirected Network. In the undirected network, node degree is used to measure the influence of all the nodes connected to a node. 2-order degree considers the influence of a node's neighbor's neighbor on the node. 3-order degree considers the influence of a node's neighbors' neighbors' neighbor on that node. As a result, the lower the order of the high-order degree is, the closer it is to the node degree. On the other hand, the higher the order of the high-order degree is, the closer it is to eigenvector centrality.

Theorem 2. *For s-order degree in undirected network, if $s = 1$, it is the node degree of the network. As s approaches to infinity, s-order degree is equivalent to eigenvector centrality.*

We only need to show that when s approaches to infinity, s-order degree is equivalent to eigenvector centrality. Suppose A is the adjacency matrix in an undirected network and d^1 is the degree sequence. Then by Theorem 1, the s-order degree sequence is $A^{s-1} * d^1$, which is consistent with calculating the largest eigenvalue and the corresponding eigenvector by the method of power rule [62]. Therefore, the normalized result of $A^{s-1} * d^1 (s \to +\infty)$ is the node eigenvector centrality. If the eigenvector centrality is deemed as the most accurate method in ranking nodes, then from node degree to high-order degree, and to eigenvector centrality, the accuracies get higher and higher, and the computation complexity gets higher and higher at the same time. Therefore, if there is not high requirement on ranking accuracy and computation complexity, high-order degree is a relatively good choice.

In fact, high-order degrees are not simple alternative to eigenvector centrality. Sometimes high-order degrees may be more intuitive than some methods including eigenvector centrality. For example, Figure 2(a), the top three nodes sorted by both 2-order degree and 4-order degree are 5, 6, 4, which we can see from Tables 1 and 4; it is the same as our supposition. Otherwise the top three nodes sorted by both betweenness and eigenvector centrality are 6, 5, 4.

4.2. The High-Order Degree and PageRank for Directed Network

Theorem 3. *For s-order out-degree in a directed network, if $s = 1$, it is the out-degree of the directed network. For s-order in-degree in a directed network, if $s = 1$, it is the in-degree of the directed network. When s approaches to infinity, the s-order in-degree is equivalent to PageRank (without damping factor).*

Theorem 3 is clearly true because PageRank is the simplification of eigenvector centrality in directed network [63]. Moreover, we can reach similar conclusions that when ranking network nodes, high-order in-degree is a good choice if one has certain but not high requirements on the accuracy and computation complexity.

Same as Section 4.2, high in-order degrees are not simple alternative to PageRank. Sometimes high in-order degrees may be more intuitive than PageRank. For example Figure 2(b), the top node sorted by both 2-order in-degree and 3-order in-degree is 9, which we can see from Tables 1 and 4, it is the same with our supposition. Otherwise the top node sorted by both PageRank without damping factor and PageRank with damping factor is 9.

4.3. The Significance of Network High-Order Out-Degree. PageRank (high-order in-degree) indicates that the *PR* value of a node v_a is larger when there are more nodes pointing to v_a in a directed network (quantity hypothesis) and/or the nodes pointing v_a have larger *PR* values (quality hypothesis) [35, 36]. Reversely, we call high-order out-degree Reverse PageRank (*RPR*). The *RPR* value of a node v_b in a directed network is determined by how many nodes pointed by v_b (reverse quantity hypothesis) and/or how large *RPR* value that nodes pointed by v_b have (reverse quality hypothesis). For example, we can say that the World Wide Web navigation websites have large *RPR* value. When someone surfs the Internet, if he does not know which website is worth visiting (with larger *PR* value), he should start with the navigation website (with larger *RPR* value). There is more research on the high-order in-degree (*PR*) for directed network, but there is no research on high-order out-degree (*RPR*) as far as we know.

Figure 2(b), although nodes 1, 2, 5 have the same 2-order out-degree, only node 1 has the largest n-order out-degree ($n \geq 3$). Node 1 is the only one that can reach any nodes of this graph.

4.4. The Combined Degree We Defined and the Mixed Degree Zeng Defined. Zeng [44] used the weighted sum of both the residual degree and the exhausted degree to define the mixed degree of the node of a network. On one hand, Zeng's definition involves Mixed Degree Decomposition (*MDD*) to

the whole network by using coreness centrality, while the combined degree we defined is based on a linear combination of various high-order degrees. Even though our method involves more terms in the weighted sum, it does not need to consider network decomposition. Therefore it is cheaper in computation than Zeng's definition. On the other hand, Zeng divides the nodes that are connected to a node into two classes according to *MDD*, while we divide the nodes that are connected to a node into finite number of classes according to path distance, which is more delicate in classification and has more accurate results.

5. Conclusion

In this paper, we define several novel centrality metrics: the high-order (out-/in-)degree and combined degree. For the values of combined degree's parameters, we usually consider the case $\alpha_1 \geq \alpha_2 \geq \alpha_3 \geq \cdots \geq 0$ and $\sum_i \alpha_i = 1$. We prove that both the degree centrality and eigenvector centrality are the special cases of the high-order degree of undirected network, and both the in-degree and PageRank algorithm without damping factor are the special cases of the high-order in-degree of directed network. We present several experiments to discuss the performance of our novel centrality metrics. It can be seen from the experiments that the centrality metrics we defined are easy to calculate and perform better than degree centrality. In a large-scale complex network study, our centrality metrics will be an effective alternative to the eigenvector centrality/PageRank algorithm. The manuscript is only limited in introducing the definition of new metrics. We hope to discuss their efficacy and computational cost in the further works.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

All authors worked together to produce the results and read and approved the final manuscript.

Acknowledgments

The research is supported by the National Natural Science Foundation of China (61572522 and 11371230) and Shandong Provincial Natural Science Foundation (ZR2018PF004).

Supplementary Materials

Supplementary 1. Supplementary Table 1: in the random network, we increase the value of p and show the discrimination of degree, 2-order degree, and 4-order degree, with the number of nodes n and initial connection edges d changing over 27 cases.

Supplementary 2. Supplementary Table 2: in the small world network, we increase the value of p and show the discrimination of degree, 2-order degree, and 4-order degree, with the number of nodes n and initial connection edges d changing over 54 cases.

Supplementary 3. Supplementary Table 3: in the undirected scale-free network, the discrimination results of various orders of degree, with initial number of nodes n, initial node connection probability p, running times t, and the new node added to the network each time have a degree of m over 90 cases.

References

[1] M. E. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[2] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 47–97, 2002.

[3] A. Rapoport, "Contribution to the theory of random and biased nets," *Bulletin of Mathematical Biology*, vol. 19, no. 4, pp. 257–277, 1957.

[4] A. Rapoport and W. J. Horvath, "A study of a large sociogram," *Behavioural Science*, vol. 6, pp. 279–291, 1961.

[5] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, Cambridge University Press, Cambridge, UK, 1994.

[6] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.

[7] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin, "Size-dependent degree distribution of a scale-free growing network," *Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 63, no. 6, 2001.

[8] J. Lee, C. Heaukulani, Z. Ghahramani, L. F. James, and S. Choi, "Bayesian inference on random simple graphs with power law degree distributions," in *Proceedings of the International Conference on Machine Learning*, pp. 2004–2013, 2017.

[9] F. Mohd-Zaid, C. M. S. Kabban, R. F. Deckro, and E. D. White, "Parameter specification for the degree distribution of simulated Barabási-Albert graphs," *Physica A: Statistical Mechanics and Its Applications*, vol. 465, pp. 141–152, 2017.

[10] X. Wang, S. Trajanovski, R. E. Kooij, and P. Van Mieghem, "Degree distribution and assortativity in line graphs of complex networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 445, pp. 343–356, 2016.

[11] P. Erdös and A. Rényi, "On the evolution of random graphs," *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, vol. 38, pp. 17–61, 2012.

[12] P. Erdös, "On extremal problems of graphs and generalized graphs," *Israel Journal of Mathematics*, vol. 2, pp. 183–190, 1964.

[13] R. Albert, H. Jeong, and A.-L. Barabási, "Internet: diameter of the World-Wide Web," *Nature*, vol. 401, no. 6749, pp. 130-131, 1999.

[14] V. M. Eguíluz and K. Klemm, "Epidemic threshold in structured scale-free networks," *Physical Review Letters*, vol. 89, no. 10, Article ID 108701, 2002.

[15] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, pp. 651–654, 2002.

[16] J. Rung, T. Schlitt, A. Brazma, K. Freivalds, and J. Vilo, "Building and analysing genome-wide gene disruption networks," *Bioinformatics*, vol. 18, no. 2, pp. S202–S210, 2002.

[17] S. Battiston, J. F. Rodrigues, and H. Zeytinoglu, "The network of inter-regional direct investment stocks across Europe," *Advances in Complex Systems (ACS)*, vol. 10, no. 1, pp. 29–51, 2007.

[18] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *American Association for the Advancement of Science: Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[19] L. D. F. Costa, O. N. Oliveira, G. Travieso et al., "Analyzing and modeling real-world phenomena with complex networks: a survey of applications," *Advances in Physics*, vol. 60, no. 3, pp. 329–412, 2011.

[20] L. Li, Y. Wei, C. To et al., "Integrated Omic analysis of lung cancer reveals metabolism proteome signatures with prognostic impact," *Nature Communications*, vol. 5, article no. 5469, 2014.

[21] Y. Zhang, M. Zhao, J. Su, X. Lu, and K. Lv, "Novel model for cascading failure based on degree strength and its application in directed gene logic networks," *Computational & Mathematical Methods in Medicine*, vol. 2018, Article ID 8950794, 2018.

[22] S. Wang, Y. Chen, Q. Wang, E. Li, Y. Su, and D. Meng, "Analysis for gene networks based on logic relationships," *Journal of Systems Science & Complexity*, vol. 23, no. 5, pp. 999–1011, 2010.

[23] H. Jeong, S. P. Mason, A. L. Barabási, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, no. 6833, pp. 41-42, 2001.

[24] Y. Zhang, K. Lv, S. Wang, J. Su, and D. Meng, "Modeling gene networks in saccharomyces cerevisiae based on gene expression profiles," *Computational and Mathematical Methods in Medicine*, vol. 2015, Article ID 621264, 10 pages, 2015.

[25] M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[26] W. Wang and T. Zhang, "Caspase-1-mediated pyroptosis of the predominance for driving CD_4^+ T cells death: a nonlocal spatial mathematical model," *Bulletin of Mathematical Biology*, vol. 80, no. 3, pp. 540–582, 2018.

[27] M. Rubinov and E. Bullmore, "Schizophrenia and abnormal brain network hubs," *Dialogues in Clinical Neuroscience*, vol. 15, no. 3, pp. 339–349, 2013.

[28] J. Bohannon, "Counterterrorism's new tool: 'metanetwork' analysis," *Science*, vol. 325, no. 5939, pp. 409–411, 2009.

[29] T. Zhang, X. Meng, and T. Zhang, "Global dynamics of a virus dynamical model with cell-to-cell transmission and cure rate," *Computational and Mathematical Methods in Medicine*, Article ID 758362, 8 pages, 2015.

[30] D. W. Franks, J. Noble, P. Kaufmann, and S. Stagl, "Extremism propagation in social networks with hubs," *Adaptive Behavior*, vol. 16, no. 4, pp. 264–274, 2008.

[31] A. Bavelas, "A mathematical model for group structures," *Human Organization*, vol. 7, no. 3, pp. 16–30, 1948.

[32] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, pp. 581–603, 1966.

[33] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978.

[34] P. Bonacich, "Technique for analyzing overlapping memberships," *Sociological Methodology*, vol. 4, pp. 176–185, 1972.

[35] E. Costenbader and T. W. Valente, "The stability of centrality measures when networks are sampled," *Social Networks*, vol. 25, no. 4, pp. 283–307, 2003.

[36] L. Page, *The PageRank citation ranking: bringing order to the web, Stanford Digital Libraries Working Paper*, vol. 9, 1998.

[37] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks*, vol. 56, no. 18, pp. 3825–3833, 2012.

[38] M. A. Kłopotek, S. T. Wierzchoń, R. A. Kłopotek, and E. A. Kłopotek, "Traditional PageRank versus network capacity bound," in *Social and Information Networks*, 2017.

[39] T. H. Haveliwala, "Topic-sensitive PageRank," in *Proceedings of the 11th International Conference on World Wide Web, WWW '02*, pp. 517–526, May 2002.

[40] A. N. Langville and C. D. Meyer, "Deeper inside PageRank," *Internet Mathematics*, vol. 1, no. 3, pp. 335–380, 2004.

[41] M. Bressan, E. Peserico, and L. Pretto, "The power of local information in pagerank," in *Proceedings of the 22nd International Conference on World Wide Web, WWW 2013*, pp. 179-180, May 2013.

[42] I. M. Kloumann, J. Ugander, and J. Kleinberg, "Block models and personalized PageRank," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 114, no. 1, pp. 33–38, 2017.

[43] A. Altman and M. Tennenholtz, "Ranking systems: the PageRank axioms," in *Proceedings of the EC'05: 6th ACM Conference on Electronic Commerce*, pp. 1–8, June 2005.

[44] J. Ai, H. Zhao, K. M. Carley, Z. Su, and H. Li, "Neighbor vector centrality of complex networks based on neighbors degree distribution," *The European Physical Journal B*, vol. 86, no. 4, 2013.

[45] A. Zeng and C.-J. Zhang, "Ranking spreaders by decomposing complex networks," *Physics Letters A*, vol. 377, no. 14, pp. 1031–1035, 2013.

[46] J. Bae and S. Kim, "Identifying and ranking influential spreaders in complex networks by neighborhood coreness," *Physica A: Statistical Mechanics and Its Applications*, vol. 395, pp. 549–559, 2014.

[47] U. Brandes, "A faster algorithm for betweenness centrality," *Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[48] K. Goel, R. R. Singh, S. Iyengar, and S. Gupta, "A faster algorithm to update betweenness centrality after node alteration," *Internet Mathematics*, vol. 11, no. 4-5, pp. 403–420, 2015.

[49] R. Guns, Y. X. Liu, and D. Mahbuba, "Q-measures and betweenness centrality in a collaboration network: a case study of the field of informetrics," *Scientometrics*, vol. 87, no. 1, pp. 133–147, 2011.

[50] J. D. Noh and H. Rieger, "Random walks on complex networks," *Physical Review Letters*, vol. 92, no. 11, Article ID 118701, 2004.

[51] V. Tejedor, O. Bénichou, and R. Voituriez, "Global mean first-passage times of random walks on complex networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 80, no. 6, 2009.

[52] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 105, no. 4, pp. 1118–1123, 2008.

[53] J. Saramäki and K. Kaski, "Scale-free networks generated by random walkers," *Physica A: Statistical Mechanics and Its Applications*, vol. 341, no. 1-4, pp. 80–86, 2004.

[54] T. Weng, J. Zhang, M. Khajehnejad, M. Small, R. Zheng, and P. Hui, "Navigation by anomalous random walks on complex networks," *Scientific Reports*, vol. 6, 2016.

[55] E. K. Lloyd, J. A. Bondy, and U. S. Murty, "Graph theory with applications," *The Mathematical Gazette*, vol. 28, pp. 237-238, 1976.

[56] E. Terzi and M. Winkler, "A spectral algorithm for computing social balance," in *Algorithms and Models for the Web Graph*, vol. 6732 of *Lecture Notes in Comput. Sci.*, pp. 1–13, Springer, Heidelberg, 2011.

[57] J. Tang, Y. Chang, C. Aggarwal, and H. Liu, "A survey of signed network mining in social media," in *ACM Computing Surveys (CSUR)*, vol. 49, pp. 237-238, 2015.

[58] D. J. Watts, *Small Worlds: The Dynamics of Networks between Order and Randomness*, Princeton University Press, Princeton, NJ, USA, 1999.

[59] R. Monasson, "Diffusion, localization and dispersion relations on small-world lattices," *The European Physical Journal B*, vol. 12, pp. 555–567, 1999.

[60] M. E. Newman and D. J. Watts, "Renormalization group analysis of the small-world network model," *Physics Letters A*, vol. 263, no. 4-6, pp. 341–346, 1999.

[61] A. L. Barabási, R. Albert, and H. Jeong, "Mean-field theory for scale-free random networks," *Physica A Statistical Mechanics & Its Applications*, vol. 272, pp. 173–178, 1999.

[62] A. Barabási, E. Ravasz, and T. Vicsek, "Deterministic scale-free networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 299, no. 3-4, pp. 559–564, 2001.

[63] M. E. J. Newman, "The mathematics of networks," in *The New Palgrave Encyclopedia of Economics*, vol. 208, pp. 1–12, 2008.

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 8058670, 6 pages
https://doi.org/10.1155/2018/8058670

Research Article

Big Data Validity Evaluation Based on MMTD

Ningning Zhou,[1] **Guofang Huang**,[2] **and Suyang Zhong**[1]

[1]*School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210023, China*
[2]*State Key Laboratory of Smart Grid Protection and Control, Nanjing 211106, China*

Correspondence should be addressed to Ningning Zhou; zhounn@njupt.edu.cn

Received 7 November 2017; Revised 23 March 2018; Accepted 10 April 2018; Published 10 June 2018

Academic Editor: Ester Zumpano

Big data has been studied extensively in recent years. With the increase in data size, data quality becomes a priority. Evaluation of data quality is important for data management, which influences data analysis and decision making. Data validity is an important aspect of data quality evaluation. Based on 3V properties of big data, dimensions that have a major influence on data validity in a big data environment are analyzed. Each data validity dimension is analyzed qualitatively using medium logic. The measuring of medium truth degree is used to propose models to measure single and multiple dimensions of big data validity. The validity evaluation method based on medium logic is more reasonable and scientific than general methods.

1. Introduction

Big data has been studied extensively in recent years and several investigations have focused on the big data phenomenon[1–7]. The top international journals 'Nature'[8] and 'Science'[9], respectively, in 2008 and 2011, took 'big data' and 'dealing with data' as the topic, which made people explore the enthusiasm of big data. However, there is no universal definition of big data in academia. In a literal sense, the most fundamental nature of big data lies in the large data size, but it also involves a high degree of complexity associated with data collection, management, and processing. The "big" of big data is mainly reflected in three aspects [10–12]: (1) data volume is large (Volume); (2) the complexity of data type is high (Variety); (3) data flow, especially the generation of information flow in Internet, is fast (Velocity). The 3V properties have now been widely accepted to describe big data. Some people will also express the potential huge value of Value into it, so that 3V is extended to 4V.

Although big data is valuable, it is a challenge to unlock the potential from the large amount of data [13]. High quality is a prerequisite for unlocking big data potential since only a high-quality big data environment yields implicit, accurate, and useful information that helps make correct decisions. Even state-of-the-art data analysis tools cannot extract useful information from an environment fraught with "rubbish" [14, 15]. However, it is difficult to maintain high quality because big data is varied, complicated, and dynamic. This highlights a need for the analysis and evaluation of big data quality while constructing a high-quality big data environment.

Data quality involves many dimensions that include data validity, timeliness, fuzziness, objectivity, usefulness, availability, user satisfaction, ease of use, and understandability. Data validity is particularly important in the evaluation of data quality. It is a priority due to the massive data size, increased demand for data processing, and broad variety of data types. However, few studies have been done on the evaluation of data validity[16, 17]. Wei Meng proposed to measure data validity using the update frequency [18]. Update frequency of date is a dimension of the quality of data. However, this dimension reflects the novelty of the data rather than the validity. Qingyun et al. proposed to evaluate data validity by formulating a constraint in the dataset [19]. The constraint of evaluating data validity is whether it is in a range compliant with the truth or not. This constraint is one of the dimensions of data validity, but it is not comprehensive. In [20], Jie et al. proposed to devise constraints using three rules (i.e., static, transaction, and dynamic) and they evaluated data validity by measuring the degree to which the rules were satisfied. The method for data validity evaluation varies with the application. It focused on the restricting rules on GIS, but it is too special and it is not general. Moreover, due to the

special attributes of big data, these methods are not entirely suitable for big data. To the best of our knowledge, there is no method for qualitative and quantitative analysis of big data validity.

In this paper, first, we comprehensively analyze dimensions that have a major influence on data validity based on the 3V properties of big data. Data validity refers to the level of need that users or enterprise have for data. Completeness, correctness, and compatibility are particularly serious in a big data environment and become the primary factors that affect data validity. Hence, big data validity is measured in this paper from the perspectives of completeness, correctness, and compatibility. It is used to indicate whether data meets the user-defined condition or falls within a user-defined range. Next, a qualitative analysis of each dimension of data validity is performed using medium logic. Finally, the measure of medium truth degree (MMTD) is used to propose models to measure single and multiple dimensions of big data validity. Our Model for measuring one dimension of big data validity is based on medium logic. Logical correctness ensures that the evaluation results are more reasonable and scientific.

2. Overview of Medium Mathematics Systems

Medium principle was established by Wujia Zhu and Xi'an Xiao in 1980s who devised medium logic tools [21] to build the medium mathematics system, the corner stone of which is medium axiomatic sets [22].

2.1. Notations for Medium Mathematics Systems. In medium mathematics system [21], predicate (concept or property) is represented by P; any variable is denoted as x, with x completely possessing property P being described as P(x). The "¬ "symbol stands for inverse opposite negative and it is termed as "opposite to". The inverse opposite of predicate is denoted as ¬ P. Then the concept of a pair of inverse opposite is represented by both P and ¬ P. Symbol "~"denotes fuzzy negative which reflects the medium state of "either or" or "both this and that" in opposite transition process. The fuzzy negative profoundly reflects fuzziness; "$\prec$" is a truth-value degree connective which describes the difference between two propositions.

2.2. Measuring of Medium Truth Degree

2.2.1. Measuring of Individual Medium Truth Degree. According to the concept of super state[23], the numerical value area of generally applicable quantification is divided into five areas corresponding to the predicted truth scale, namely ¬ ^{+}P, ¬ P, ~P, P, and ^{+}P. In "true" numerical value area T, α_T is ε_T standard scale of predication P; In "false" numerical value area F, $\alpha_F$is ε_F standard scale of predicate ¬ P. $f(x)$ is an arbitrary numeric function of variable x. According to the numeric interval of $f(x)$, the distance ratio function h_T(or h_F) which can scale the individual truth degree is defined. Adopting the concept of distance and using length of numerical value interval to different predicate truth as norm, the distance ratio function is defined, and from this the individual truth degree function is established as follows [23].

For $f(X)\to R$ and $y= f(x) \in f(X)$, the distance ratio $h_T(y)$ which relates to P is

$$h_T(y) = \begin{cases} \dfrac{-d(y, \alpha_F - \varepsilon_F)}{d(\alpha_T - \varepsilon_T, \alpha_F - \varepsilon_F)} & y < \alpha_F - \varepsilon_F \\ 0 & \alpha_F - \varepsilon_F \le y \le \alpha_F + \varepsilon_F \\ \dfrac{d(y, \alpha_F + \varepsilon_F)}{d(\alpha_T - \varepsilon_T, \alpha_F + \varepsilon_F)} & \alpha_F + \varepsilon_F < y < \alpha_T - \varepsilon_T \\ 1 & \alpha_T - \varepsilon_T \le y \le \alpha_T + \varepsilon_T \\ \dfrac{d(y, \alpha_F + \varepsilon_F)}{d(\alpha_T + \varepsilon_T, \alpha_F + \varepsilon_F)} & y > \alpha_T + \varepsilon_T. \end{cases} \tag{1}$$

For $f(X)\to R$ and $y= f(x) \in f(X)$, the distance ratio $h_F(y)$ which relates to ¬ P is

$$h_F(y) = \begin{cases} \dfrac{-d(y, \alpha_T + \varepsilon_T)}{d(\alpha_T + \varepsilon_T, \alpha_F + \varepsilon_F)} & y > \alpha_T + \varepsilon_T \\ 0 & \alpha_T - \varepsilon_T \le y \le \alpha_T + \varepsilon_T \\ \dfrac{d(y, \alpha_T - \varepsilon_T)}{d(\alpha_T - \varepsilon_T, \alpha_F + \varepsilon_F)} & \alpha_F + \varepsilon_F < y < \alpha_T - \varepsilon_T \\ 1 & \alpha_F - \varepsilon_F \le y \le \alpha_F + \varepsilon_F \\ \dfrac{d(y, \alpha_T - \varepsilon_T)}{d(\alpha_T - \varepsilon_T, \alpha_F - \varepsilon_F)} & y > \alpha_F - \varepsilon_F \end{cases} \tag{2}$$

where $d(a,b)$ is the Euclidean distance.

The bigger the value of $h_T(y)$ is, the higher the individual truth degree related to P is. The bigger the value of $h_F(y)$ is, the higher the individual truth degree related to ¬ P is.

2.2.2. Measuring of Set Medium Truth Degree. $f{:}X\to R^n$ is the n-dimensional numerical mapping of the set X. The measuring of truth scale of disperse set X which relates to P (or ¬ P) can be scaled by the additivity of the truth scale [23, 24]$h_{nT-S}(y_i)$ (or $h_{nF-S}(y_i)$) and the average additivity of the truth scale [23, 24]$h_{nT-M}(y_i)$ (or$h_{nF-M}(y_i)$) of set which relates to P (or ¬ P).

When $y_i = (f_1(x_i), f_2(x_i), \ldots, f_n(x_i)) = (y_{i1}, y_{i2}, \ldots, y_{in}) \in f(X)$, the additivity of the truth degree of disperse set X which relates to P is

$$h_{nT-S}(y_i) = \sum_{k=1}^{n} (h_T(y_{ik})). \tag{3}$$

The average additivity of the truth degree of disperse set X which relates to P is

$$h_{nT-M}(y_i) = \frac{1}{n}\sum_{k=1}^{n} (h_T(y_{ik})). \tag{4}$$

The additivity of the truth degree of disperse set X which relates to ¬ P is

$$h_{nF-S}(y_i) = \sum_{k=1}^{n} (h_F(y_{ik})). \quad (5)$$

The average additivity of the truth degree of disperse set X which relates to ¬ P is

$$h_{nF-M}(y_i) = \frac{1}{n}\sum_{k=1}^{n} (h_F(y_{ik})). \quad (6)$$

3. Qualitative Analysis of Big Data Validity

Data validity refers to the degree of data demand for users or enterprises. It is used to describe whether data satisfies user-defined conditions or falls within a user-defined range.

3.1. Selection of Dimension for Big Data Validity Evaluation. A large amount of incompatible data is generated due to the 3V properties of big data. Furthermore, data correctness and completeness can be compromised during generation, transmission, and processing. These problems are particularly serious in a big data environment and become the primary factors that affect data validity. Hence, big data validity is measured in this paper from the perspectives of completeness, correctness, and compatibility.

3.2. Dimensions of Big Data Validity

3.2.1. Data Completeness. In Cihai (an encyclopedia of the Chinese language), completeness refers to the state where components or parts are maintained without being damaged. In the Collins English Dictionary and Oxford Dictionary, completeness is defined as the state including all the parts, etc., that are necessary: whole. In the 21st Century Unabridged English-Chinese Dictionary, completeness means including all parts, details, facts, etc. and with nothing missing.

A universal definition of big data completeness is lacking. In the context of a specific application, big data completeness can be defined as follows.

Definition 1. If data has n properties and each property has all necessary parts, it is regarded as complete. Otherwise, it is incomplete.

Definition 2. Completeness refers to the degree to which data is complete. It is denoted by *C1*.

Let $R_1, R_2, \ldots R_n$ denote the n data properties and $V(R_i)$ denote the completeness of property R_i. Note that R_i has different forms for different applications. For example, the completeness of a property is zero if the property value is missing for some data, and 1 otherwise. Hence, R_i can be defined as

$$V(R_i) = \begin{cases} 0, & R_i \text{ missing} \\ 1, & R_i \text{ exists.} \end{cases} \quad (7)$$

The importance of each data property varies with the application. Let $w_1, w_2 \ldots w_n$ denote the weights for n properties in an application, where

$$\sum_{i=1}^{n} w_i = 1. \quad (8)$$

Consider data with n properties; its completeness is computed as the weighted sum of the completeness of all its properties.

$$C1 = \sum_{i=1}^{n} V(R_i) \times w_i. \quad (9)$$

3.2.2. Data Correctness. In Cihai, correctness refers to compliance with truth, law, convention, and standard, contrary to "wrongness". In the Collins English Dictionary and Oxford Dictionary, correctness is defined as accurate or true, without any mistakes. In the 21st Century Unabridged English-Chinese Dictionary, completeness means accurate, compliant with truth, and having no mistakes.

Currently, there is no universal definition for data correctness in the field of big data. Whether data is correct and the degree to which data is correct are defined as follows from the perspective of the application.

Definition 3. Consider data with n properties. If each property is compliant with a recognized standard or truth, it is regarded as correct. Otherwise, it is incorrect.

Definition 4. Correctness refers to the degree to which data is correct. It is denoted by C2.

Let $R_1, R_2, \ldots R_n$ denote the n data properties and $Z(R_i)$ denote the correctness of property R_i. If the value of R_i is in a range compliant with the truth, the correctness of this property is 1. Otherwise, it is 0. The correctness of the property, $Z(R_i)$, is defined as

$$Z(R_i) = \begin{cases} 1, & R_i \in \text{dom}(R_i) \\ 0, & R_i \notin \text{dom}(R_i) \end{cases} \quad (10)$$

where $\text{dom}(R_i)$ denotes the range of R_i.

Data correctness C2 is computed as the weighted sum of each property:

$$C2 = \sum_{i=1}^{n} Z(R_i) \times w_i \quad (11)$$

where w_i denotes the weight of each property in the application and satisfies (8).

3.2.3. Data Compatibility. In Cihai, compatibility refers to coexistence without causing problems. In the 21st Century Unabridged English-Chinese Dictionary, compatibility means that ideas, methods, or things can be used together. In the case of big data, data compatibility is defined as follows.

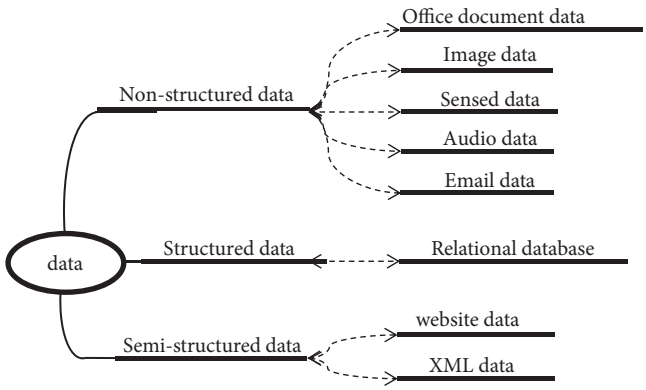


Figure 1: Data types in a big data environment.

Definition 5. If a group of data is of the same type and describes the same object consistently, the data is regarded as compatible with one another; otherwise, it is mutually exclusive.

Definition 6. Compatibility C3 refers to the degree to which a group of data is compatible with one another. Compatibility C3 is defined as

$$C3 = 1 - \frac{d_t}{d_a} \tag{12}$$

where d_a denotes the total amount of data in the group and d_t denotes the amount of incompatible data in the group.

4. Medium Truth Degree-Based Model for Measuring Big Data Validity

4.1. Data Normalization. Data variety is a significant aspect of big data. In addition to traditional structured data, a large amount of nonstructured and semistructured data has been generated by advances in the Internet and the Internet of Things (IoT). Examples include website data, sensed data, audio data, image data, and signal data, as shown in Figure 1. While this enriches content, it is more challenging to store, analyze, and evaluate data. Data needs to be normalized before appropriately evaluating big data validity.

Structured and nonstructured data in a big data environment have different content, forms, and structures, so they cannot be managed uniformly. Hence, a data model needs to be developed to provide a uniform description of both structured and nonstructured data.

Based on [25], a tetrahedron data model is proposed for nonstructured data. The proposed model consists of four parts: basic property, semantic feature, bottom-layer feature, and original document. In order to process structured and nonstructured data uniformly, a new part of data type is introduced to describe document type. Consider an audio document as an example of nonstructured data. Its document type belongs to audio document. Its basic property includes document name and intuitive information on document size and creation time. Its semantic feature is the information in the document. The bottom-layer feature is audio frequency and bandwidth. As for structured data, it does not have a basic property, semantic feature, or bottom-layer feature. It is thus directly stored in the original document. Semistructured data like an XML document has some structured data, which is dynamic. Hence, it is difficult to store these data by constructing a mapping table. Fortunately, these data can be extracted to form a string, enabling them to be stored in the database like structured data.

In this manner, structured and nonstructured data can be stored in the database uniformly. For nonstructured data like an image, the content can be analyzed using a description of the image in terms of the basic property, semantic feature, and bottom-layer feature. Structured and semistructured data can be analyzed directly.

4.2. Determination of Logical Predicate and True-Value Range

4.2.1. Determination of Logical Predicate. In order to evaluate data completeness, correctness, and compatibility, let the predicate W denote the high degree, ¬ W low degree, and transition ~W. The correspondence between numerical range and predicates is shown in Figure 2.

4.2.2. Determination of Logical Interval. Weights need to be allocated to the completeness and correctness of data in an application. Data usefulness will not be compromised as long as the major property exists, even if the subordinate property is missing. Based on the proportions of major and subordinate properties, values A and B are computed as follows:

$$B = \sum_{i=1}^{m} w_i$$
$$A = 1 - B \tag{13}$$

where w_i denotes weight and m denotes the largest weight of subordinate properties. Assume that the weights of n properties are sorted in descending order as follows: $w_1 \le w_2 \le \cdots \le w_m \le w_{m+1} \cdots \le w_n$, where $w_1,\ldots,w_m$ denote weights of subordinate properties and $w_{m+1},\ldots,w_n$ denote weights of major properties. The value of m is determined as follows. Sort all weights and compute the sum of weights starting with the smallest weight w_1 until the sum of weights is no larger than the weight w_{m+1}, as shown in

$$\sum_{i=1}^{m} w_i \le w_{m+1}. \tag{14}$$

4.3. Model for Measuring One Dimension of Big Data Validity. The weight of each property in each dimension of the data is first determined to obtain the correspondence between the numerical range of one dimension and the logical predicates: high degree, low degree, and transition, as shown in Figure 2. The distance ratio function $h_T(C)$ with respect to W is selected as the model to measure completeness:

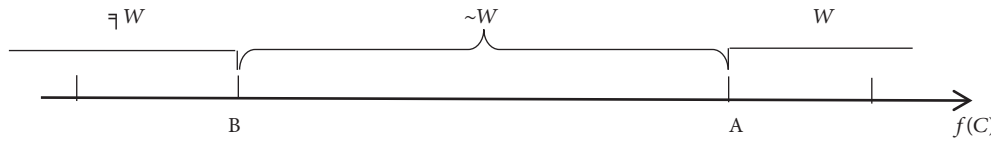


Figure 2: Correspondence between numerical range and predicates.

$$h_T(C) = \begin{cases} 0 & f(C) \le B \\ \dfrac{f(C) - B}{A - B} & B < f(C) < A \\ 1 & f(C) \ge A \end{cases} \quad (15)$$

where f(C) is defined as in (9), (11), and (12). Use the completeness measuring model as an example for the analysis. f(C) in (15) is C1 in (9) and the completeness measuring model is $h_T(C1)$. If the value of data completeness is in the false range (low degree of logic truth¬ W), the value of data completeness is 0 and means that data is missing. If the value of data completeness is in the true range (high degree of logic truth W), the value of data completeness is 1 and means that data is complete. If the value of data completeness is in the transition range (medium degree of logic truth ~W), the value of data completeness is between 0 and 1; closer to 1 means more complete data, and closer to 0 means more missing data.

The model for measuring data correctness or compatibility is similar to the model for completeness. The model measures data correctness $h_T(C2)$ when f(C) in (15) is C2 in (11) and measures data compatibility $h_T(C3)$ when f(C) in (15) is C3 in (12).

4.4. Multidimension Model for Measuring the Integrated Value of Big Data Validity. For a set of K data, completeness and correctness can be measured by the average additive truth scales $h_{kT-M}(C1)$ and $h_{kT-M}(C2)$ which are defined as

$$h_{kT-M}(C1) = \frac{\sum_{i=1}^{K} h_T(C1(i))}{K}$$
$$h_{kT-M}(C2) = \frac{\sum_{i=1}^{K} h_T(C2(i))}{K} \quad (16)$$

where C1(i) and C2(i) denote completeness and correctness for each element in the data set, as defined in (9) and (11).

For a data set in a big data application, the integrated value of data validity can be measured by the weighted sum of metric values for each dimension. Hence, an integrated multidimension model H for measuring data validity in a big data application is

$$H = h_{kT-M}(C1) \times W(C1) + h_{kT-M}(C2) \times W(C2) + h_T(C3) \times W(C3) \quad (17)$$

where $h_{kT-M}(C1)$, $h_{kT-M}(C2)$, and $h_T(C3)$ denote completeness, correctness, and compatibility, respectively, and $W(C1), W(C2), W(C3)$ denote the weights of completeness, correctness, and compatibility, respectively, according to certain application. Thus, we have

$$W(C1) + W(C2) + W(C3) = 1. \quad (18)$$

Compared with the tetrahedron evaluation models, the two models have both similarities and differences. The idea of the multidimension model for measuring data validity in a big data application in this paper (17) is similar to the tetrahedron evaluation models, but the difference between these two models lies in the measuring of each dimension. Our model for measuring one dimension of big data validity is based on medium logic. Logical correctness ensures that the evaluation results are more reasonable and scientific.

5. Conclusions

Medium mathematics systems are introduced for the evaluation of big data validity. A medium logic-based data validity evaluation method is proposed. The contributions of this paper are as follows: (1) Based on the 3V properties of big data, dimensions that have a major influence on data validity are determined. (2) Data completeness, correctness, and compatibility are defined. (3) A medium truth degree-based model is proposed to measure each dimension of data validity. (4) A medium truth degree-based multidimension model is proposed to measure the integrated value of data validity. In the future, other factors that influence big data quality will be studied and corresponding measurement models will be developed.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the State Key Laboratory of Smart Grid Protection and Control of China (2016, no. 10) and the National Natural Science Foundation of China no. 61170322, no. 61373065, and no. 61302157.

References

[1] N. Ramakrishnan and R. Kumar, "Big Data," *The Computer Journal*, vol. 49, no. 4, pp. 20–22, 2016.

[2] V. Marx, "Biology: the big challenges of big data," *Nature*, vol. 498, no. 7453, pp. 255–260, 2013.

[3] A. I. Naimi and D. J. Westreich, "Big Data: A Revolution That Will Transform How We Live, Work, and Think," *American Journal of Epidemiology*, vol. 179, no. 9, pp. 1143-1144, 2014.

[4] W. Pan, Q. Yang, C. Aggarwal, and C. Koch, "Big Data," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 7-8, 2017.

[5] C. L. P. Chen and C. Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data," *Information Sciences*, vol. 275, pp. 314–347, 2014.

[6] H. V. Jagadish, J. Gehrke, A. Labrinidis et al., "Big data and its technical challenges," *Communications of the ACM*, vol. 57, no. 7, pp. 86–94, 2014.

[7] I. Anagnostopoulos, S. Zeadally, and E. Exposito, "Handling big data: research challenges and future directions," *The Journal of Supercomputing*, vol. 72, no. 4, pp. 1494–1516, 2016.

[8] F. Frankel and R. Reid, "Big data: Distilling meaning from data," *Nature*, vol. 455, no. 7209, p. 30, 2008.

[9] S. Staff, "Dealing with data. Challenges and opportunities. Introduction," *Science*, vol. 331, no. 6018, pp. 692-693, 2011.

[10] J. Manyika, M. Chui, and B. Brown, *Big data: The next frontier for innovation, competition, and productivity[J]. Analytics*, Big data, The next frontier for innovation, 2011.

[11] M. S. Viktor, *Big data : a revolution that will transform how we live, work, and think*, John Murray, 2013.

[12] A. I. Naimi and D. J. Westreich, "Big Data: A Revolution That Will Transform How We Live, Work, and Think," *Mathematics & Computer Education*, vol. 17, pp. 181–183, 2013.

[13] D. B. Lindenmayer and G. E. Likens, "Analysis: don't do big-data science backwards," *Nature*, vol. 499, no. 7458, article 284, 2013.

[14] S. Bryson, D. Kenwright, M. Cox, D. Ellsworth, and R. Haimes, "Visually exploring gigabyte data sets in real time," *Communications of the ACM*, vol. 42, no. 8, pp. 82–90, 1999.

[15] N. R. Gough and M. B. Yaffe, "Focus issue: Conquering the data mountain," *Science Signaling*, vol. 4, no. 160, pp. 2-3, 2011.

[16] R. H. Moe, A. Garratt, B. Slatkowsky-Christensen et al., "Concurrent evaluation of data quality, reliability and validity of the Australian/Canadian Osteoarthritis Hand Index and the Functional Index for Hand Osteoarthritis," *Rheumatology*, vol. 49, no. 12, Article ID keq219, pp. 2327–2336, 2010.

[17] F. Bray and D. M. Parkin, "Evaluation of data quality in the cancer registry: Principles and methods. Part I: Comparability, validity and timeliness," *European Journal of Cancer*, vol. 45, no. 5, pp. 747–755, 2009.

[18] W. Meng, *Research and application of data quality evaluation in data warehouse*, Hebei University of Technology, Tianjin, China, 2004.

[19] Q. Yang, P. Zhao, and D. Yang, "Research on Data Qulity Assesment Methodology," *Computer Engineering and Applications*, vol. 40, no. 9, pp. 3-4, 2004.

[20] Jie. Liang, "The Designing Method of Data Validity Restricting Rule Based on GIS," *Computer Engineering and Applications*, vol. 7, pp. 215–217, 2005.

[21] W. J. Zhu and X. A. Xiao, "Propositional calculus system of medium logic," *Nature*, vol. 8, pp. 315-316, 1985.

[22] X. A. Xiao and W. J. Zhu, "A system of medium axiomatic set theory," *Science in China (A)*, vol. 31, no. 11, pp. 1320–1335, 1988.

[23] L. Hong, X.-A. Xiao, and W.-J. Zhu, "Measure of medium truth scale and its application," *Journal of Computer*, vol. 29, no. 12, pp. 2186–2193, 2006.

[24] L. Hong, X.-A. Xiao, and W.-J. Zhu, "Measure of medium truth scale and its application," *Journal of Computer*, vol. 30, no. 9, pp. 1551–1558, 2007.

[25] B. Lang and B. Zhang, "Key Techniques for Building big-data-oriented Unstructured Data Management platform," *Information technology and Standardization*, vol. 10, pp. 53–57, 2013.