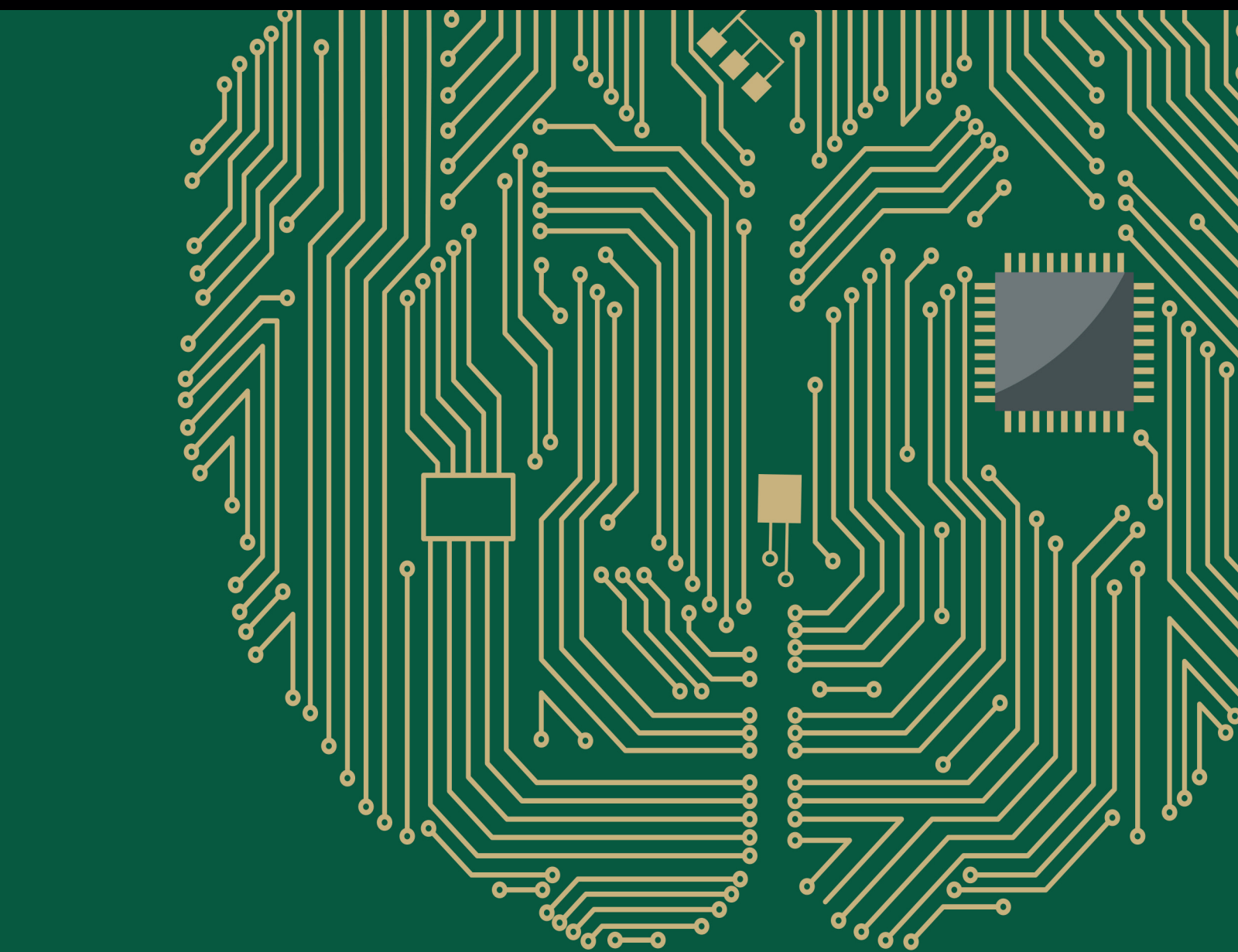
Computational Intelligence and Neuroscience

Applications of Computational Intelligence in Time Series

Lead Guest Editor: Francisco Martínez-Álvarez
Guest Editors: Alicia Troncoso, Jorge Reyes, María Martínez-Ballesteros, and José C. Riquelme



Applications of Computational Intelligence in Time Series

Computational Intelligence and Neuroscience

Applications of Computational Intelligence in Time Series

Lead Guest Editor: Francisco Martínez-Álvarez
Guest Editors: Alicia Troncoso, Jorge Reyes,
María Martínez-Ballesteros, and José C. Riquelme

Editorial Board

Contents

Applications of Computational Intelligence in Time Series
Francisco Martínez-Álvarez, Alicia Troncoso, Jorge Reyes, María Martínez-Ballesteros, and José C. Riquelme
Volume 2017, Article ID 9361749, 2 pages

An Evolutionary Method for Financial Forecasting in Microscopic High-Speed Trading Environment
Chien-Feng Huang and Hsu-Chih Li
Volume 2017, Article ID 9580815, 18 pages

Statistical Modeling and Prediction for Tourism Economy Using Dendritic Neural Network
Ying Yu, Yirui Wang, Shangce Gao, and Zheng Tang
Volume 2017, Article ID 7436948, 9 pages

Main Trend Extraction Based on Irregular Sampling Estimation and Its Application in Storage Volume of Internet Data Center
Beibei Miao, Chao Dou, and Xuebo Jin
Volume 2016, Article ID 9328062, 12 pages

Artificial Neural Network and Genetic Algorithm Hybrid Intelligence for Predicting Thai Stock Price Index Trend
Montri Inthachot, Veera Boonjing, and Sarun Intakosum
Volume 2016, Article ID 3045254, 8 pages

A Forecasting Model for Feed Grain Demand Based on Combined Dynamic Model
Tiejun Yang, Na Yang, and Chunhua Zhu
Volume 2016, Article ID 5329870, 6 pages

A Long-Term Prediction Model of Beijing Haze Episodes Using Time Series Analysis
Xiaoping Yang, Zhongxia Zhang, Zhongqiu Zhang, Liren Sun, Cui Xu, and Li Yu
Volume 2016, Article ID 6459873, 7 pages

Hindawi
Computational Intelligence and Neuroscience
Volume 2017, Article ID 9361749, 2 pages
https://doi.org/10.1155/2017/9361749

Editorial

Applications of Computational Intelligence in Time Series

Francisco Martínez-Álvarez,[1] Alicia Troncoso,[1] Jorge Reyes,[2] María Martínez-Ballesteros,[3] and José C. Riquelme[3]

[1]*Division of Computer Science, Pablo de Olavide University, 41013 Seville, Spain*
[2]*NT2 Labs, Santiago, Chile*
[3]*Department of Computer Science, University of Seville, Sevilla, Spain*

Correspondence should be addressed to Francisco Martínez-Álvarez; fmaralv@upo.es

Received 26 April 2017; Accepted 26 April 2017; Published 22 May 2017

The prediction of the future has fascinated the human being since its early existence. Actually, many of these efforts can be noticed in everyday events such as energy management, telecommunications, pollution, bioinformatics, and seismology and, obviously, in neuroscience. Accurate predictions are essential in economic activities as remarkable forecasting errors in certain areas may involve large loss of money.

In this context, the successful analysis of temporal data has been a challenging task for many researchers during the last decades and, indeed, it is difficult to figure out any scientific branch with no time-dependent variables.

Computational intelligence is known for including powerful techniques such as artificial neural networks, fuzzy systems, evolutionary computation, learning theory, or probabilistic methods. Thus, this special issue has been focused on the application of such techniques to time series. In particular, the goal was sharing recent advances in time series analysis and providing an interesting opportunity to present and discuss the latest practical advances in real-world applications.

Rigorous and extensive review processes have been carried out. Papers selected for this special issue present new findings and insights in the field of time series forecasting. A broad range of topics has been discussed, especially in the following areas: finances, tourism, feed grain demand, haze episodes, stock price, or data storage.

Genetic algorithms have been developed with binary coding to analyze high-speed trading research using price data of stocks on the microscopic level. This problem is certainly new and unexplored from computational intelligence techniques. Reported results show that the system is able to improve the accuracy for price movement forecasting, thus encouraging conducting research in this direction.

Tourism demand forecasting has also been addressed by means of seasonal trend autoregressive integrated moving averages with dendritic neural networks. Data from Japan are used to test the predictive performance of the proposed model. The model generates short-term predictions, after applying SARIMA models to exclude the long-term linear trend. This study mixes linear and nonlinear models and suggests that further analysis in the combination of such techniques is desirable.

Likewise, a new irregular sampling estimation method to extract the main trend of the time series has been proposed. To achieve this, first, the Kalman filter is used to remove dirty data. Second, the cubic spline interpolation and average method are used to reconstruct the main trend. The proposed approach has been applied to storage volume series of Internet data center. Results are quite promising.

A novel hybrid artificial intelligent system has been proposed to forecast stock price index trend. In particular, artificial neural networks combined with genetic algorithms have been applied to data from Thailand's SET50 index trend, from years 2009 to 2014. Multiple features and different time spans have been considered. Comparisons to other methods confirm the success of this novel methodology.

The long-term prediction of feed grain demand issue has been extensively discussed. A multivariate regression

model along with a dynamic forecasting model has been introduced. Firstly, the correlation between the demand and its influence factors are studied and, secondly, changes in trend in factors affecting the demand are forecasted. Reported results corroborate the effectiveness of the methodology.

A novel long-term prediction model for Beijing haze episodes has been introduced. The authors have built a dynamic structural measurement model of daily haze increment and have reduced the model to a vector autoregressive model. Such model performs satisfactorily on next day's air quality index forecasting, reaching in many cases an accurate rate close to 90%. Therefore, sudden haze burst could be predicted with this method.

Acknowledgments

Finally, we would like to thank all the authors for their excellent work and contributions to this special issue. We would also like to express our gratitude to all the reviewers for their thorough revisions and patience in assisting us.

Francisco Martínez-Álvarez
Alicia Troncoso
Jorge Reyes
María Martínez-Ballesteros
José C. Riquelme

Hindawi
Computational Intelligence and Neuroscience
Volume 2017, Article ID 9580815, 18 pages
https://doi.org/10.1155/2017/9580815

Research Article

An Evolutionary Method for Financial Forecasting in Microscopic High-Speed Trading Environment

Chien-Feng Huang and Hsu-Chih Li

Dept. of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan

Correspondence should be addressed to Chien-Feng Huang; cfhuang15@nuk.edu.tw

Received 27 August 2016; Accepted 11 January 2017; Published 20 February 2017

Academic Editor: Jorge Reyes

The advancement of information technology in financial applications nowadays have led to fast market-driven events that prompt flash decision-making and actions issued by computer algorithms. As a result, today's markets experience intense activity in the highly dynamic environment where trading systems respond to others at a much faster pace than before. This new breed of technology involves the implementation of high-speed trading strategies which generate significant portion of activity in the financial markets and present researchers with a wealth of information not available in traditional low-speed trading environments. In this study, we aim at developing feasible computational intelligence methodologies, particularly genetic algorithms (GA), to shed light on high-speed trading research using price data of stocks on the microscopic level. Our empirical results show that the proposed GA-based system is able to improve the accuracy of the prediction significantly for price movement, and we expect this GA-based methodology to advance the current state of research for high-speed trading and other relevant financial applications.

1. Introduction

The advances of information technology and big data research in finance have led to an ever increasing pace to market-driven events and information that prompt decision-making and actions by computerized high-speed trading strategies. Speed has become more important to traders in financial markets because faster trading may bring about more profit opportunities, which appears to drive an arms race among traders to utilize high-speed trading technology for an edge over others. The net result is that today's markets experience intense activity in the highly dynamic environment of seconds or only a little fraction of seconds, where trading algorithms respond to others at a pace much faster than it would take for human traders to blink. This new breed of trading technology and platform involves the implementation of low-latency, high-speed trading strategies and has now resulted in remarkable portion of activities in the financial markets [1].

The time series data in high-speed trading ranges from the granular data of stock transactions at regular intervals of several seconds to the price data irregularly spaced with quotes arriving randomly at intervals of a fraction of seconds, which is mostly referred to as high-frequency trading (HFT). Over the past years, high-speed/frequency trading has been playing an important role in global financial markets due to the massive increases recently in trading volumes by such strategies. HFT has accounted for about 40% of all equity trades in the European market in 2009 [2]. As of 2013, it is reported that HFT activities accounted for 49% trading volume in the American equity markets [3]. More recently, the technologies of HFT have also been diffusing into the financial markets worldwide, including Asia [3].

In high-speed trading research, the process of price formation in market microstructure generally produces large amount of data in relatively short periods of time. The sheer volume of trading data generated in such environments provides plenty of resources for modeling and decision-making in big data research for financial applications. Recent microstructure research and advances in econometric modeling have facilitated an understanding of the characteristics of high-speed data [4]. In Taiwan, the Taiwan Stock Exchange (TWSE) is the major platform for stock trading in which

transactions of various stocks typically occur at regular intervals of five seconds. In this study, we thus aim to develop novel methodology to shed light on the research in the context of high-speed trading.

In the past decades, there have been a number of computational intelligence (CI) approaches studied for financial applications due to its significant impact on the human society, ranging from fuzzy systems, artificial neural networks (ANNs), support vector machines (SVMs), and evolutionary algorithms (EAs) [5] to hybrid and ensemble models, along with other approaches [6]. These studies encompass a wide range of applications, including abnormal noise and fraud detection [7, 8], arbitrage [9, 10], bankruptcy detection [11, 12], financial forecasting [13–15], and portfolio optimization [16–18].

Although there exist many aforementioned CI methods developed for solving various financial problems, a recent survey by Aguilar-Rivera et al. [19] indicated that genetic algorithms (GA, a branch of evolutionary algorithms) have remained one of the most popular approaches in the CI literature for financial research and applications. Among several major financial areas for CI studies, forecasting is a subject that has been extensively investigated. Typically, it consists of the estimation of future values or trends of investment vehicles for relevant decision-making and investment action. Although perfect prediction is not possible, several GA-based methods have been developed to improve the accuracy of prediction. In these works, the GA techniques have been employed mainly for the optimization task in the proposed models. For instance, Kim and Han [20] proposed a GA approach to the discretization of continuous variables and the determination of optimal range for the connection weights of the ANNs to predict the stock price index. They suggested that their approach was able to reduce the numbers of attributes and the performance of forecasting was improved. In [14], Araújo and Ferreira proposed the GAs to search for optimal linear filters in forecasting applications. By comparing against several other models, they showed that their proposed GA-based models outperformed the benchmarks.

In addition to the GA-based methods, the class of Genetic Programming (GP) has been used for similar forecasting tasks, as well. For example, Shao et al. [21] proposed an improved GP-based financial forecasting tool for EDDIE (Evolutionary Dynamic Data Investment Evaluator [22]), which provided a new version of grammar to increase the search space for decision trees. In this work, a guided local search along with hill climbing was also employed to assist EDDIE with the optimization task for the rules and the forecasting time horizons. The proposed method was then tested against several financial time series and it was reported that the method was able to improve the previous version of the EDDIE for financial forecasting.

Recently, the methods of estimation of distribution algorithms (EDAs) have been studied in the area of evolutionary computation for several research problems. For financial prediction, for instance, Peralta Donate and Cortez [23] developed a NN-based forecasting approach in which a univariate marginal distribution algorithm (UMDA) was proposed for the optimization task. In this method, the best half of the current population is selected to form a portion of the new population whereas the remaining individuals are generated by the probability distribution computed by the method. Using the time series data of the Dow Jones Industrial Average, the authors compared their method with the ARIMA models, random forest, echo state networks, and SVMs and showed that their method was able to attain lower mean-squared error than others.

In addition to the EA studies discussed thus far, various types of GA-based methodologies have been developed for financial research and applications, and an extended survey is provided in [19]. However, since the advanced IT technology for fast trading platforms has been made available to public just recently, high-speed trading is still a relatively new subject to CI researchers. In particular, to the best of our knowledge, the existing major CI research has provided forecasting techniques based on the information extracted from regular, macroscopic prices, for example, daily price of a stock. In contrast, in the context of high-speed trading, the microscopic price structures are more important because the formation of the actual transaction price typically resulted from different auction prices on the microscopic level. Therefore, these microstructures shall provide more information than macrostructures for price forecasting. By this rationale, in this study, we thus aim to develop a CI-based methodology to tackle the forecasting task for high-speed trading. Since the review by Aguilar-Rivera et al. [19] indicates that the class of GA is one of the most popular approaches in CI literature for financial applications, our goal is thus to bring about a novel GA methodology to shed light on this relatively unexplored area for CI research. As our experimental results show later, using the microscopic price data from the call auction market, the methodology we proposed is indeed more effective than conventional approaches for forecasting in the context of high-speed trading.

To sum up, the overall proposed methodology in this study is to offer feasible models for the real-world high-speed trading applications. Our objective is to advance the current state of the research for the class of CI-based search algorithms particularly tailored for forecasting in the high-speed trading environment, in order to further our understanding of the complex characteristics in stock market and the applicability of the CI-based algorithms to such problems.

2. Materials and Methods

Currently, in the call auction market of Taiwan, the transaction prices of a stock, both the best five bid and ask prices, and their sizes are available to all market participants. In this work, we propose to use these publically available microscopic data to construct intelligent models for price forecasting. Before delving into the details of the methods studied, we provide the financial background for the call auction market first.

2.1. Trading Mechanism of the Call Auction Stock Market in Taiwan. In the Taiwan Stock Exchange (TWSE), the execution prices of stock trading during regular trading sessions are

Table 1: Bid and ask quotes before matching (price in NT$ and volume in Lots).

Cumulative bid volume	Bid volume	Price	Ask volume	Cumulative ask volume
162	162	107.00	94	311
162		106.50	25	217
185	23	106.00	20	192
195	10	105.50	15	172
195		105.00	46	157
252	57	104.50	55	111
252		104.00	20	56
282	30	103.50	13	36
282		103.00	3	23
381	99	102.50		20
381		102.00		20
403	22	101.50		20
408	5	~~~		20
441	33	93.00	20	20

Table 2: Remaining unexecuted orders after matching.

Cumulative bid volume	Bid volume	Price	Ask volume	Cumulative ask volume
		107.00	94	119
		106.50	25	32
		106.00	7	7
10	10	105.50		
10		105.00		
67	57	104.50		
67		104.00		
97	30	103.50		
97		103.00		
196	99	102.50		
196		102.00		
218	22	101.50		
223	5	~~~		
256	33	93.00		

determined by the periodic call auction principles (http://www.twse.com.tw/en/products/trading_rules/mechanism01.php#2). Orders are collected over a specified period of time (the current period is five seconds per auction), which will be matched at the end of that period using the following rules:

(1) Orders are first matched according to their price priority.

(2) If the orders are of the same price, they will be matched according to their time priority.

(3) For each call auction, an execution price is selected for the greatest number of orders to be executed.

In this auction system, right after the end of each matching period, a set of information is disclosed to the public, including the execution price and volume and the prices and volumes of both the five highest unexecuted bid quotes and the five lowest unexecuted ask quotes. As a result, the five best bid/ask prices and volumes observed by the public are for unexecuted orders in the prior call auction. The unexecuted orders, together with the new, subsequent orders from the investors will then enter into the system to participate in the next call auction. For illustration, Table 1 shows an example of bid and ask quotes prior to matching.

For this example, using the call auction rule for the price that enables the largest volume of orders to be executed, the system then determines the execution price to be 106.00, and 185 lots in total are executed at that price. The results for the remaining unexecuted bid and ask quotes are shown in Table 2.

Table 3 shows the disclosure of the five best unexecuted bid and ask quote prices/volumes after matching. Together with the disclosed execution price and volume, the unexecuted bid prices and volumes that are disclosed include NT$105.50, 10 lots; NT$104.50, 57 lots; NT$103.50, 30 lots; NT$102.50, 99 lots; NT$101.50, 22 lots. And the unexecuted ask prices and volumes that are disclosed are NT$107.00, 94 lots; NT$106.50, 25 lots; NT$106.00, 7 lots.

Table 3: Disclosure of information after matching.

Five best unexecuted quotes			
Bid price	Bid volume	Ask price	Ask volume
		107.00	94
		106.50	25
		106.00	7
105.50	10		
104.50	57		
103.50	30		
102.50	99		
101.50	22		

2.2. Stock Price Forecasting in the Call Auction Market. In the call auction market, the disclosed bid quotes contain the five highest prices and corresponding volumes for sale of the stock, and the disclosed ask quotes contain the five lowest prices and corresponding volumes for buying it. Intuitively, these bid and ask orders indicate the extent of demand and supply for a stock, respectively, which may be used to forecast the price movement in the future because the price tends to go up (or down) if the demand is more (or less) than the supply. In this paper, using the disclosed information, we thus intend to propose an intelligent GA-based system for the forecasting task of stock price.

In this section, we provide descriptions for several methodologies employed for this study: a rule-based forecasting method, two regression-based methods, and our proposed GA-based methods.

2.2.1. Rule-Based Forecasting Models. As discussed in the previous section, at the end of each call auction, the disclosed information includes the execution price and volume and the five best unexecuted prices and volumes of bid and ask orders. According to the matching mechanism for the bid and ask orders, the new execution price at the next call shall be determined by the unexecuted orders at the current call and the other continuous influx of new orders entering into the system before the next call. Since the new coming orders are not disclosed to the public, market participants can only utilize the execution price and volume and the five best unexecuted bid and ask quotes at the current call for price prediction in the future. In order to predict the price at the next call, Hu and Chan [24] proposed the following rule to infer the imbalance in the order book.

Rule 1. "If a call reports both bid and ask and the transaction price is equal to the bid (ask), then the transaction price of the next call tends to go up (down)."

The rationale for this rule is that when the execution (transaction) price is equal to the bid, there are certain buying orders left unfulfilled; so these remaining demand orders may push up the price in the future. Conversely, if the execution (transaction) price is equal to the ask, there are some selling orders left unfulfilled and these supply orders then tend to push down the price. Therefore, Rule 1 may be used to predict the movement direction of the transaction price at the next call.

In addition to this method, we propose to study other versions of more sophisticated models for the call auction market. We describe the regression-based methods in the next subsection.

2.2.2. Regression-Based Forecasting Models

(A) Linear Regression Models. Linear regression models have been studied in several financial applications, including the task of stock selection [25, 26] and the studies for the impact of order imbalance in the call auction market [24, 27]. Since the major component of this study is concerning the forecasting task under the call auction mechanism, which is similar to the ones in [24, 27] where the authors mainly used linear regression methods for their studies, in this work, we thus also propose to employ the linear regression models as follows.

Consider a given set S with n training instances $\{(x_1(t), y_1(t)), (x_2(t), y_2(t)), \ldots, (x_n(t), y_n(t))\}$ at time t. Each training instance $x_i(t)$ serves as the input to generate a corresponding output $y_i(t)$, for $i = 1, \ldots, n$. Using β and $\varepsilon(t)$ to denote the regression coefficients and the error terms, respectively, a linear regression model often takes the form

$$Y(t) = X(t)\beta + \varepsilon(t), \tag{1}$$

where if p is the input dimension,

$$Y(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \end{pmatrix},$$

$$X(t) = \begin{pmatrix} x_{11}(t) & \cdots & x_{1p}(t) \\ x_{21}(t) & \cdots & x_{2p}(t) \\ \vdots & \ddots & \vdots \\ x_{n1}(t) & \cdots & x_{np}(t) \end{pmatrix}, \tag{2}$$

$$\beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix},$$

$$\varepsilon(t) = \begin{pmatrix} \varepsilon_1(t) \\ \varepsilon_2(t) \\ \vdots \\ \varepsilon_n(t) \end{pmatrix}.$$

In contrast to the simple rule-based forecasting method in the previous subsection, a more sophisticated model can be constructed through the linear regression model above. In a nutshell, the transaction price of the next call may hinge on the disclosed, best five ask (from the selling side) and best

five bid (from the buying side) quotes which represent the supply and demand pressure of the stock, respectively. In this call auction market, here we designate each input variable to be the product of the bid (or ask) price and corresponding volume in order to model the degree of the buying (or selling) strength at each particular price level. Thus, these ten variables may be used as the inputs to the regression model and the price of the stock at the next call may serve as the output of the model. Therefore, once the regression model is constructed, it can be used to predict the future price of the stock as long as the values of the input variables are provided.

(B) Logistic Regression Models. The linear regression method may be used to model continuous variables such as the future stock price discussed above. However, other versions of regression models may be useful. For instance, if the goal is to predict future price to go up or down, then a binary output of the model may be more appropriate. In this case, a binary logistic model is studied here as an alternative to linear regression, which is used to estimate the probability of the binary response variable (price going up or down) based on the same set of input variables in the last subsection. The logistic regression model we use in this study works as follows.

Here, we consider again, in the previous subsection, the given set S with n training instances $\{(x_1(t), y_1(t)), (x_2(t), y_2(t)), \ldots, (x_n(t), y_n(t))\}$ at time t. The logistic regression employs a standard logistic function $f(x)$, which can be defined as

$$P(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}}. \tag{3}$$

Note that $P(X)$ is interpreted as the probability of the dependent variable equal to one of the binary outcomes. In this model, the 10 variables of best five ask and five bid quotes and the binary response variable y (price up or down at the next call) are used to compute the logistic regression model. Once the regression model is constructed, it can be used to predict the direction of price change in the future as long as the values of the input variables are provided. We will describe how to use the linear and logistic regression models for forecasting in Section 2.2.4.

2.2.3. GA-Based Forecasting Models. Genetic algorithms [28] have been used as computational models of natural evolutionary systems and as a class of adaptive algorithms for solving optimization problems. GA operate on an evolving population of artificial agents. Each agent is comprised of a genotype (often a binary string) encoding a solution to some problem and a phenotype (the solution itself). GA regularly start with a population of randomly generated agents within which solution candidates are embedded. In each iteration, a new generation is created by applying crossover and mutation to promising candidates selected according to probabilities biased in favor of the relatively fit agents. As a result, evolution occurs by iterated stochastic variation of genotypes and selection of the best phenotypes in an environment according to how well the respective solution solves a problem. Successive generations are created in the same manner until a well-defined termination criterion is met. The core of this class of algorithms lies in the production of new genetic structures along the course of evolution, thereby providing innovations to solutions for the problem at hand. The steps of a simple GA are shown in the following.

Step 1. Randomly generate an initial population of l agents, each being an n-bit genotype (chromosome).

Step 2. Evaluate each agent's fitness.

Step 3. Repeat until l offspring have been created.

(a) Select a pair of parents for mating.

(b) Apply variation operators (crossover and mutation).

Step 4. Replace the current population with the new population.

Step 5. Go to Step 2 until terminating condition.

The GA-based methods have been widely employed to solve optimization problems and applications in computational finance and investment [19]. In this survey, the GA-based methods have been shown to be very useful in stock selection and portfolio optimization, as well as various types of financial prediction. Motivated by these research results, we intend to employ the GA to develop intelligent systems for price forecasting in this study through the optimization of parameters of the forecasting models that account for the demand and supply of a stock. More specifically, we propose to use the ten disclosed quotes (the best five bid and five ask prices and volumes) of a stock as the inputs to the GA-based model in order to predict the direction of future price movement. Here, we define the extent of the bid strength (BS) and ask strength (AS) of a stock at time t as follows:

$$\begin{aligned} \mathrm{BS}(t) &= \sum_{i}^{5} u_i b_i(t); \\ \mathrm{AS}(t) &= \sum_{i}^{5} v_i s_i(t), \end{aligned} \tag{4}$$

where $b_i(t)$ and $s_i(t)$ denote the product of price and volume (lots) for the ith bid and ask quote of the stock at time t, respectively, and u_i and v_i denote the corresponding weight for the ith bid and ask quote, respectively.

The difference between a stock's demand (buying) and supply (selling) power can then be defined as

$$x(t) = \mathrm{BS}(t) - \mathrm{AS}(t). \tag{5}$$

In this study, a prediction rule for the price movement according to (5) can be proposed as follows: When the difference $x(t)$ is positive (i.e., the demand is more than the supply), one predicts the price of the stock to go up in the future. Conversely, it is predicted to go down if $x(t)$ is negative.

As an illustration, in Table 3, the disclosed unexecuted bids include 10 lots of \$105.50, 57 lots of \$104.50, 30 lots of NT\$103.50, 99 lots of \$102.50, and 22 lots of \$101.50. These

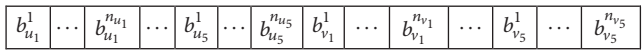


Figure 1: Chromosome encoding.

unexecuted bids represent the remaining demand force for the stock where the higher the bid order is, intuitively, the more likely it is to be matched with new ask orders in the future. Conversely, for the unexecuted ask orders, they remain as the supply force of the stock; and the lower an ask order, the more likely it is to be matched with new bid orders. Therefore, various levels of bid and ask quotes may have different degree (weight) of impact on determining the final transaction price at each call. In this study, we thus propose the model by (4) and (5) to calculate the force that potentially leads to the price change due to various levels of demand and supply. The result can be then used as the prediction for price movement of a stock (i.e., the prediction rule by (5) mentioned above); and the corresponding forecasting performance can be further evaluated by the precision measure in (7) discussed later in the next subsection (Section 2.2.4).

In order to determine the weight for each price level of bids and asks, we hereby propose the GA to search for optimal u_i and v_i's as follows.

For the model studied here, we devise a chromosome as two portions that encode the parameters u_i and v_i, $i = 1, \ldots, 5$, for the weight corresponding to the ith bid and ask quotes, respectively. A binary coding scheme is used in this study to represent a chromosome in the GA. For instance, in Figure 1, loci $b_{u_i}^1$ through $b_{u_i}^{n_{u_i}}$, $i = 1, \ldots, 5$, represent the encoding configuration for u_i; and loci $b_{v_i}^1$ through $b_{v_i}^{n_{v_i}}$, $i = 1, \ldots, 5$, represent the encoding for v_i, where n represents the bit length.

In our encoding scheme here, a chromosome representing the genotypes of parameters is to be transformed into the phenotype by the following equation for further fitness computation:

$$y = \min_y + \frac{d}{2^l - 1} \times \left(\max_y - \min_y\right), \tag{6}$$

where y is the corresponding phenotype for the particular parameter; $\min_y$ and $\max_y$ are the minimum and maximum values of the parameter; d is the corresponding decimal value (d being truncated to integers if the parameter is of integer type), and l is the length of the block used to encode the parameter in the chromosome.

Once the GA is employed to search for optimal u_i's and v_i's for any prespecified objective, these GA-based models can be used for the prediction of a stock price in the future. For instance, as the new best five bid and ask quotes of a stock at the current call are made available to the public, the proposed GA-based model can use this information to calculate the difference between the demand and supply and further predict whether the price would go up or down in the future.

2.2.4. Performance Measures and Forecasting Systems for Comparison. In this work, the performance of the forecasting system can be measured by the precision defined as

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{7}$$

where TP and FP denote the number of true positives and false positives, respectively. In this study, we propose to calculate true and false positives as follows:

> "When a system predicts the price of a stock to go up at some point in the future, and if the price indeed goes up then, a true positive occurs; otherwise, it is a false positive."

Alternatively, the performance of a system may also be evaluated by the accuracy metric defined as follows:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{8}$$

where TN and FN denote the number of true negatives and false negatives, respectively, which can be defined as follows:

> "When the stock price is predicted to go down at some point in the future, and if the price indeed goes down then, a true negative occurs; otherwise, it is a false negative."

In this study, we are more interested in the task of predicting the price to go up, so that one may make profit if a proper buying strategy is employed using this prediction result. Therefore, the major experimental results we present later in the Results (Section 3) center around the precision performance evaluated by (7). However, in the final portion of the Results, we also present some results using the accuracy metric by (8) for additional comparison of different systems, which is used to show that our proposed system is effective under both the precision and accuracy measures.

In addition, it is worthwhile to provide further discussions concerning some pitfalls in the high-speed trading context of the call auction market. In the stock market, there are three cases for the price movement of a stock at any moment in the future.

Case 1. The new price remains the same as the current one.

Case 2. The price goes up (i.e., the new price is higher than the current one).

Case 3. The price goes down (i.e., the new price is lower than the current one).

We notice that, in current Taiwan's call auction market, a stock's transaction price tends to remain unchanged among several consecutive calls. Thus, the distribution of the transaction price of a stock is typically skewed toward Case 1. As an illustration, we consider the stock of the Taiwan Semiconductor Manufacturing Company Limited (TSMC) from Sept./22/2015 through Nov./24/2015. If the call auction system reports that a transaction price is equal to the bid, then according

to Section 2.2.1's Rule 1, the next transaction price tends to go up. We then inspect the real data under this condition and observe that the corresponding transaction prices at next calls actually consist of 41585, 14982, and 41 occurrences of price going flat, up, and down between two consecutive calls, respectively. As a result, if one designates the true positive (TP) by Section 2.2.1's forecasting rule to include both the flat case (Case 1) and up case (Case 2), then the number of TPs is raised up to 56567 and the precision is 0.99, so that the forecasting task (i.e., forecasting the price to remain the same or go up) becomes trivial. However, if the flat cases are excluded when computing TP, the resultant precision drops significantly to 0.26 and the forecasting (i.e., forecasting the price to only go up) would be difficult. Therefore, once the event of price remaining unchanged between two consecutive calls is excluded, the rate of correct prediction by Section 2.2.1's rule may drop significantly. As the experimental results show later, this challenge will be substantial to several forecasting methods in the call auction market studied here. Motivated by this, we hereby intend to propose new systems in order to improve the accuracy of prediction for the movement of stock price in the high-speed call auction market.

In this study, we will conduct a comparative study on the prediction performance for the aforementioned forecasting systems as follows.

(i) System 1 (S1). The forecasting rule in Section 2.2.1 is used to predict the price of a stock at the next call to go up (down) if one observes that both bid and ask are reported and the transaction price is equal to the bid (ask) at the current call.

(ii) System 2a (S2a). The linear regression model in Section 2.2.2(A) is employed for this system. Using both the disclosed five best bid and ask quotes at the current call as the input to the regression model, the output of the model then serves as the prediction of the price at the next call.

(iii) System 2b (S2b). The logistic regression model in Section 2.2.2(B) is employed for this system. Again using the 10 variables of both the five bid and five ask quotes at the current call, the logistic regression model calculates the probability of the price going up or down at the next call. Here, we predict the price to go up if the probability is greater than 0.5; otherwise, the price is predicted to go down.

(iv) System 3 (S3). The GA-based model in Section 2.2.3 is employed for this system. The fitness function of a chromosome is defined as the precision by (7) or the accuracy by (8). Once the GA is employed to search for the optimal parameters for the models, the resultant models may be used for the prediction of price movement of a stock in the future. Therefore, both the disclosed five best bid and ask quotes at the current call are used as the input to the model. If the model detects that the demand of a stock is larger (less) than the supply, it predicts the price to go up (down) at the next call.

(v) System 4 (S4). This system combines S1 and S3 to predict the price in the future. The fitness function of a chromosome is defined as the precision by (7) or the accuracy by (8). Therefore, the disclosed five best bid and ask quotes at the current call and the transaction price are used as the input to the model. Intuitively, we expect the combination of the two systems to improve either S1 or S3 alone and thus refine the forecasting system. In addition, once the system detects that the demand of a stock is larger than the supply, we expect the stock's price to go up at some point in the future, rather than the next call. In this system, we thus propose to enrich the encoding of the chromosome by including a few bits to search for the optimal period for calculating the precision.

3. Results

In this section, we present experimental results for the five systems described in Section 2.2.4. We use 10 stocks with large market capitalization in the Taiwan Stock Exchange for illustration, among which 5 are from the semiconductor and electronics industry and the other 5 are from the financial industry. These two industries are the two major ones in Taiwan and thus consist of a large portion of commercial activities in Taiwan. Therefore, we choose these 10 stocks in order to provide more representative characteristics of Taiwan's stock market to examine our proposed methods. The datasets are made available to the public by the TWSE where the transaction prices, five best bid and ask quotes, and trading volumes are used to examine the performance of the systems. Table 4 shows the 10 stocks used in this study. In this study, the datasets were extracted from two periods of time (each period accounts for the total of 30 trading days): (1) Sept./15/2015 through Nov./03/2015, during which time the value of the Taiwan Stock Exchange Capitalization Weighted Stock Index (TSEC weighted index) went up from 8259.99 to 8713.19, and that is a period of time the broad stock market is achieving positive gain; (2) Dec./10/2015 through Jan./21/2016, during which time the TSEC weighted index went down from 8216.17 to 7664.01, thus the broad stock market achieving negative gain. The reason we selected these two periods of time is to examine whether our proposed method would be generally effective when the broad stock market makes either gains (i.e., the market encounters favorable conditions) or losses (i.e., the market faces adverse challenges). For each trading day, the market opens from 9:00 am through 1:30 pm. The transaction data is sampled per 5 seconds. Each sample contains the information of ticker, transaction price, number of transactions, volume, the best five ask and bid quotes, etc.

In order to examine the effectiveness of the systems studied, statistical validation is presented in this section. As shown in Figure 2, we use the data of the first several days to train the model, and the remaining data is used for testing. This setup is to provide a set of temporal validations to examine the effectiveness of the models for the dynamic characteristics in many financial applications, which is different from the regular cross-validation procedure where the process of data being split into two independent sets is randomly repeated several times without taking into account the data's temporal order [5, 10].

In the training phase of each TV, we conduct 50 runs for the GA experiments with population size of 50 individuals in each generation. For each weighting parameter, we use

Table 4: Datasets of the 10 companies used in this study.

Ticker	Name (English)	Name (Chinese)	Market capitalization as of June/21/2016 ($NT billions)	Industry
2311	Advanced Semiconductor Engineering Inc.	日月光半導體製造股份有限公司	277.40	Semiconductor and electronics
2325	Siliconware Precision Industries Co., Ltd.	矽品精密工業股份有限公司	160.80	Semiconductor and electronics
2330	Taiwan Semiconductor Manufacturing Company Limited	台灣積體電路製造股份有限公司	4278.52	Semiconductor and electronics
2409	AU Optronics Corp.	友達光電股份有限公司	90.66	Semiconductor and electronics
2454	MediaTek Inc.	聯發科技股份有限公司	361.24	Semiconductor and electronics
2882	Cathay Financial Holding Co., Ltd.	國泰金融控股股份有限公司	474.89	Finance
2883	China Development Financial Holding Corporation	中華開發金融控股股份有限公司	117.61	Finance
2885	Yuanta Financial Holding Co., Ltd.	元大金融控股股份有限公司	123.92	Finance
2886	Mega Financial Holding Company	兆豐金融控股股份有限公司	333.20	Finance
2891	CTBC Financial Holding Co., Ltd.	中國信託商業銀行股份有限公司	298.81	Finance

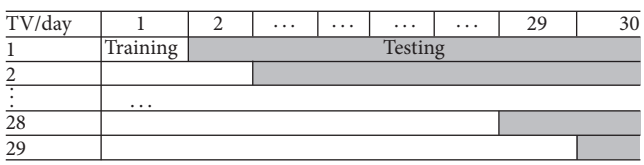


Figure 2: Temporal validation.

8 bits to encode it whose range is designated from 0 to 1 (using more bits are possible to offer higher resolution, but the computational overhead is increased, as well). We also use a binary tournament selection [29], one-point crossover, and mutation rates of 0.7 and 0.005, respectively (the values of crossover and mutation rates have various effects on the performance of the GA search; the two values we chose here are typical as suggested in [30, 31]). In order to track the change of the quality of solutions searched by the GA over time, a traditional performance metric for search algorithms is the "best-so-far" curve that plots the fitness of the best individual that has been seen so far by generation n for the GA—i.e., a point in the search space that optimizes the objective function thus far [5]. As an illustration, Figure 3 displays a typical averaged best-so-far curve over 50 runs attained by the GA in this study (these results were obtained for MediaTek Inc.; since the results for other stocks are similar, they are not displayed here). The averaged best-so-far performance curve is calculated by averaging the best-so-far solutions obtained at each generation for all 50 runs, where the vertical bars (error bars) overlaying the curve represent the 95% confidence intervals about the means. From this

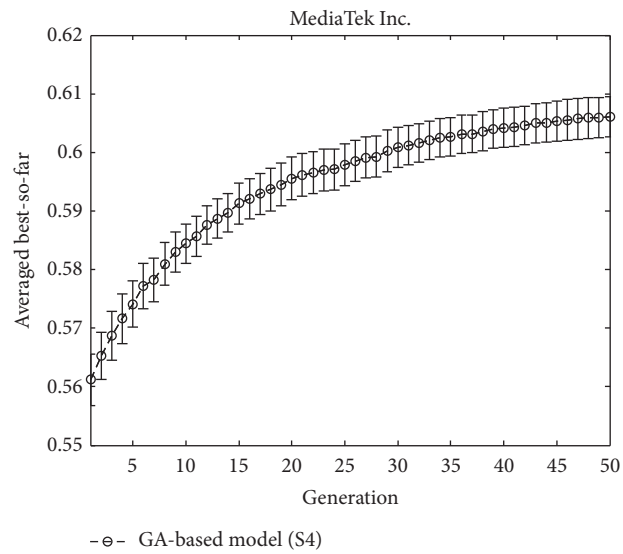


Figure 3: Best-so-far curve by the GA for MediaTek Inc.

figure, one can clearly see the convergence of the GA search and the length of the error bars are sufficiently small, thereby indicating the GA search results are robust.

Furthermore, in this study, the best model learned in the training phase for each run is examined in the testing phase. Therefore, in both training and testing phases of each TV, the averaged fitness (precision) of the models can be calculated. For illustration, Table 5 displays the averaged precisions for Advanced Semiconductor Engineering Inc. during the period

Table 5: Averaged precision in training and testing in each TV.

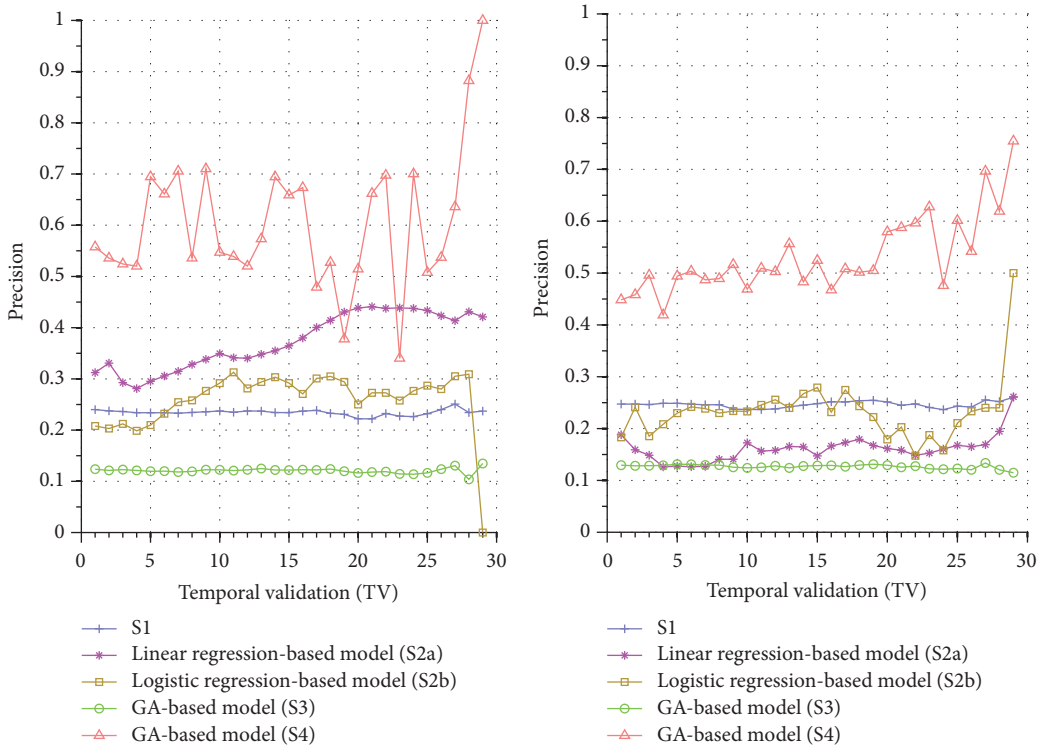
TV	Averaged training precision					Averaged testing precision				
	S1	S2a	S2b	S3	S4	S1	S2a	S2b	S3	S4
1	0%	35.47%	26.05%	93.73%	97.25%	23.99%	31.23%	20.39%	12.37%	55.76%
2	25.94%	33.09%	26.30%	34.51%	100.00%	23.74%	33.08%	20.76%	12.12%	53.58%
3	29.66%	30.49%	26.17%	86.67%	98.43%	23.61%	29.27%	21.41%	12.28%	52.43%
4	28.58%	32.59%	24.80%	65.10%	99.61%	23.40%	28.10%	22.65%	12.12%	52.02%
5	28.61%	30.27%	23.38%	57.65%	100.00%	23.38%	29.52%	24.66%	11.95%	69.46%
6	27.73%	28.99%	19.79%	79.22%	100.00%	23.38%	30.57%	26.25%	11.99%	66.10%
7	26.84%	31.02%	19.85%	55.29%	100.00%	23.30%	31.47%	24.54%	11.81%	70.57%
8	26.56%	32.27%	20.32%	70.59%	99.22%	23.44%	32.81%	24.95%	11.92%	53.59%
9	25.69%	31.66%	19.67%	75.29%	100.00%	23.58%	33.81%	25.90%	12.26%	71.02%
10	25.07%	30.84%	18.93%	68.63%	98.82%	23.72%	34.93%	28.19%	12.23%	54.71%
11	24.62%	31.38%	19.56%	74.12%	98.82%	23.48%	34.12%	26.49%	12.09%	53.91%
12	24.95%	31.71%	19.85%	83.92%	100.00%	23.73%	34.03%	27.46%	12.23%	52.03%
13	24.43%	31.33%	20.74%	35.69%	100.00%	23.70%	34.79%	27.14%	12.49%	57.34%
14	24.41%	31.94%	21.14%	75.69%	100.00%	23.42%	35.49%	26.06%	12.21%	69.45%
15	24.67%	31.64%	21.79%	86.67%	98.43%	23.42%	36.47%	25.24%	12.19%	65.91%
16	24.59%	31.57%	21.46%	86.27%	99.22%	23.71%	37.99%	26.97%	12.27%	67.34%
17	24.26%	31.81%	21.68%	65.10%	99.61%	23.85%	40.03%	26.82%	12.21%	47.89%
18	24.13%	31.79%	21.79%	98.82%	100.00%	23.28%	41.43%	28.21%	12.38%	52.73%
19	24.46%	31.58%	22.43%	72.16%	81.18%	23.08%	43.05%	26.69%	11.96%	37.78%
20	24.50%	31.34%	22.94%	91.37%	99.61%	22.20%	43.84%	26.77%	11.61%	51.45%
21	24.82%	30.68%	23.54%	58.43%	99.22%	22.20%	44.10%	24.88%	11.80%	66.22%
22	24.72%	30.64%	23.38%	83.53%	100.00%	23.22%	43.76%	26.25%	11.88%	69.75%
23	24.27%	30.60%	24.90%	72.16%	99.22%	22.72%	43.86%	27.06%	11.44%	34.03%
24	24.38%	30.42%	24.97%	60.39%	99.61%	22.63%	43.75%	26.65%	11.37%	70.04%
25	24.34%	30.50%	24.58%	49.80%	99.61%	23.19%	43.36%	26.68%	11.65%	50.78%
26	24.17%	30.61%	24.82%	77.25%	100.00%	24.00%	42.31%	26.72%	12.35%	53.71%
27	24.01%	30.97%	24.98%	74.12%	99.61%	25.07%	41.36%	26.34%	13.06%	63.60%
28	23.90%	31.04%	25.16%	87.45%	99.22%	23.43%	43.10%	25.97%	10.37%	88.24%
29	24.05%	31.51%	24.93%	71.37%	98.43%	23.70%	42.09%	0.00%	13.48%	100.00%

of time from Sept./15/2015 through Nov./03/2015. In this table, an inspection on the means shows that in all the 29 TVs of the training case the GA-based model S4 outperforms all the other models. For the testing phase, except TVs 19 and 23 in which S4 underperforms S2a, S4 still outperforms all the other systems in the remaining 27 TVs.

Figure 4(a) further displays a visual gist on this performance discrepancy of the five systems in the testing phase. In this figure, we notice that all the precisions by S1 are around 0.25. For S2a, most of the precisions are between 0.27 and 0.45. For S2b, most of the precisions are between 0.2 and 0.3, except the final TV where the model did not find any TPs with testing data of only one day available. Similar phenomenon can be seen for other companies, for example, the precision of S3 in the final TV of Figure 5(a) and those of S2b in the last 3 TVs in Figure 5(b) (notice that it does not mean that a random coin-toss strategy is better than these systems; the precisions of these systems are below 0.5 because we remove the flat price cases in computing the TP, as discussed in Section 2.2.4, which thus presents substantial challenge to the forecasting task).

For S3, the results are quite poor where all the precisions are between 0.1 and 0.15. An investigation into these unsatisfactory performances indicates that even though the model detects that the demand for a stock is more than the supply and thus expects the price to go up, the price may still remain unchanged for next several calls and finally go up after that. In other words, using the price of the next call may not be the best timing for calculating precision. Therefore, in S4, our proposed system allows the GA to evolve a better timing for computing the precision and the results indeed show that it improves the performance of the system significantly. However, we also notice that the precisions suffer high variance, indicating that the performance of system S4 is not very stable across various TVs (different TVs have different number of training and testing days). This is perhaps because the period of the testing phase is not long enough for S4 to exhibit stable performance in the final few TVs, and we intend to investigate this issue in more detail in the future.

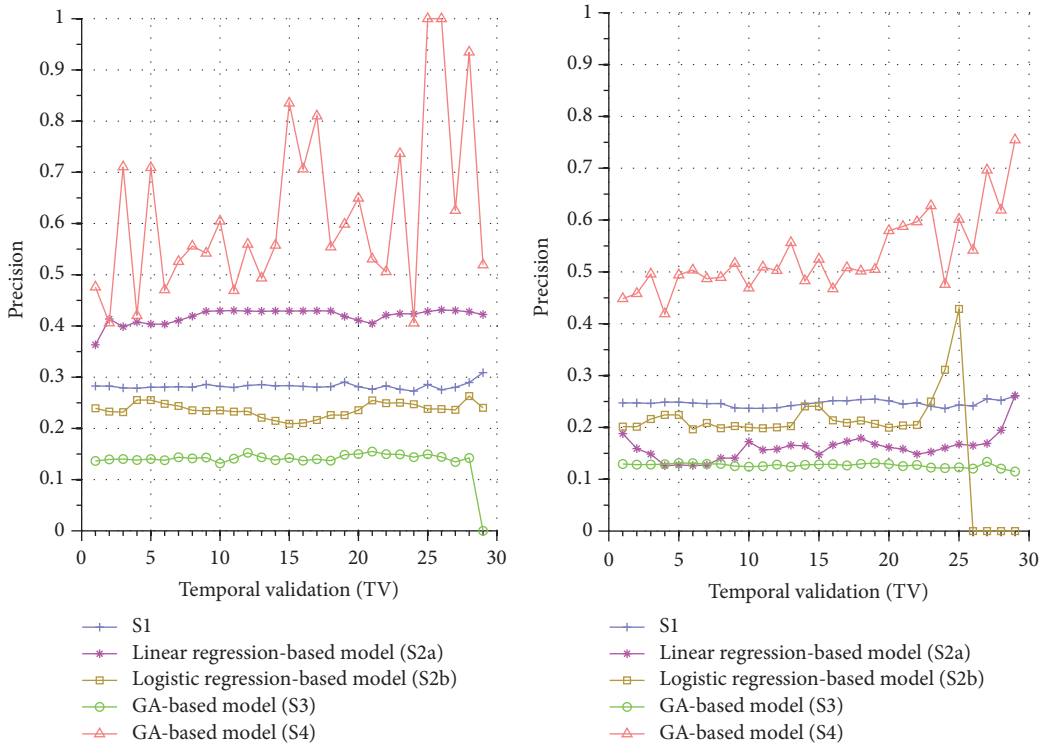
Similar performance discrepancy of our proposed GA-based S4 model and other methods can be seen for the other stocks. For instance, Figure 4(b) shows the results for Siliconware Precision Industries Co., Ltd. Figures 5(a) and 5(b) show



(a) Advanced Semiconductor Engineering Inc.

(b) Siliconware Precision Industries Co., Ltd.

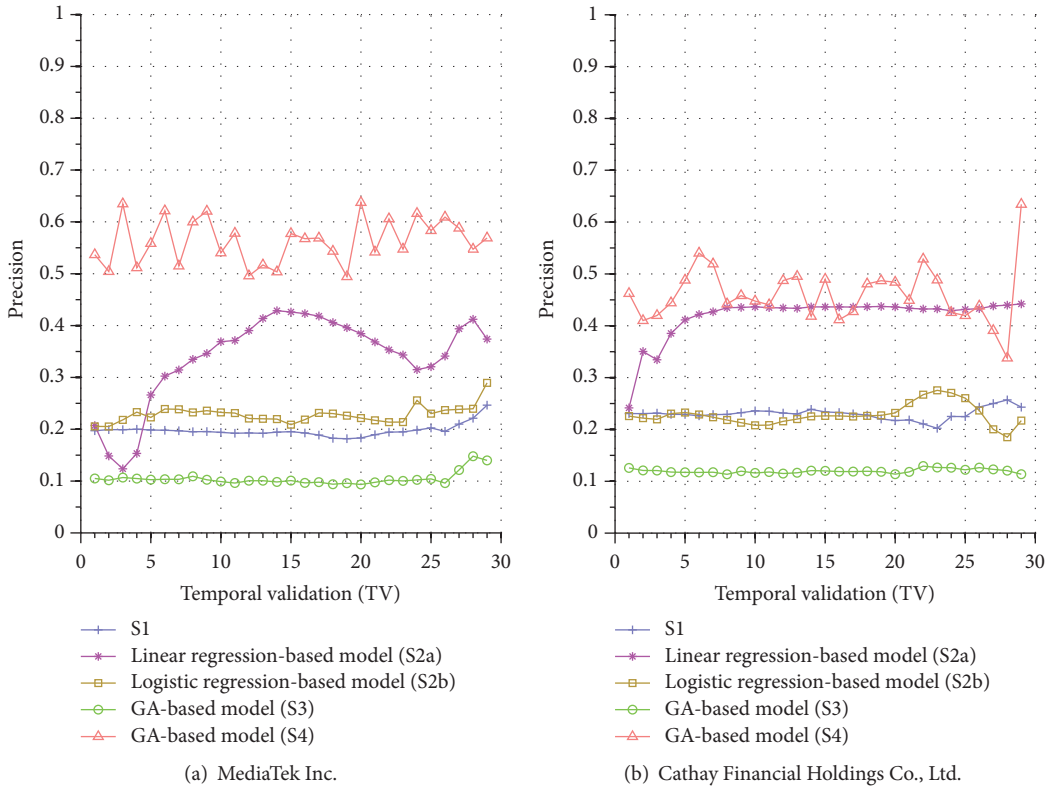
Figure 4: Precision comparison of the forecasting systems in the testing phase for 9/15/2015–11/3/2015.



(a) Taiwan Semiconductor Manufacturing Company Limited

(b) AU Optronics Corp.

Figure 5: Precision comparison of the forecasting systems in the testing phase for 9/15/2015–11/3/2015.



(a) MediaTek Inc. (b) Cathay Financial Holdings Co., Ltd.

Figure 6: Precision comparison of the forecasting systems in the testing phase for 9/15/2015–11/3/2015.

the results for Taiwan Semiconductor Manufacturing Company Limited and AU Optronics Corp., respectively. Figures 6(a) and 6(b) show the results for MediaTek Inc. and Cathay Financial Holding Co., Ltd. Figures 7(a) and 7(b) show the results for China Development Financial Holding Corporation and Yuanta Financial Holding Co., Ltd. Figures 8(a) and 8(b) show the results for Mega Financial Holding Company and CTBC Financial Holding Co., Ltd. As can be seen, the results in these ten stocks have shown that S4 outperforms S1, S2a, S2b, and S3 in most TV's, thereby indicating the effectiveness of our proposed S4 system.

The results just shown were obtained using the data from Sept./15/2015 to Nov./03/2015, during which time the broad stock market encountered favorable conditions and delivered positive gain as mentioned previously. In contrast, here we also display the results using the data from Dec./10/2015 to Jan./21/2016, during which time the market faced adverse situations and thus made loss. The results are shown in Figures 9(a) and 9(b) for Advanced Semiconductor Engineering Inc. and Siliconware Precision Industries Co., respectively. Figures 10(a) and 10(b) are for Taiwan Semiconductor Manufacturing Company Ltd. and AU Optronics Corp., respectively. Figures 11(a) and 11(b) are for MediaTek Inc. and Cathay Financial Holding Co., respectively. Figures 12(a) and 12(b) are for China Development Financial Holding Corporation and Yuanta Financial Holding Co., respectively. Figures 13(a) and 13(b) are for Mega Financial Holding Company and CTBC Financial Holding Co., respectively. As can be seen again, the results in these ten stocks have shown that S4 outperforms S1, S2a, S2b, and S3 in most TV's, thereby indicating the effectiveness of our proposed S4 system.

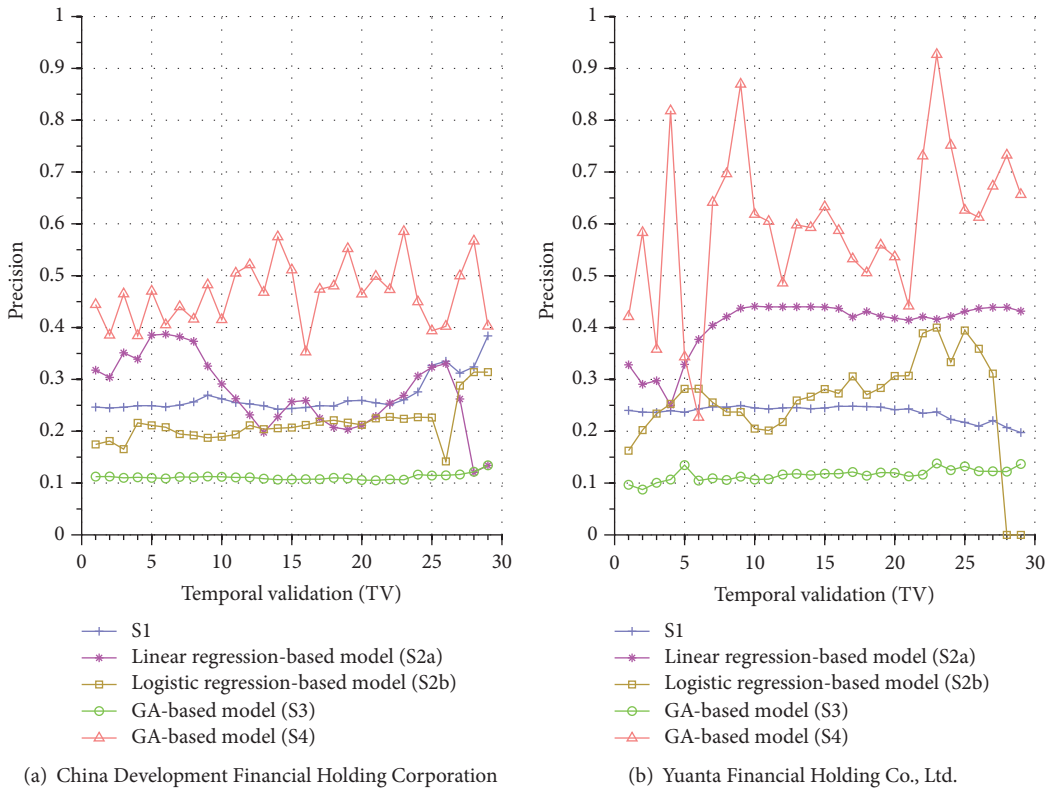
From these results, it appears that there is an upward trend of the precisions over training data size, for example, Figures 4(a), 4(b), 5(a), and 5(b) and others. However, there is also a downward trend of the precisions over training data size, for example, Figures 8(a) and 12(a), although the upward trend appears to be more frequent. For this situation, our conjecture is that, with more training data, the GA-based system S4 is more capable of learning to acquire better models for forecasting, and we intend to investigate this issue in the future.

Furthermore, in order to show the effectiveness of S4, Table 6 provides a summary that displays the mean and standard deviation computed for the precisions shown in each of Figures 4(a)–13(b). In this table, it can be seen that the higher value of the averaged precision for system S4 in each figure indicates this system outperforms the other systems, although many of the corresponding standard deviations are still relatively high. As we have already mentioned previously, this is again perhaps because the period of the testing phase in the final few TVs is not long enough for S4 to exhibit stable performance along temporal validations. Therefore, we intend to investigate this issue for the future work in order to achieve a more stable prediction performance for our proposed system S4.

All the results presented thus far were obtained using the precision metric by (7). For the accuracy metric, for illustrations, we display the results in Figures 14 and 15 using

Table 6: Averaged precisions and standard deviations for Figures 4(a)–13(b).

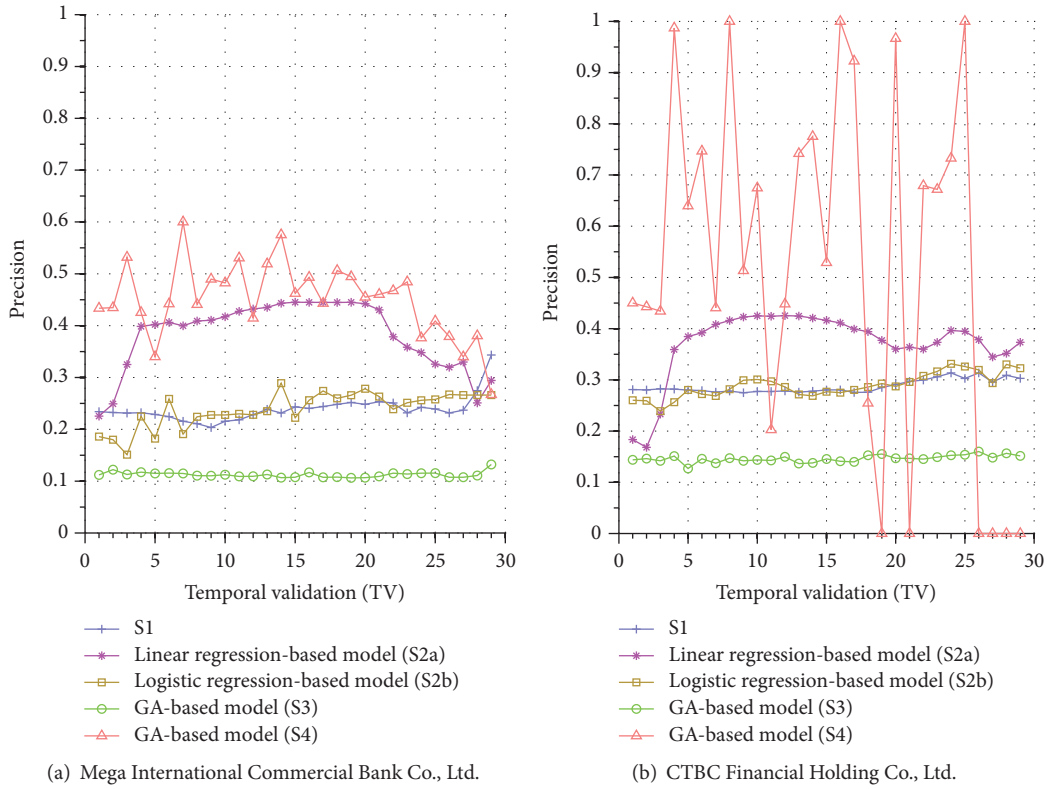
	Mean					Standard deviation				
	S1	S2a	S2b	S3	S4	S1	S2a	S2b	S3	S4
Figure 4(a)	0.2343	0.3737	0.2589	0.1207	0.6039	0.0055	0.0542	0.0606	0.0054	0.135
Figure 4(b)	0.2464	0.1611	0.2339	0.1267	0.5317	0.0061	0.026	0.0604	0.0041	0.0753
Figure 5(a)	0.2825	0.4191	0.2363	0.1375	0.6177	0.0063	0.0147	0.0137	0.027	0.1674
Figure 5(b)	0.2464	0.1611	0.1929	0.1267	0.5317	0.0061	0.026	0.0908	0.0041	0.0753
Figure 6(a)	0.1973	0.3394	0.2284	0.1042	0.5633	0.0121	0.085	0.0165	0.0123	0.0435
Figure 6(b)	0.2301	0.4189	0.2281	0.1196	0.4609	0.0108	0.0425	0.0203	0.0041	0.055
Figure 7(a)	0.2672	0.2747	0.214	0.1114	0.365	0.0346	0.0711	0.0375	0.0057	0.0605
Figure 7(b)	0.2366	0.4052	0.2589	0.1159	0.5988	0.0138	0.0526	0.0924	0.0116	0.1536
Figure 8(a)	0.2387	0.3822	0.2387	0.1124	0.451	0.0248	0.0663	0.0339	0.0055	0.0719
Figure 8(b)	0.2872	0.3717	0.2891	0.1461	0.526	0.0127	0.0665	0.0237	0.0069	0.348
Figure 9(a)	0.258	0.2714	0.1451	0.1768	0.4352	0.0226	0.0979	0.0181	0.0744	0.0464
Figure 9(b)	0.1764	0.3476	0.2737	0.0791	0.5032	0.0155	0.0803	0.0109	0.0081	0.0391
Figure 10(a)	0.25	0.26	0.2585	0.1236	0.4507	0.0123	0.0868	0.0108	0.0043	0.1299
Figure 10(b)	0.2966	0.1762	0.3215	0.1233	0.437	0.0304	0.0517	0.1119	0.006	0.1537
Figure 11(a)	0.1741	0.114	0.126	0.1054	0.4238	0.0225	0.0459	0.1092	0.0959	0.091
Figure 11(b)	0	0.3617	0.1808	0.1115	0.4782	0	0.0813	0.0899	0.0098	0.0475
Figure 12(a)	0.3336	0.1773	0.2049	0.0888	0.4533	0.105	0.07	0.0306	0.0182	0.0646
Figure 12(b)	0.1985	0.3915	0.2108	0.1036	0.5726	0.0371	0.0782	0.0274	0.0096	0.0921
Figure 13(a)	0.3109	0.1127	0.1624	0.1042	0.4986	0.0752	0.0105	0.0216	0.003	0.0732
Figure 13(b)	0.2537	0.1399	0.2472	0.1261	0.5351	0.0255	0.0226	0.026	0.0071	0.0885



(a) China Development Financial Holding Corporation

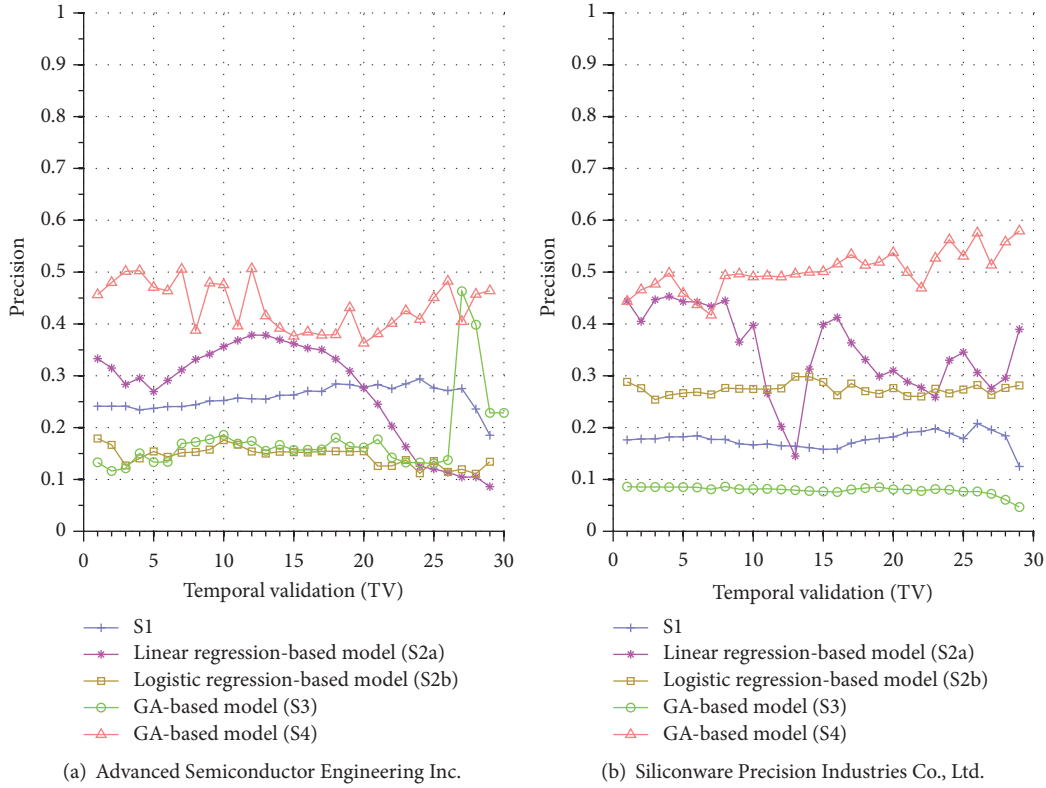
(b) Yuanta Financial Holding Co., Ltd.

Figure 7: Precision comparison of the forecasting systems in the testing phase for 9/15/2015–11/3/2015.



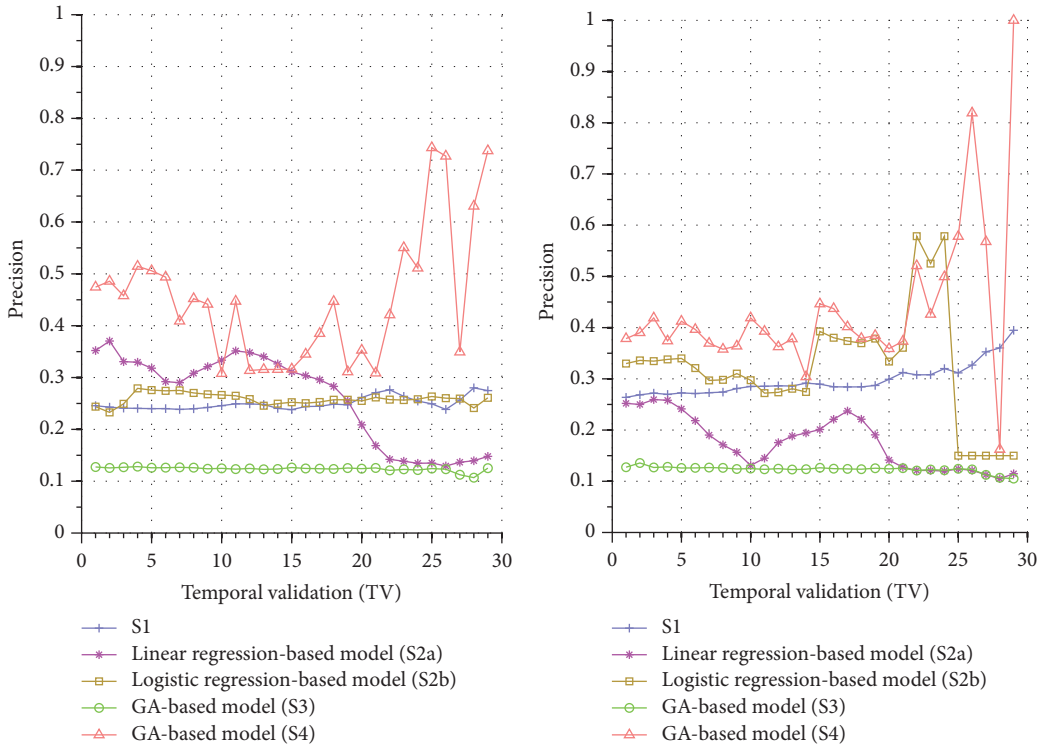
(a) Mega International Commercial Bank Co., Ltd. (b) CTBC Financial Holding Co., Ltd.

Figure 8: Precision comparison of the forecasting systems in the testing phase for 9/15/2015–11/03/2015.



(a) Advanced Semiconductor Engineering Inc. (b) Siliconware Precision Industries Co., Ltd.

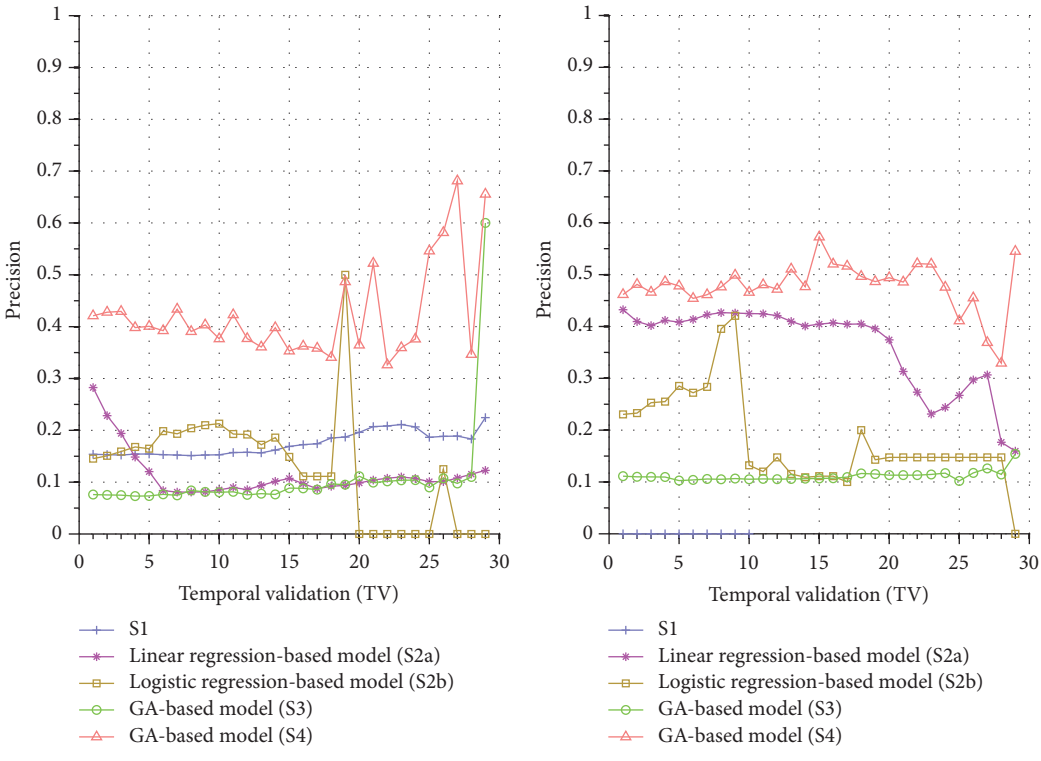
Figure 9: Precision comparison of the forecasting systems in the testing phase for 12/10/2015–1/21/2016.



(a) Taiwan Semiconductor Manufacturing Company Limited

(b) AU Optronics Corp.

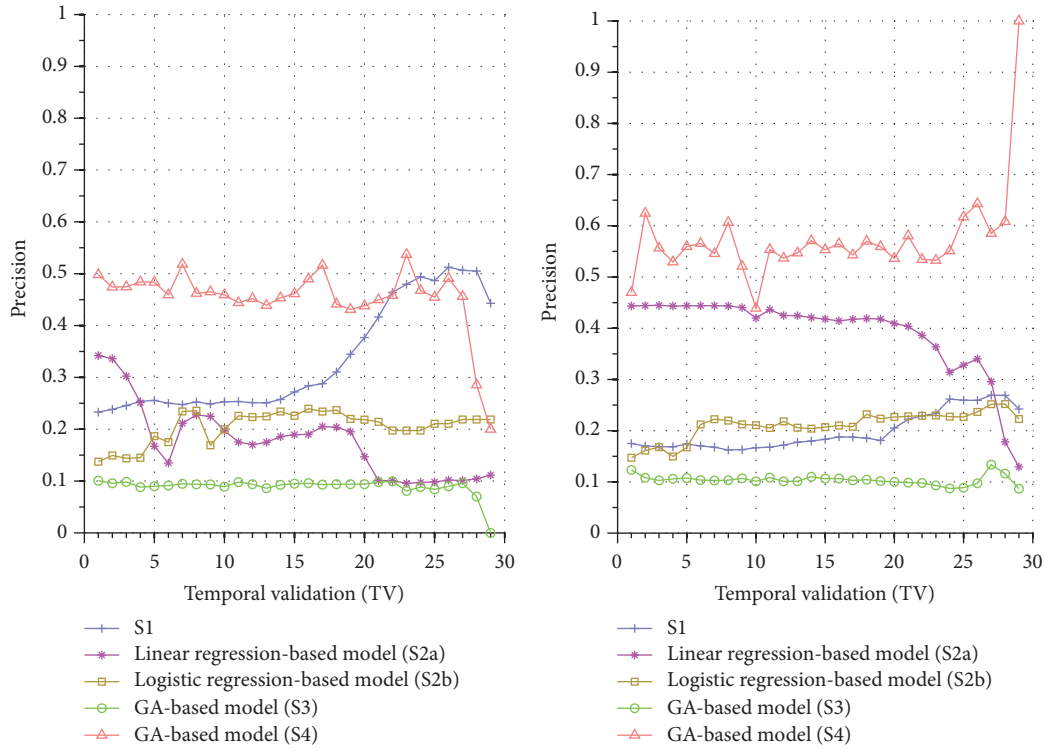
Figure 10: Precision comparison of the forecasting systems in the testing phase for 12/10/2015–1/21/2016.



(a) MediaTek Inc.

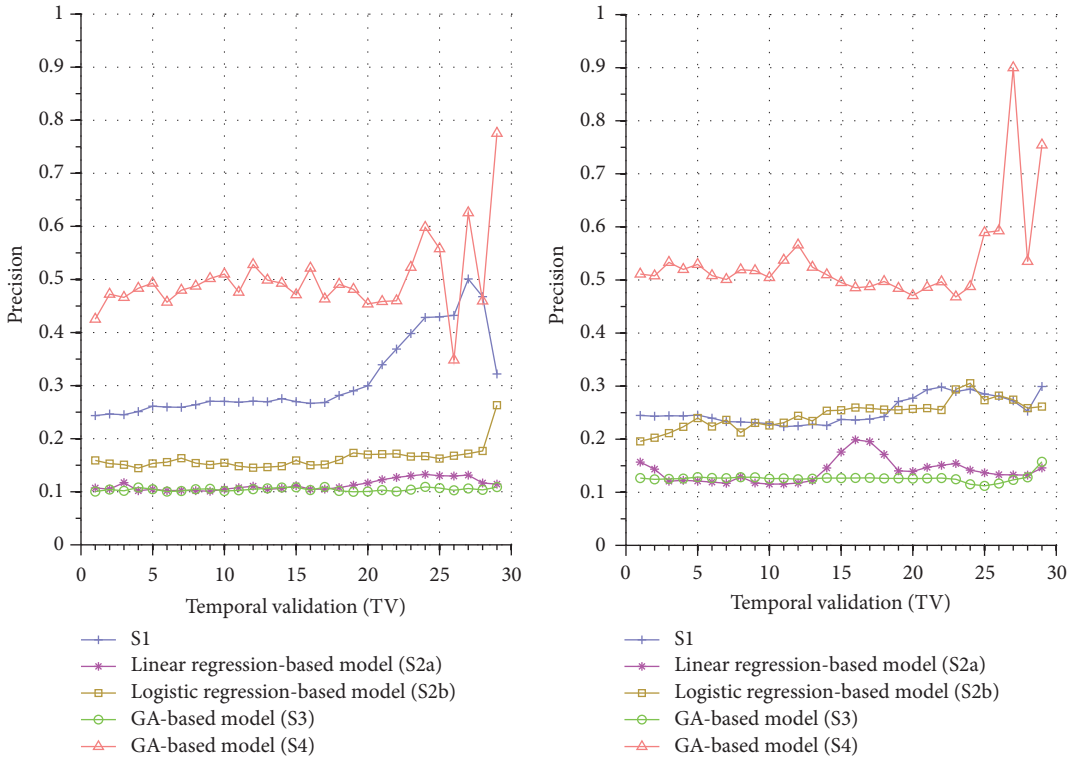
(b) Cathay Financial Holdings Co., Ltd.

Figure 11: Precision comparison of the forecasting systems in the testing phase for 12/10/2015–1/21/2016.



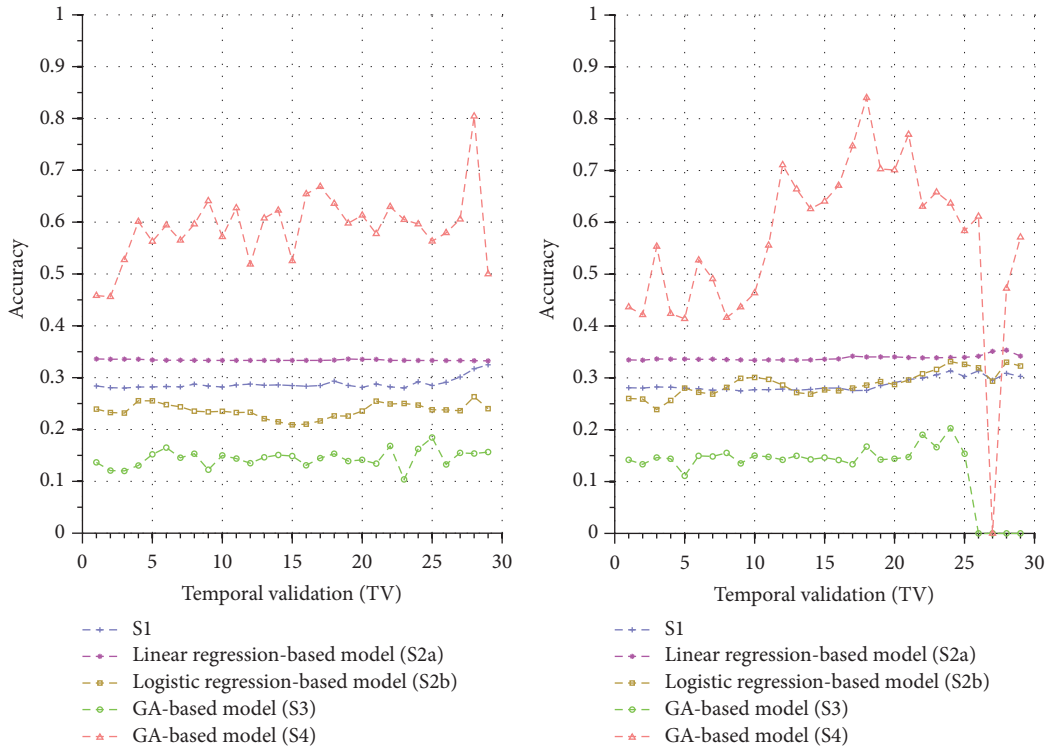
(a) China Development Financial Holding Corporation (b) Yuanta Financial Holding Co., Ltd.

Figure 12: Precision comparison of the forecasting systems in the testing phase for 12/10/2015–1/21/2016.



(a) Mega International Commercial Bank Co., Ltd. (b) CTBC Financial Holding Co., Ltd.

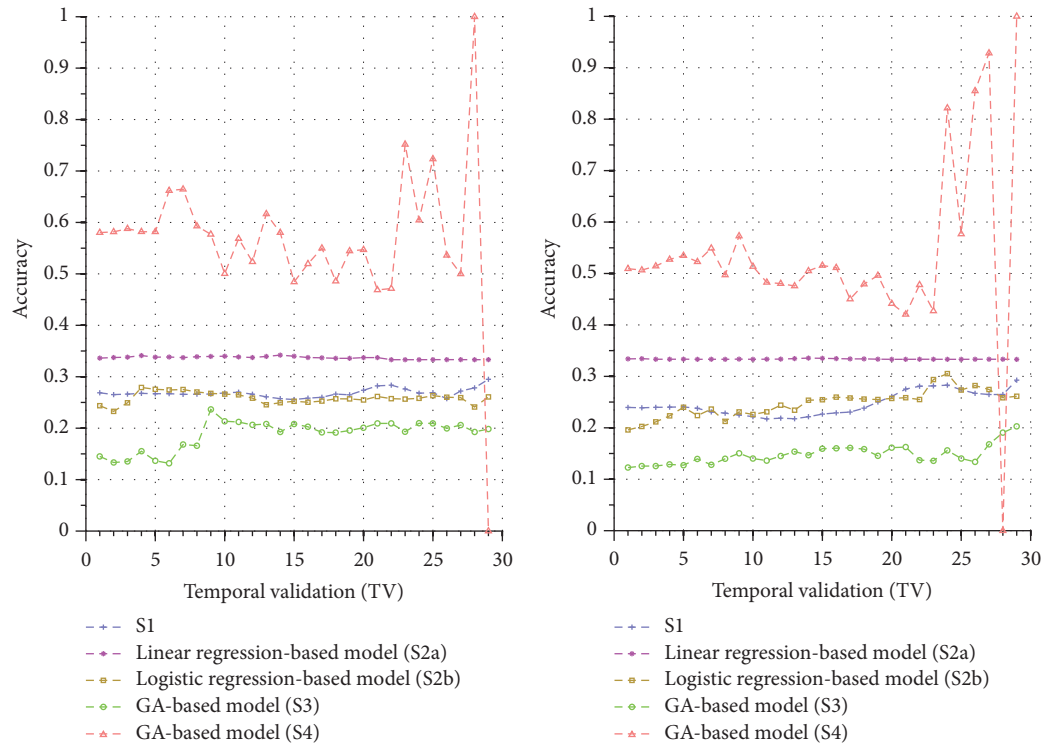
Figure 13: Precision comparison of the forecasting systems in the testing phase for 12/10/2015–1/21/2016.



(a) Taiwan Semiconductor Manufacturing Company Limited

(b) CTBC Financial Holding Co., Ltd.

Figure 14: Accuracy comparison of the forecasting systems in the testing phase for 9/15/2015–11/3/2015.



(a) Taiwan Semiconductor Manufacturing Company Limited

(b) CTBC Financial Holding Co., Ltd.

Figure 15: Accuracy comparison of the forecasting systems in the testing phase for 12/10/2015–1/21/2016.

only two companies for the two periods of Sept./15/2015 to Nov./03/2015 and Dec./10/2015 to Jan./21/2016 (since the results for other companies are similar to those in these two figures, they are skipped here). As can be seen again, the results in Figures 14 and 15 have shown that S4 outperforms S1, S2a, S2b, and S3 in most TVs, thereby indicating that our proposed S4 system is more effective than others under the accuracy metric, as well.

4. Conclusions

In this paper, we presented a GA-based methodology for the research of high-speed trading. The particular domain of this study centers around the call auction market, which provides significant amount of price data of stocks at the microscopic level in financial markets nowadays. Through the optimization of parameters of the forecasting models, our experimental results showed that the proposed GA-based method is able to improve the accuracy of prediction for price movement on the microscopic level. In order to further examine the validity of our models, we conducted a statistical validation on the learned models and showed that our proposed models are effective in the dynamic environment of stock market, which is critical for practical investment where one expects the models constructed to gain profits in the future. With these results, we expect this GA-based method to advance the research in computational intelligence for financial applications and provide useful insights for high-speed trading.

In the future, since there exist several studies that use returns instead of prices for modeling, we thus propose to use returns for financial modeling as a potential line of research for high-speeding trading. Furthermore, since the results show that the precisions suffer high variance, we also intend to investigate this issue in more detail in order to reduce the variance in precisions, so that our proposed system can deliver more stable performance.

In addition, in this work, we have provided studies using the precision and accuracy metrics for performance evaluation. It may be worthwhile to conduct more detailed comparisons using these two metrics to investigate if both metrics have delicate discrepancy on the effectiveness of the models we proposed. Other alternatives are also possible; for instance, in the design for the fitness measure of a chromosome, we may somehow penalize its fitness by the variance of performance of the models. For other future work, we also intend to investigate the reason that led to the upward tendency of the precisions over increasing size of the training data. Our goal is to investigate if there exists a suitable size of training datasets for the GA-based method to acquire models that shall deliver more stable performance.

Furthermore, in this work, we employed simple GA with binary coding to search for the optimal parameters for the models. Indeed, there are various types of GA that could be employed for this study, as well. However, since the research of high-speed trading is a relatively new and unexplored area to computational intelligence, in this work, our major goal is thus to show, in principle, the GA is a useful tool for finding promising forecasting systems in the context of high-speed trading. In the future, we thus intend to employ more sophisticated versions of the GA in order to further improve the performance of our proposed systems.

Competing Interests

The authors declare no conflict of interests.

Acknowledgments

This work is fully supported by the Ministry of Science and Technology, Taiwan, Republic of China, under Grant no. MOST-104-2221-E-390-019.

References

[1] J. Hasbrouck and G. Saar, "Low-latency trading," *Journal of Financial Markets*, vol. 16, no. 4, pp. 646–679, 2013.

[2] K. Swinburne, "On regulation of trading in financial instruments—'dark pools'," in *Report, Committee on Economic and Monetary Affairs*, European Parliament, Brussels, Belgium, 2010.

[3] R. J. Kauffman, Y. Hu, and D. Ma, "Will high-frequency trading practices transform the financial markets in the Asia Pacific Region?" *Financial Innovation*, vol. 1, no. 1, pp. 1–27, 2015.

[4] I. Aldridge, *High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems*, John Wiley & Sons, 2010.

[5] C.-F. Huang, "A hybrid stock selection model using genetic algorithms and support vector regression," *Applied Soft Computing Journal*, vol. 12, no. 2, pp. 807–818, 2012.

[6] R. C. Cavalcante, R. C. Brasileiro, V. L. F. Souza, J. P. Nobrega, and A. L. I. Oliveira, "Computational intelligence and financial markets: a survey and future directions," *Expert Systems with Applications*, vol. 55, pp. 194–211, 2016.

[7] J. Tang and L. He, "Genetic optimization of BP neural network in the application of suspicious financial transactions pattern recognition," in *Proceedings of the 6th International Conference on Management of e-Commerce and e-Government (ICMeCG '12)*, pp. 280–284, IEEE, Beijing, China, October 2012.

[8] L. Jing, "Data modeling for searching abnormal noise in stock market based on genetic algorithm," in *Proceedings of the International Symposium on Computational Intelligence and Design*, vol. 2, pp. 129–131, Hangzhou, China, October 2010.

[9] E. Tsang, P. Yung, and J. Li, "EDDIE-automation, a decision support tool for financial forecasting," *Decision Support Systems*, vol. 37, no. 4, pp. 559–565, 2004.

[10] C.-F. Huang, C.-J. Hsu, C.-C. Chen, B. R. Chang, and C.-A. Li, "An intelligent model for pairs trading using genetic algorithms," *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 939606, 10 pages, 2015.

[11] F. Varetto, "Genetic algorithms applications in the analysis of insolvency risk," *Journal of Banking & Finance*, vol. 22, no. 10-11, pp. 1421–1439, 1998.

[12] A. Gaspar-Cunha, G. Recio, L. Costa, and C. Estébanez, "Self-adaptive MOEA feature selection for classification of bankruptcy prediction data," *The Scientific World Journal*, vol. 2014, Article ID 314728, 20 pages, 2014.

[13] P. Parracho, R. Neves, and N. Horta, "Trading with optimized uptrend and downtrend pattern templates using a genetic

algorithm kernel," in *Proceedings of the IEEE Congress of Evolutionary Computation (CEC '11)*, pp. 1895–1901, New Orleans, La, USA, June 2011.

[14] R. D. A. Araújo and T. A. E. Ferreira, "A Morphological-Rank-Linear evolutionary method for stock market prediction," *Information Sciences*, vol. 237, pp. 3–17, 2013.

[15] D. Bernardo, H. Hagras, and E. Tsang, "A Genetic type-2 fuzzy logic based system for financial applications modelling and prediction," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE '13)*, pp. 1–8, Hyderabad, India, July 2013.

[16] P. Gupta, M. K. Mehlawat, and G. Mittal, "Asset portfolio optimization using support vector machines and real-coded genetic algorithm," *Journal of Global Optimization*, vol. 53, no. 2, pp. 297–315, 2012.

[17] A. Adebiyi Ayodele and K. Ayo Charles, "Portfolio selection problem using generalized differential evolution 3," *Applied Mathematical Sciences*, vol. 9, no. 42, pp. 2069–2082, 2015.

[18] V. Ranković, M. Drenovak, B. Stojanović, Z. Kalinić, and Z. Arsovski, "The mean-value at risk static portfolio optimization using genetic algorithm," *Computer Science and Information Systems*, vol. 11, no. 1, pp. 89–109, 2014.

[19] R. Aguilar-Rivera, M. Valenzuela-Rendón, and J. J. Rodríguez-Ortiz, "Genetic algorithms and Darwinian approaches in financial applications: a survey," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7684–7697, 2015.

[20] K.-J. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert Systems with Applications*, vol. 19, no. 2, pp. 125–132, 2000.

[21] M. Shao, D. Smonou, M. Kampouridis, and E. Tsang, "Guided Fast Local Search for speeding up a financial forecasting algorithm," in *Proceedings of the IEEE Conference on Computational Intelligence for Financial Engineering and Economics (CIFEr '14)*, pp. 325–332, London, UK, March 2014.

[22] E. P. Tsang, J. Li, S. Markose, H. Er, A. Salhi, and G. Iori, "EDDIE in financial decision making," *Journal of Management and Economics*, vol. 4, no. 4, 2000.

[23] J. Peralta Donate and P. Cortez, "Evolutionary optimization of sparsely connected and time-lagged neural networks for time series forecasting," *Applied Soft Computing Journal*, vol. 23, pp. 432–443, 2014.

[24] S. Hu and C. Chan, "Infer the order imbalance in a call auction market-evidence from taiwan stock market," *Journal of Financial Studies*, vol. 16, no. 1, pp. 19–63, 2008.

[25] C.-F. Huang, T.-N. Hsieh, B.-R. Chang, and C.-H. Chang, "A comparative study of stock scoring using regression and genetic-based linear models," in *Proceedings of the IEEE International Conference on Granular Computing (GrC '11)*, pp. 268–273, IEEE, Kaohsiung, Taiwan, November 2011.

[26] C.-F. Huang, T.-N. Hsieh, B. R. Chang, and C.-H. Chang, "A comparative study of regression and evolution-based stock selection models for investor sentiment," in *Proceedings of the 3rd International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA '12)*, pp. 73–78, IEEE, Kaohsiung, Taiwan, September 2012.

[27] A. Kalay, O. Sade, and A. Wohl, "Measuring stock illiquidity: an investigation of the demand and supply schedules at the TASE," *Journal of Financial Economics*, vol. 74, no. 3, pp. 461–486, 2004.

[28] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.

[29] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *Foundation of Genetic Algorithms*, pp. 69–93, 1991.

[30] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.

[31] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Mass, USA, 1996.

Hindawi
Computational Intelligence and Neuroscience
Volume 2017, Article ID 7436948, 9 pages
https://doi.org/10.1155/2017/7436948

Research Article

Statistical Modeling and Prediction for Tourism Economy Using Dendritic Neural Network

Ying Yu, Yirui Wang, Shangce Gao, and Zheng Tang

Faculty of Engineering, University of Toyama, Toyama-shi 930-8555, Japan

Correspondence should be addressed to Shangce Gao; gaosc@eng.u-toyama.ac.jp

Received 4 September 2016; Revised 16 December 2016; Accepted 4 January 2017; Published 26 January 2017

Academic Editor: Francisco Martínez-Álvarez

With the impact of global internationalization, tourism economy has also been a rapid development. The increasing interest aroused by more advanced forecasting methods leads us to innovate forecasting methods. In this paper, the seasonal trend autoregressive integrated moving averages with dendritic neural network model (SA-D model) is proposed to perform the tourism demand forecasting. First, we use the seasonal trend autoregressive integrated moving averages model (SARIMA model) to exclude the long-term linear trend and then train the residual data by the dendritic neural network model and make a short-term prediction. As the result showed in this paper, the SA-D model can achieve considerably better predictive performances. In order to demonstrate the effectiveness of the SA-D model, we also use the data that other authors used in the other models and compare the results. It also proved that the SA-D model achieved good predictive performances in terms of the normalized mean square error, absolute percentage of error, and correlation coefficient.

1. Introduction and Literature Review

With the impact of global internationalization, tourism is also in a state of rapid development. As we all know, tourism's impact on the economic and social development of a country can be enormous. It can not only up business, trade, and capital investment but also create jobs and entrepreneurialism for workforce and protect heritage and cultural values (as shown in Table 1). Each country wants to know the data of its inbound visitors and tourism in order to choose an appropriate strategy for its economic well-being. Hence, a reliable forecast is needed and plays a major role in tourism planning.

Accurate forecasts build the foundation for better tourism planning and administration. Then more efficient forecasting techniques in tourism demand studies are being called for.

Over the past two decades, tourism demand modeling and forecasting which are two of the most important areas in tourism research have attracted more and more attention of both academics and practitioners. As Song and Li concluded, twenty years ago, there were only a handful of academic journals that published tourism-related research [1]. Now there are more than 70 journals that serve a thriving research community covering more than 3000 tertiary institutions across five continents. However, there has not been a panacea for tourism demand forecasting.

In recent years, statistics has been widely applied to the tourism economy under study. Among the statistical methods, time series forecasting is an important area of forecasting. And it can be classified into two categories: the linear methods and the nonlinear methods. The most popular of the linear methods are the Naïve model [2–5], the exponential smoothing (ES) model [2, 6], and the autoregressive integrated moving averages (ARIMA) model [3, 4, 6]. Among them, the most advanced forecasting model of linear methods is autoregressive integrated moving average model (ARIMA) which has been successfully tested in many practical applications. If the linear models can approximate the underlying data generating process well, they could be considered as the preferred models. However, if the linear models fail to perform well in both in-sample fitting and out-of-sample forecasting, more complex nonlinear models should be considered. Based on this view, many scholars have also turned to nonlinear methods such as the neural

Table 1: Inbound tourism consumption.

Products	(Billion Yen)		
	Same-day visitors	Tourists	Total visitors
Characteristic products	0	1167	1167
Accommodation services	0	496	496
Food and beverage servicing services	0	303	303
Passenger transport services	0	328	328
Travel agency, tour operator, and tourist guide services	0	8	8
Cultural services	0	10	10
Recreation and other entertainment services	0	8	8
Miscellaneous tourism service	0	14	14
Connected products	0	483	483
Total	0	1650	1650

network (NN) [3, 4, 7, 8]. Although there are still a few doubts about neural network based tourism demand forecasting, it is generally believed that the nonlinear methods outperform the linear methods in modeling the economic behavior and efficiently helping wise decision-making.

Neuron networks have been regarded by many experts as a promising technology for time series forecasting. Consequently, in the last few decades, more than 2000 articles on neural network forecasting have been published covering a wide range of applications [9]. Compared to statistical forecasting techniques, neural network approaches have several unique characteristics, such as (1) being both nonlinear and data driven, (2) having no requirement for an explicit underlying model, and (3) being more flexible and universal and thus applicable to more complicated models [10]. Furthermore, Nelson et al. and Zhang and Kline [11, 12] suggested that time series preprocessing (e.g., detrending and deseasonalizing) contributes significantly to neuron network model performance.

Up to now, there are many researchers using a lot of methods to forecast the tourism demand. And they can be divided into three types: time series, neural network, and combined models. In 2014, Teixeira and Fernandes published [13], in which the three methods are all mentioned. Except those, there are also a lot of authors using the three methods separately. For example, Box et al., Cho, Chu, Song, and Li, Law, Qu, and Zhang, Shahrabi et al., Li et al., and Kawakubo and Kubokawa have used the traditional time series methods to forecast the tourism demand [1, 3–7, 14–17]. As neural network is widely known, there are many authors turning to use the neural network to forecast the time series data such as Chen et al., Claveria and Torra, Davies et al., Constantino et al., Law, Lin et al., and Pai and Hong [3, 4, 8, 18–22]. With the progress of science, more and more methods are being used. The combined models are the most popular methods in them. And, up to now, Bates and Granger, Chen, Shen et al., and Yan have used this method and got the expected results [23–26]. Besides these, some other methods such as support vector regression [27, 28] and novel hybrid system [29, 30] are proposed. They have made great achievements in the optimization problem and the prediction problem; however, the data preprocessing and the late parameter selection problem are relatively complex.

When analyzing time series data, we should pay particular attention to the seasonality of the time series involved. Seasonality is a notable characteristic of tourism demand and cannot be ignored in the modeling process when monthly data are used. How to handle the seasonal fluctuations of tourism data has always been an important issue in tourism demand forecasting. We always use normal quantile transform or seasonal difference method to eliminate the impact of seasonality [31, 32].

In this paper, we mix the most advanced linear model (SARIMA model) with the innovative neural network model (DNN model) together and call the mixed model SA-D model. We obtained that the SA-D model performs much better than the DNN model in the tourism demand forecasting as the comparing results showed.

This paper is organized as follows. In Section 2, the SARIMA model, the DNN model, and the combined model (SA-D model) are described. Section 3 describes the data set and discusses the evaluation methods to compare the forecasting methods and takes statistical tests to check the SA-D model and then compares the models that other authors had given by using the same data. After that, the experimental results are given. Section 4 provides concluding remarks.

2. Modeling (Statistical Modeling and Neural Network)

A time series model explains a variable with regard to its own past and a random disturbance term. Time series models have been widely used for tourism demand prediction in the past four decades. In this section, two models are described as follows.

2.1. ARIMA Model and SARIMA Model. ARIMA is the most popular linear model for forecasting time series. It has made great success in both academic research and industrial applications. A general ARIMA model is ordered by (p, d, q), and it can be written as

$$\phi(B)\nabla^d x_t = \theta(B)\varepsilon_t, \tag{1}$$

where x_t and ε_t represent the number of visitors and random error terms at time t, respectively. B is a backward shift operator defined by $Bx_t = x_{t-1}$ and related to ∇ by $\nabla = 1 - B$; $\nabla^d = (1-B)^d$; d is the order of differencing. $\phi(B)$ and $\theta(B)$ are autoregressive (AR) and moving averages (MA) operators of orders p and q, respectively, and they are defined as

$$\begin{aligned}\phi(B) &= 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p,\\ \theta(B) &= 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q.\end{aligned} \tag{2}$$

$\phi_1, \phi_2, \ldots, \phi_p$ are the autoregressive coefficients and $\theta_1, \theta_2, \ldots, \theta_q$ are the moving average coefficients.

When fitting ARIMA model to the raw data, the ARIMA model involves the following four steps:

(I) Identification of the ARIMA (p, q, d) structure

(II) Estimation of the unknown parameters

(III) Goodness-of-fit tests on the estimated residuals

(IV) Forecast future outcomes based on the known data

ε_t should be independently and identically distributed as normal random variables with mean = 0 and constant variance = σ^2. The roots of $\phi_p(x_t) = 0$ and $\theta_q(x_t) = 0$ should all lie outside the unit circle. It was suggested by Box et al. that at least 50 or preferably 100 observations should be used for the ARIMA model [14].

If the data has significant seasonal changes periodically. We can use the SARIMA model which uses the seasonal difference method to eliminate the effects of seasonal cycles. However, if the seasonality is regarded as deterministic, introducing seasonal dummies into the time series models would be sufficient in accounting for the seasonal variation. To test for the presence of seasonal unit roots, the HEGY test [33] is widely used. Unlike the HEGY test, an alternative method known as the test for fractional integration to test the seasonal components in the time series was introduced in 2004 [34]. Another approach to model seasonal fluctuations is to use the periodic autoregressive model. This model allows parameters to vary according to the seasons of a year and therefore may reflect seasonal economic decision-making more adequately than constant parameter specifications.

2.2. DNN Model (Neuron Model with Dendritic Nonlinearity). Recently, more and more nonlinear forecasting models are proposed to address the time series' issues. As Song and Li concluded, among them, ANNs (artificial neural networks) are receiving increasing interests due to their ability to imperfect data, functions of self-organizing, self-study, data-driven, associated memory, and arbiter function mapping [1].

As we all know, the structure of every neuron is unique; it contains three parts: the cell body, dendrite, and axon. The dendrite receives the signal from other neurons; then the signal is computed at the synapse and transmitted to the cell body. If the signal into the cell body exceeds the holding threshold, the cell will fire and send the signal down to other neurons through axon.

In 1943, a simple neuron model is proposed by McCulloch and Pitts in which the dendrites and synapses are independents and there are no effects on them from one to another (Figure 1) [35]. However, in 1987, Minsky and Papert indicated that the McCulloch-Pitts model is limited to solving complex problems [36].

Different from the McCulloch-Pitts model which does not consider the dendritic structure in the neuron, neuron model with dendritic nonlinearity model (DNN model) is proposed in our researches. The DNN model can be generalized as follows:

(1) The dendrites can be initialized by any arbitrary decision.

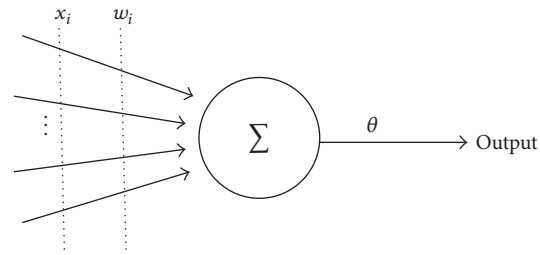


Figure 1: McCulloch-Pitts model.

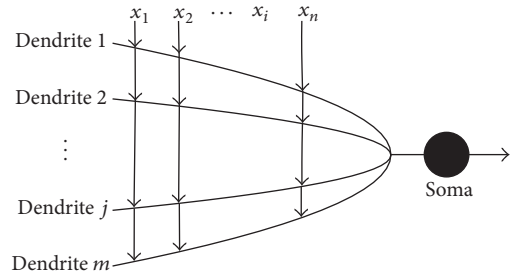


Figure 2: Neuron model with dendritic nonlinearity.

(2) The synapses on the same branch interact with each other.

(3) The nonlinear interaction produced in a dendrite can be expressed by a logical network.

(4) After learning, the branches' ripened number and the locations and types of synapses on the branches will be synthesized.

As shown in Figure 2, the dendritic branches receive signals from x_1 to x_n and then perform a simple multiplication on their own signal. At the junction of the branches, the outputs are summed up and then conducted to soma (the cell body). If the input of the soma exceeds a threshold, the cell will fire it and send it to other neurons through the axon.

Synaptic Function. In the connection layer, a sigmoid function reflects the interaction among the synapses in a dendrite. The output of the synapse whose address is from the ith ($i = 1, 2, \ldots, m$) input to the jth ($j = 1, 2, \ldots, n$) branch is given by the following equation:

$$Y_{ij} = \frac{1}{1 + e^{-k(w_{ij} - \theta_{ij})}}. \tag{3}$$

w_{ij} and θ_{ij}, respectively, mean the connection parameters, and k is a positive constant. When k becomes large enough, the sigmoid function will turn out to be similar to a step function. Through the change of the value of w_{ij} and θ_{ij}, four types of synaptic connections can be defined: a direct connection, an inverted connection, a constant-0 connection, and constant-1 connection.

Dendritic Function. It performs a simple multiplication on various synaptic connections of the branch. The output of the jth branch is given by

$$Z_j = \prod_{i=1}^{n} Y_{ij}. \tag{4}$$

Membrane Function. It is approximated as follows:

$$V = \sum_{j=1}^{m} Z_j. \tag{5}$$

Soma Function. The function of the soma is described by a sigmoid operation; when k is taken as a positive constant, γ is taken as a threshold from 0 to 1.

$$O = \frac{1}{1 + e^{-k(V-\gamma)}}. \tag{6}$$

Learning Function. Because DNN is a feed-forward network with continuous functions, the error back-propagation-like algorithm is valid for DNN. By using the learning rule, the error between the target vector and the actual output vector can be expressed as follows:

$$E = \frac{1}{2}(T - O)^2. \tag{7}$$

And, according to the gradient descent learning algorithm, the synaptic parameters w_{ij} and θ_{ij} can be modified in the direction to decrease the value of E. The equations are shown as follows:

$$\begin{aligned} \Delta w_{ij}(t) &= -\mu \frac{\partial E}{\partial w_{ij}}, \\ \Delta \theta_{ij}(t) &= -\mu \frac{\partial E}{\partial \theta_{ij}}, \end{aligned} \tag{8}$$

where μ is a positive constant that represents the learning rate. A low learning rate makes the convergence very slow, while a high learning rate is difficult for making the error converge. And the partial differentials of E with respect to w_{ij} and θ_{ij} are computed as follows:

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}} &= \frac{\partial E}{\partial O} \cdot \frac{\partial O}{\partial V} \cdot \frac{\partial V}{\partial Z_j} \cdot \frac{\partial Z_j}{\partial Y_{ij}} \cdot \frac{\partial Y_{ij}}{\partial w_{ij}}, \\ \frac{\partial E}{\partial \theta_{ij}} &= \frac{\partial E}{\partial O} \cdot \frac{\partial O}{\partial V} \cdot \frac{\partial V}{\partial Z_j} \cdot \frac{\partial Z_j}{\partial Y_{ij}} \cdot \frac{\partial Y_{ij}}{\partial \theta_{ij}}. \end{aligned} \tag{9}$$

2.3. The Combined Model (SA-D Model). Both linear and nonlinear models have achieved successes in their own linear or nonlinear problems. However, none of them is a universal model that is suitable for all situations. Bates and Granger said that a combined model having both linear and nonlinear modeling abilities will be a good alternative for forecasting the time series data [23]. Both the linear and nonlinear models have different unique strength to capture data characteristics in linear or nonlinear domains, so the combined model proposed in this study is composed of the linear component and the nonlinear component. Therefore, the combined model can model linear and nonlinear patterns with improved overall forecasting performance.

It may be reasonable to consider a time series to be composed of a linear autocorrelation structure and a nonlinear component which can be performed as

$$Y_t = L_t + N_t \quad (10), \tag{10}$$

where L_t is the linear component and N_t is the nonlinear component of the combined model. Both L_t and N_t have to be estimated for the data set. First, the author let linear model (here we use the SARIMA model to perform the obvious seasonal trends) to model the linear part; then the residuals from the linear model will contain only the nonlinear relationship. Let R_t represent the residual at time t; then we can know

$$R_t = Z_t - \widehat{L_t} \quad (11), \tag{11}$$

where $\widehat{L_t}$ denotes the forecast value of the linear model at time t. By modeling residuals using nonlinear model (here we use the DNN model), nonlinear relationships can be discovered. In this paper, we built the model with the following input layers:

$$R_t^{\text{linear}} = f^{\text{nonlinear}}\left(R_{t-1}^{\text{linear}}, R_{t-2}^{\text{linear}}, R_{t-3}^{\text{linear}}, R_{t-4}^{\text{linear}}\right) + e_t, \tag{12}$$

where R_t^{linear} represents the residual at time t from the ARIMA model, $f^{\text{nonlinear}}$ is a nonlinear function determined by the DNN model, and e_t is the random error. And the combined forecast can be performed as

$$\widehat{Y_t} = \widehat{L_t} + \widehat{N_t} \quad (13), \tag{13}$$

where $\widehat{N_t}$ is the forecast value of (12).

3. Results and Prediction

3.1. Data Set and the Process. Due to rapid economic growth and international tourism promotion, the number of tourists coming to Japan is greatly increasing year by year. Here we choose the inbound tourists from 2009:1 to 2015:12. And the process of data set is shown in Figure 3. The collected data were divided into two sets: the training data (data before 2015) and the testing data (data of 2015) [37, 38].

3.2. Evaluation Methods. Some quantitative statistical metrics such as normalized mean square error (NMSE), absolute percentage of error (APE), R (correlation coefficient), and program running time (PRT) are used to evaluate the forecasting performance of the forecasting models (Table 2). NMSE and APE are used to measure the deviation between the predicted and actual values. The smaller the values of NMSE and APE are, the closer the predicted values to the actual values are. The metric R is adopted to measure the correlation of the actual and the predicted values. The PRT can measure the running speed of the models.

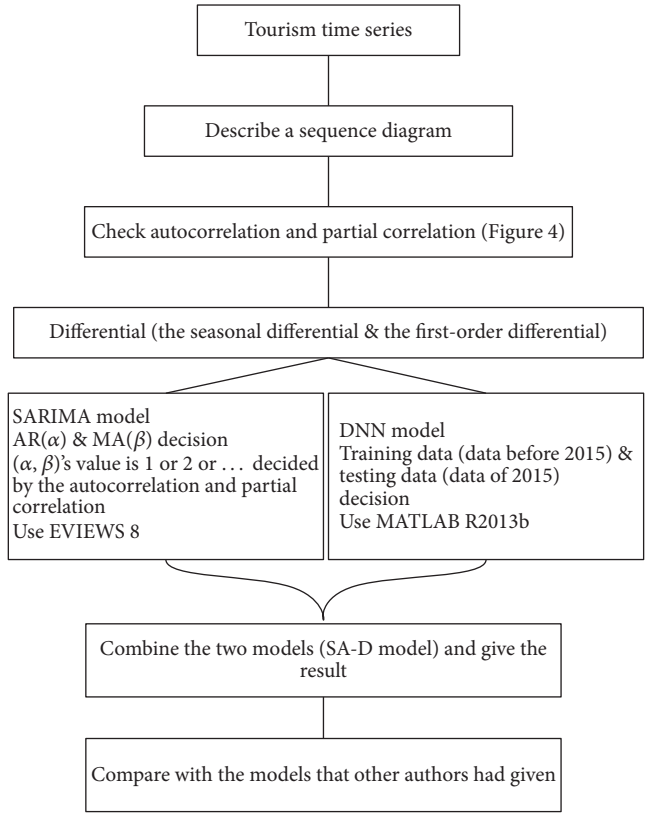


Figure 3: Process of data set.

Table 2: Calculations of the performance metrics.

Metrics	Calculation
NMSE	$\text{NMSE} = \frac{\sum_{i=1}^{n}(a_i - b_i)^2}{n\sigma^2}$; $\sigma = \frac{\sum_{i=1}^{n}(a_i - \overline{a})^2}{n-1}$
APE	$\text{APE} = \frac{\sum_{i=1}^{n} \lvert(a_i - b_i)/a_i\rvert}{n} \times 100\%$
R	$R = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \times \sqrt{\sum_{i=1}^{n} b_i^2}}$
PRT	Decided by the actual operation

Note: a_i and b_i are the actual values and the predicted values.

3.3. Experimental Results. For the data having significant seasonal changes periodically, we use the SARIMA model in this paper to eliminate the linear trend. As Figure 4 shows, we can decide the possible generations of the ARIMA model and use the Akaike Information Criterion (AIC) to test which of the generations is the best.

Through the SARIMA model, we get the data that has no linear trend and train the data separately by the DNN model and the SA-D model. We can get the results of the DNN model and the SA-D model as follows.

Table 3: The compared results of the DNN model and the SA-D model.

Metrics	The DNN model	The SA-D model
NMSE	2.245	0.219
APE	0.87	0.78
R	0.32	0.89
PRT	The DNN model is rapider than the SA-D model	

As Figures 5–7 show, we can see that the results of the SA-D model perform much better than those of the DNN model. In order to deeply evaluate the performance of the DNN model and the SA-D model, we calculate APE, NMSE, and R of the testing data set as Table 3 shows.

We can see that although the PRT of the DNN model is rapider than that of the SA-D model, the NMSE, APE, and R of the SA-D model are much better than those of the DNN model.

3.4. Models Comparison. To demonstrate the validity of the SA-D model, we train the same data that other authors had used in the other combination models and compare the results of the SA-D model and the other combination models. We collected the monthly outbound traveling population data of Taiwan to three areas (Americas, Europe and Oceania) from the Tourism Bureau, M.O.T.C. Republic of China

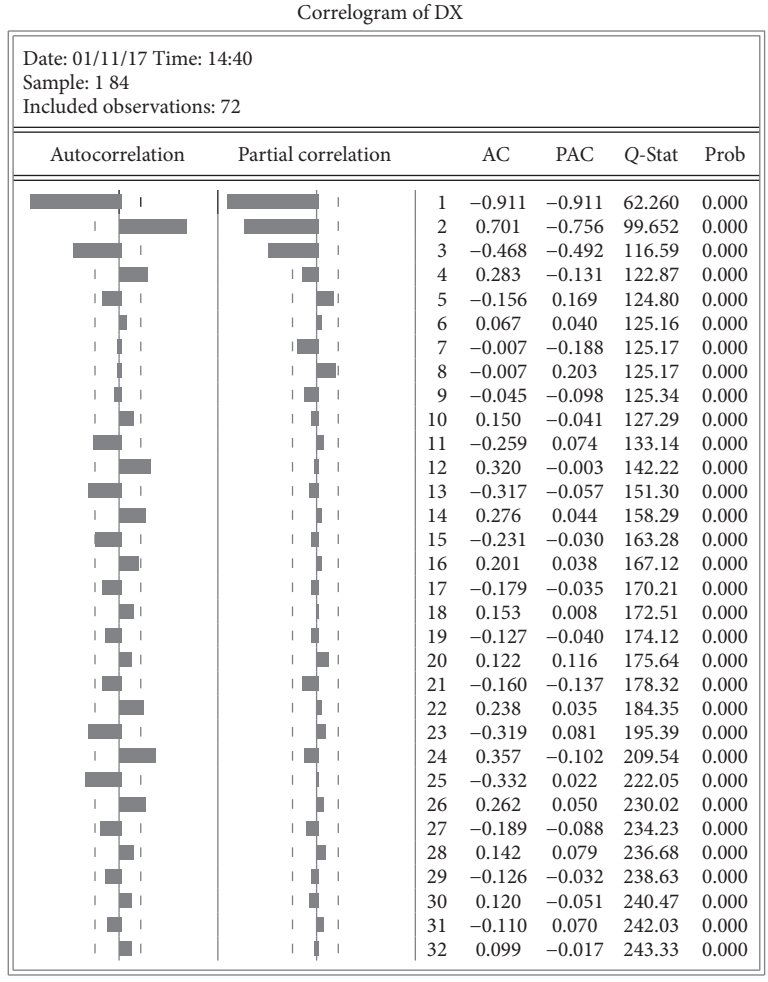


Figure 4: Autocorrelation and partial correlation.

Table 4: Results based on the orthogonal array factor assignment and statistical tests of the SA-D model.

Number	M	μ	k_{soma}	θ_{soma}	MSD	p
1	15	0.05	1	0	0.401 ± 0.169	0.1938
2	15	0.05	3	0.3	0.386 ± 0.170	0.2013
3	15	0.01	5	0.5	0.391 ± 0.171	0.1854
4	15	0.01	10	0.9	0.389 ± 0.167	0.191
5	25	0.05	1	0	0.392 ± 0.165	0.2563
6	25	0.05	3	0.3	0.395 ± 0.164	0.2742
7	25	0.01	5	0.5	0.398 ± 0.161	0.3011
8	25	0.01	10	0	0.390 ± 0.168	0.2916
9	30	0.05	1	0.9	0.402 ± 0.172	0.1928
10	30	0.05	3	0.3	0.399 ± 0.171	0.1897
11	30	0.01	5	0.5	0.394 ± 0.168	0.2001
12	30	0.01	10	0.9	0.397 ± 0.170	0.1936

Note: M means number of dendrites.

(Taiwan). The study time ranges from January of 1998 to June of 2009 [39]. The collected data were divided into two parts, training data (data from 1998 to 2007) and testing data (data after 2007), for each tourism demand time series. The

Table 5: Comparison of the SA-D model and the other combination models.

	Americas	Europe	Oceania
ARIMA + BPNN			
APE	13.41	12.95	13.46
NMSE	0.3992	0.8153	0.5327
R	0.9918	0.9917	0.9856
ARIMA + SVR			
APE	11.46	11.37	11.87
NMSE	0.2878	0.6316	0.5102
R	0.9923	0.9917	0.9871
The SA-D model (with data preset as other authors did)			
APE	9.61	9.73	9.89
NMSE	0.2788	0.4561	0.4968
R	0.9934	0.9921	0.9864
The SA-D model (without data preset)			
APE	10.34	10.51	10.87
NMSE	0.3458	0.5619	0.6027
R	0.9912	0.9906	0.9891

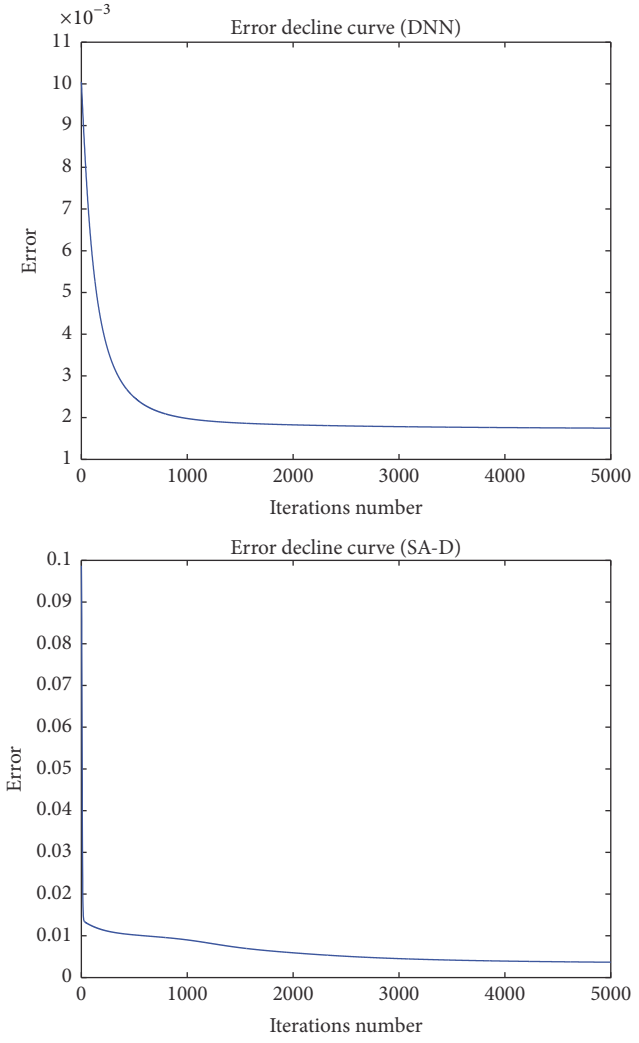


Figure 5: Error decline curve of the DNN model and the SA-D model.

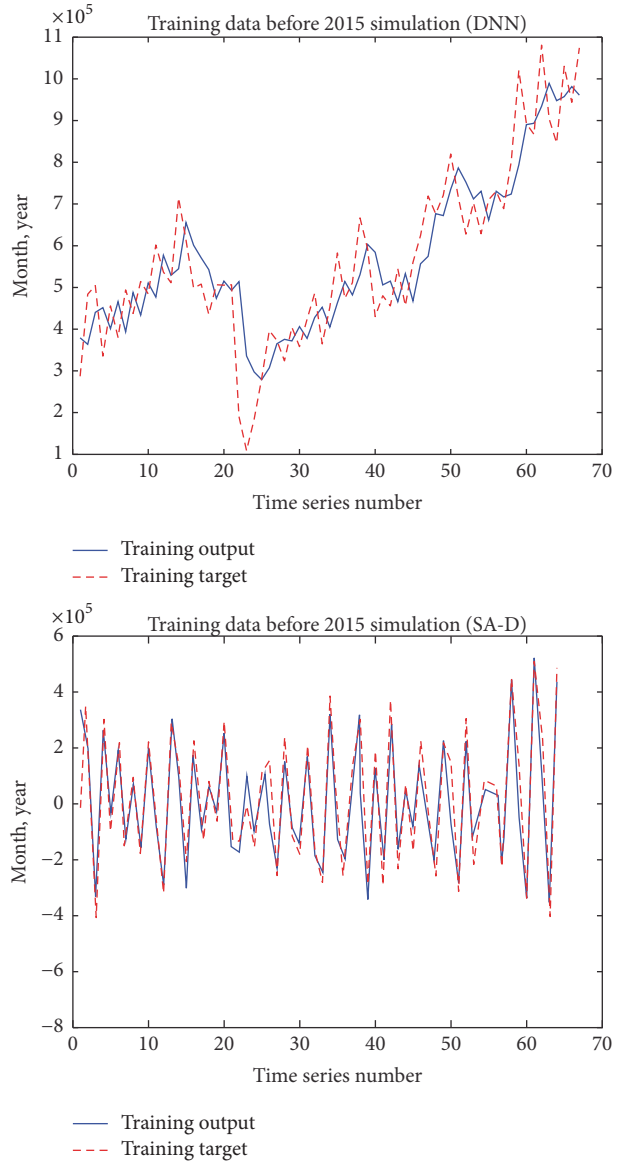


Figure 6: Training data before 2015 simulation of the DNN model and the SA-D model.

author scaled the data within the range of (0, 1) through the following formula:

$$\frac{x_t - x_{\min}}{x_{\max} - x_{\min}} \times 0.7 + 0.15. \tag{14}$$

So we use the data with the same preset as the author did and without the data preset separately and get our experimental results.

Before comparing with the models, we summarize the experimental results based on the orthogonal array, factor assignment, and statistical tests as Table 4 shows. Here the MSD values are calculated by $\overline{x} \pm s$, where $\overline{x}$ means the mean of the results over 20 runs and s means the standard deviation. It can verify whether the data is closer to reality or not. And p value can determine whether the residual is white noise sequence or not after the statistical test by using QLB statistic. Finally, we choose the result of number 7 to do the comparison.

As Table 5 shows, our model had much better results than other authors' models. But we have to say that the data preset by (14) made the results better and reduced the running time of program.

4. Conclusions

In this study, we proposed a new model, the SA-D model, which mixed the SARIMA model and the DNN model together. First, we used the data collected from Japan Tourism Agency Ministry of Land, Infrastructure, Transport and Tourism and Japan National Tourism Organization to compare the SA-D model and DNN model; the results showed that the SA-D model performed much better in fitting and forecasting the time series data. Then we verified the effectiveness of our model by comparing with other authors' models and got the expected result.

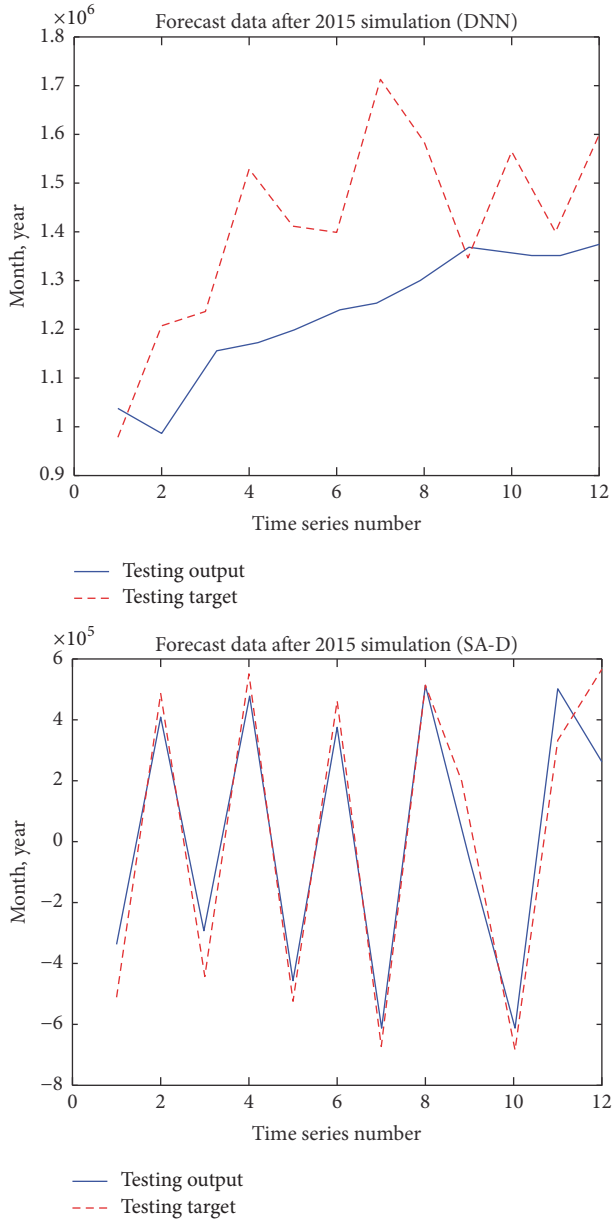


FIGURE 7: Forecast data after 2015 simulation of the DNN model and the SA-D model.

The contributions of this study lie in two aspects. Our study is based on neuron model with dendritic nonlinearity model and it theoretically strengthens the assumption that a neural network model performs better than linear models when forecasting nonlinear variables.

This study which mixed the linear model and the nonlinear model together opens the door for further combination models with different methods and models.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This research was partially supported by JSPS KAKENHI (Grant no. 15K00332).

References

[1] H. Song and G. Li, "Tourism demand modelling and forecasting—a review of recent research," *Tourism Management*, vol. 29, no. 2, pp. 203–220, 2008.

[2] C. J. S. C. Burger, M. Dohnal, M. Kathrada, and R. Law, "A practitioners guide to time-series methods for tourism demand forecasting—a case study of Durban, South Africa," *Tourism Management*, vol. 22, no. 4, pp. 403–409, 2001.

[3] R. Law, "Back-propagation learning in improving the accuracy of neural network-based tourism demand forecasting," *Tourism Management*, vol. 21, no. 4, pp. 331–340, 2000.

[4] R. Law, "Demand for hotel spending by visitors to Hong Kong: a study of various forecasting techniques," *Journal of Hospitality & Leisure Marketing*, vol. 6, no. 4, pp. 17–29, 1999.

[5] F.-L. Chu, "Forecasting tourism demand: a cubic polynomial approach," *Tourism Management*, vol. 25, no. 2, pp. 209–218, 2004.

[6] H. Qu and H. Q. Zhang, "Projecting International Tourist Arrivals in East Asia and the Pacific to the Year 2005," *Journal of Travel Research*, vol. 35, no. 1, pp. 27–34, 1996.

[7] V. Cho, "A comparison of three different approaches to tourist arrival forecasting," *Tourism Management*, vol. 24, no. 3, pp. 323–330, 2003.

[8] P.-F. Pai and W.-C. Hong, "An improved neural network model in forecasting arrivals," *Annals of Tourism Research*, vol. 32, no. 4, pp. 1138–1141, 2005.

[9] S. F. Crone and P. C. Graffeille, "An evaluation framework for publications on artificial neural networks in sales forecasting," in *Proceedings of the International Conference on Artificial Intelligence (IC-AI '04)*, pp. 221–227, Las Vegas, Nev, USA, June 2004.

[10] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: the state of the heart," *The International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[11] M. Nelson, T. Hill, W. Remus, and M. O'Connor, "Time series forecasting using neural networks: should the data be deseasonalized first?" *Journal of Forecasting*, vol. 18, no. 5, pp. 359–367, 1999.

[12] G. P. Zhang and D. M. Kline, "Quarterly time-series forecasting with neural networks," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1800–1814, 2007.

[13] J. Teixeira and P. Fernandes, "Tourism time series forecast with artificial neural networks," *Tékhne*, vol. 12, no. 1-2, pp. 26–36, 2014.

[14] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, John Wiley & Sons, 2015.

[15] J. Shahrabi, E. Hadavandi, and S. Asadi, "Developing a hybrid intelligent model for forecasting problems: case study of tourism demand time series," *Knowledge-Based Systems*, vol. 43, pp. 112–122, 2013.

[16] X. Li, B. Pan, R. Law, and X. Huang, "Forecasting tourism demand with composite search index," *Tourism Management*, vol. 59, pp. 57–66, 2017.

[17] Y. Kawakubo and T. Kubokawa, "Modified conditional AIC in linear mixed models," *Journal of Multivariate Analysis*, vol. 129, pp. 44–56, 2014.

[18] C.-F. Chen, M.-C. Lai, and C.-C. Yeh, "Forecasting tourism demand based on empirical mode decomposition and neural network," *Knowledge-Based Systems*, vol. 26, pp. 281–287, 2012.

[19] O. Claveria and S. Torra, "Forecasting tourism demand to Catalonia: neural networks vs. time series models," *Economic Modelling*, vol. 36, pp. 220–228, 2014.

[20] N. Davies, J. Pemberton, and J. D. Petruccelli, "An automatic procedure for identification, estimation and forecasting univariate self exiting threshold autoregressive models," *The Statistician*, vol. 37, no. 2, pp. 199–204, 1988.

[21] H. Constantino, P. Fernandes, and J. Teixeira, "Tourism demand modelling and forecasting with artificial neural network models: the Mozambique case study," *Tekhen*, vol. 14, no. 2, pp. 113–124, 2016.

[22] C. J. Lin, H. F. Chen, and T. S. Lee, "Forecasting tourism demand using time series, artificial neural networks and multivariate adaptive regression splines: evidence from taiwan," *International Journal of Business Administration*, vol. 2, no. 2, article no. 14, 2011.

[23] J. M. Bates and C. W. J. Granger, "The combination of forecasts," *Operational Research Quarterly*, vol. 20, no. 4, pp. 451–468, 1969.

[24] K.-Y. Chen, "Combining linear and nonlinear model in forecasting tourism demand," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10368–10376, 2011.

[25] S. Shen, G. Li, and H. Song, "Combination forecasts of International tourism demand," *Annals of Tourism Research*, vol. 38, no. 1, pp. 72–89, 2011.

[26] W. Yan, "Toward automatic time-series forecasting using neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1028–1039, 2012.

[27] K. Lin, P. Pai, Y. Lu, and P. Chang, "Revenue forecasting using a least-squares support vector regression model in a fuzzy environment," *Information Sciences*, vol. 220, pp. 196–209, 2013.

[28] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Information Processing-Letters and Reviews*, vol. 11, no. 10, pp. 203–224, 2007.

[29] P.-F. Pai, K.-C. Hung, and K.-P. Lin, "Tourism demand forecasting using novel hybrid system," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3691–3702, 2014.

[30] H. Yang, L. Lu, and W. Zhou, "A novel optimization sizing model for hybrid solar-wind power generation system," *Solar Energy*, vol. 81, no. 1, pp. 76–84, 2007.

[31] K. Bogner, F. Pappenberger, and H. L. Cloke, "Technical note: the normal quantile transformation and its application in a flood forecasting system," *Hydrology and Earth System Sciences*, vol. 16, no. 4, pp. 1085–1094, 2012.

[32] E. B. Dagum, *The X-II-ARIMA Seasonal Adjustment Method*, Seasonal Adjustment and Time Series Staff, Statistics Canada, 1980.

[33] S. Hylleberg, R. F. Engle, C. W. Granger, and B. S. Yoo, "Seasonal integration and cointegration," *Journal of Econometrics*, vol. 44, no. 1-2, pp. 215–238, 1990.

[34] J. Cunado, L. A. Gil-Alana, and F. P. de Gracia, "Is the US fiscal deficit sustainable?: a fractionally integrated approach," *Journal of Economics and Business*, vol. 56, no. 6, pp. 501–526, 2004.

[35] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[36] M. Minsky and S. Papert, *Perceptrons-Expanded Edition: An Introduction to Computational Geometry*, 1987.

[37] http://www.mlit.go.jp/kankocho/siryou/toukei/shukuhakutoukei.html.

[38] http://www.jnto.go.jp/jpn/statistics/data_info_listing/index.html.

[39] http://admin.taiwan.net.tw/statistics/year.aspx?no=134.

Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2016, Article ID 9328062, 12 pages
http://dx.doi.org/10.1155/2016/9328062

Research Article

Main Trend Extraction Based on Irregular Sampling Estimation and Its Application in Storage Volume of Internet Data Center

Beibei Miao,[1] Chao Dou,[2] and Xuebo Jin[3]

[1]*Baidu, Inc., Beijing 100085, China*
[2]*Center of Quality Engineering, AVIC China Aero-Polytechnology Establishment, Beijing 100028, China*
[3]*School of Computer and Information Engineering, Beijing Technology and Business University, Beijing 100048, China*

Correspondence should be addressed to Xuebo Jin; jinxuebo@btbu.edu.cn

Received 31 August 2016; Accepted 15 November 2016

Academic Editor: Francisco Martínez-Álvarez

The storage volume of internet data center is one of the classical time series. It is very valuable to predict the storage volume of a data center for the business value. However, the storage volume series from a data center is always "dirty," which contains the noise, missing data, and outliers, so it is necessary to extract the main trend of storage volume series for the future prediction processing. In this paper, we propose an irregular sampling estimation method to extract the main trend of the time series, in which the Kalman filter is used to remove the "dirty" data; then the cubic spline interpolation and average method are used to reconstruct the main trend. The developed method is applied in the storage volume series of internet data center. The experiment results show that the developed method can estimate the main trend of storage volume series accurately and make great contribution to predict the future volume value.

1. Introduction

In general, the internet data center stores a huge scale of data, for example, the data of search engines and the data of E-commerce services. It is necessary to predict the future storage volume value of a data center, because it is helpful for operation engineers to make purchasing devices plan for data center. As the devices always have limited warranty time and the maintenance cost is large, it is better to buy devices when needed, because this will cost less. Meanwhile, considering the devices transportation delay, engineers have to buy devices in advance to offer enough storage space for the increasing data. Thus, making an accurate prediction result for storage volume series is very important. However, the collected storage data often contains white noise, outliers, and missing data (replaced by 0) besides the real main trend. Such "dirty" data adds great difficulties to make accurate prediction [1]. In fact, dirty data will inflict daunting waste, which had cost US businesses 600 billion dollars each year [2]. Thus, it is very important to clean the dirty data and extract its main trend.

Figure 1 shows a storage volume series of a data center, which comes from an internet company (the values have been desensitized). Figure 1 labels the main features of the dirty time series data. Feature "a" in Figure 1 represents the missing data, which may be caused by machines' sudden halt or data collection subsystem. Feature "b" represents the outliers, which may be the wrong record data caused by "bugs" in the record programme. Feature "c" represents the noise which is the change of the usage and the enlarged subplot shows the detail fluctuation of the noise in the main trend. Figure 2 gives the main trend of Figure 1, which is the trend we expect to be used for the further prediction of the storage volume. Therefore, our assignment is to extract the main trend in Figure 2 from the "dirty" series in Figure 1.

In fact, there are many techniques of extracting the main trend. For example, [3] employs an STL method to decompose the main trend based on Loess, while [4] applies a piecewise approximation method to extract the underlying long-term trend. Reference [5] explores a quantile regression method to extract the main trend. Though these methods can achieve the main trend, they still have some

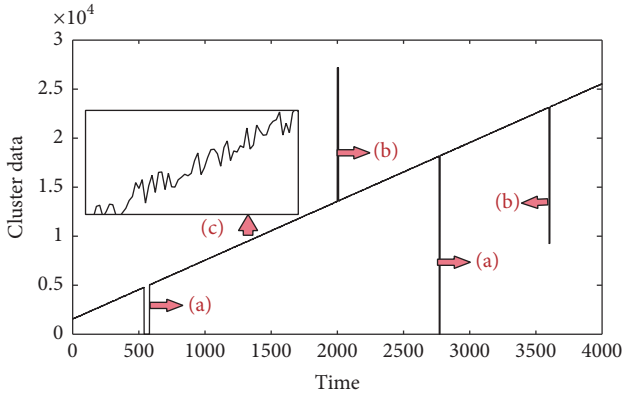


Figure 1: The dirty storage volume time series.

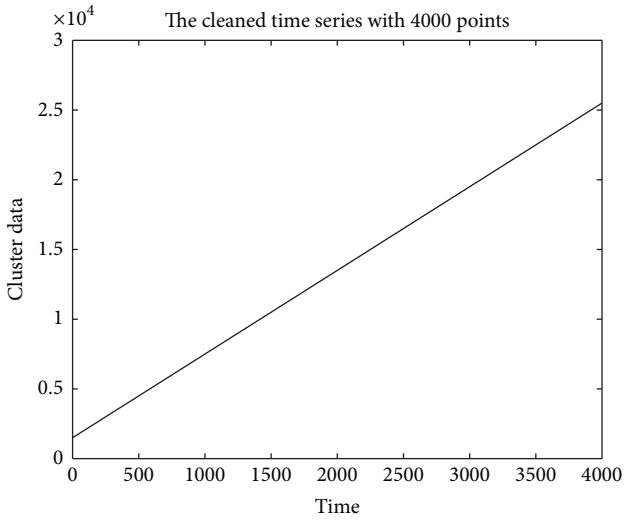


Figure 2: The main trend of storage volume time series.

shortcomings. There are too many parameters and decomposition steps in STL method. Therefore, it is difficult to initialize the parameters and the computation costs are very large. The piecewise approximation method [4] can ignore the distortion of outliers and have good extraction result for the long-term trend. But it cannot give specific data of the main trend because it estimates the long-term trend by median values in certain window. As for the regression method [5], it cannot obtain the main trend directly and has to go further by using interpolation methods to get the main trend, for example, the linear or B-spline interpolation method. This may cause overfitting when the time series is short or achieve poor extraction result because the large blocks of outliers would distort the spline and yield a large number of incorrect interpolation results.

Unlike the mentioned approach, this paper offers an estimation method to achieve the main trend. To the best of our knowledge, the Kalman filter can estimate the dynamic features of time series, as well as its main trend. What is more, researchers have received a lot of results about Kalman filter [6–9], by which the extracted high dimension characters can also be used to predict the tendency of series [10–12] or make further decision through other methods like the intelligence method [13, 14]. For example, [6] gives a real-time correction method to estimate data online and forecast the water stage, which is effective and fast. Reference [15] considers a prediction problem for multivariate time series and proposed an online model by echo state network (ESN) based on square root cubature Kalman filter.

However, the above Kalman filter methods are based on regular sampling estimation, which means that all of the measurements will be used. If we employ these methods on the storage volume time series directly, the outliers and missing data will greatly degrade the estimation performance. For our "dirty" time series case, we prefer to use some of the measurement data so as to discard the outliers and the missing data. Thus, the "irregular" estimation method will be helpful.

As for the irregular estimation methods, researchers have got lots of results; for example, [16] develops a method to handle time-varying and uncertain delay problem. Based on the modification of the Kalman filter and the negative-time measurement update strategy, [17] used the full augmented order models to handle the long delay problem for the networked control systems and scarce measurement problem for out-of-sequence measurements. Reference [18] discussed the irregular estimation method in detail and transformed the irregular sampling time to a time-varying parameter by calculating the matrix exponential with inverse Laplace transform method. Based on the statistic relation between autocorrelation function and the covariance of Markov random processing, [19] develops a model to track video signal by Kalman filter, which can adaptively adjust the model parameters while tracking and obtain good estimation performance even at a very low irregular rate. These research results show us that the Kalman filter based on irregular sampling estimation can use the part of the measurement data and we hope it will help us to cut down the effect of the outliers and the missing data.

The result of irregular sampling estimation is the compressed data series with unknown amount of series and sampling intervals, which will confuse the following prediction of the future main trend. So we apply cubic spline method to interpolate the storage volume time series and reconstruct the whole time series with the same number of the former series. Also, we note that the irregular estimation may select some "dirty" measurements. Such data may distort the estimation, which will cause the quality of the main trend with poor performance. To get robust main trend, we estimate and reconstruct the time series several times instead of only once. Then, an average method is used to achieve the main trend. Part of the developed method has been mentioned in the conference paper [20] with 4 pages. By comparison, this manuscript gives the details of the main trend extraction method and discusses the experiments more comprehensively.

This paper is organized as follows. Section 2 details the main trend estimation algorithm, including the irregular sampling estimation based on Kalman filter, the cubic spline interpolation reconstruction method, and the averaging method. Section 3 gives the experiment results. The

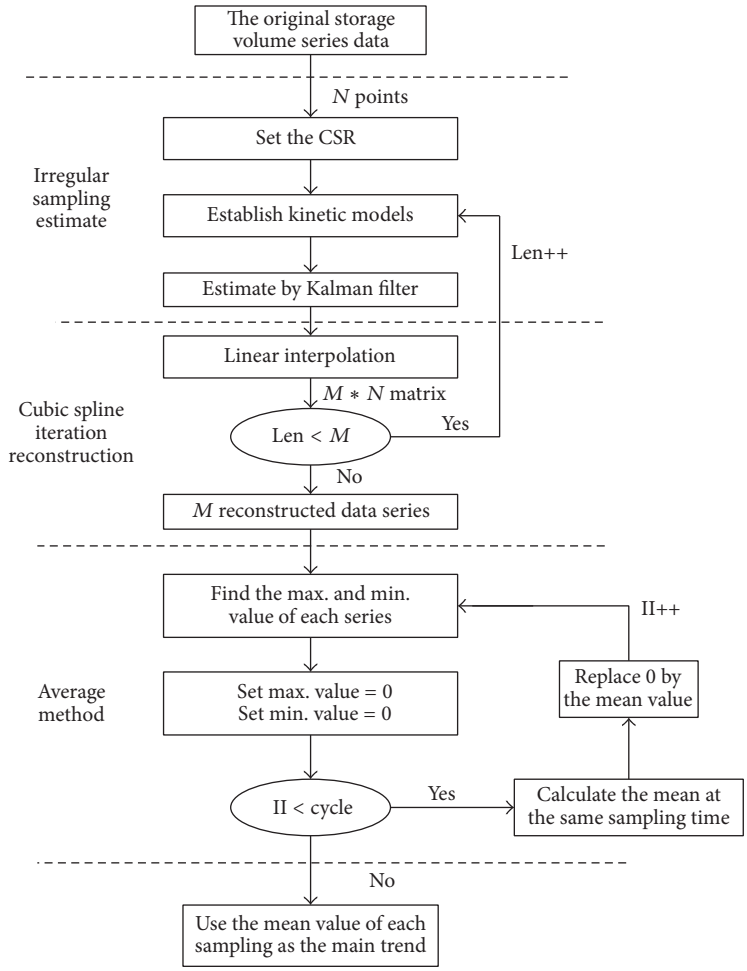


Figure 3: The flow chart of main trend estimation algorithm.

developed method is applied for the storage volume of the internet data center; meanwhile the results of some reference methods, such as the Piecewise Median Underlying (PMU) method, the local regression smoothing (Loess) method, and the Moving Average (MA) smoothing method, are also given. Conclusions and future works are presented in Section 4.

2. Main Trend Estimation Algorithm

Before introducing the algorithm, we have to give the definition of Compression Sampling Rate (CSR) to value the degree of compression:

$$\mathrm{CSR} = \frac{N_s}{N}, \tag{1}$$

where N_s is the number of selected data and N is the total number of original storage volume time series. We can note that lower CSR means higher compression degree. By using the irregular sampling method, we can retain the main information of original storage volume time series under a low CSR value.

Figure 3 gives the flow chart of the main trend estimation algorithm. The main trend estimation algorithm contains three parts: the compressed estimation step by Kalman filter, the cubic spline interpolation reconstruction method, and the average method. By using the irregular sampling method, we can compress the original dirty data series and try to discard outliers and missing data. The interpolation method is helpful to achieve reconstructed time series with the same length as the original storage volume time series. To avoid the influence of some "lucky" selected outliers, we estimate the time series M times and obtain M time series with the same CSR value. The average method is used to obtain final extracted main trend.

2.1. The Irregular Sampling Compression and Kalman Filter Estimation Method. Our research is designed for discrete time series, and the following is the Kalman filter equations:

$$\begin{aligned}
&\text{Initialization: } k = 0\\
&\qquad \widehat{x}(0 \mid 0) = V_0,\\
&\qquad P(0 \mid 0) = P_0,\\
&\qquad \alpha(0) = \alpha_0,\\
&\qquad \delta_w^2(0) = \delta_{w0}^2,\\
&\qquad \lambda = \lambda_0\\
&\text{Recursion: } k \coloneqq k + \delta
\end{aligned} \tag{2}$$

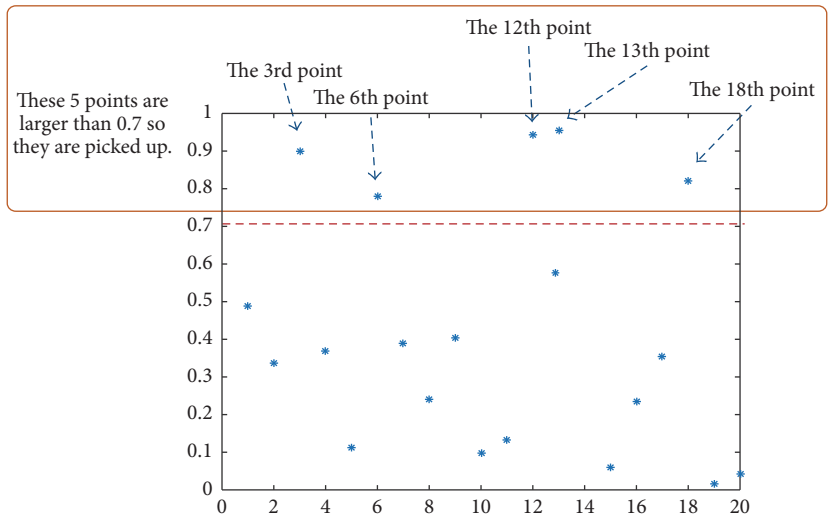


Figure 4: The relation of A and δ.

Here δ represents the interval between two pieces of input data. Usually $\delta > 1$, so we can compress the series and use the Kalman filter to extract the basic trend. The method about how to select δ is discussed in Note 1.

(a) Prediction

$$\begin{aligned} \widehat{x}(k+\delta \mid k) &= \Phi(k+\delta, k)\,\widehat{x}(k \mid k), \\ P(k+\delta \mid k) &= \Phi(k+\delta, k)\,P(k \mid k)\,\Phi^{T}(k+\delta, k) \\ &\quad + Q(k). \end{aligned} \tag{3}$$

(b) Update

$$\begin{aligned} &\widehat{x}(k+\delta \mid k+\delta) \\ &\quad = \widehat{x}(k+\delta \mid k) \\ &\qquad + K(k+\delta)\left[y(K+\delta) - H\widehat{x}(k+\delta \mid k)\right], \\ &K(k+\delta) \\ &\quad = P(k+\delta \mid k)\,H^{T}\left[HP(k+\delta \mid k)\,H^{T} + R\right]^{T}, \\ &P(k+\delta \mid k+\delta) = [I - K(k+\delta)\,H]\,P(k+\delta \mid k), \end{aligned} \tag{4}$$

where the observation vector $y(k)$ is the storage capacity time series with N points and $k = 1, 2, 3, \ldots, N$; $\widehat{x}(k+\delta, k+\delta)$ is the estimated current time series data. I is a unit matrix and R is the covariance of the series noise. $\phi(k+\delta, k)$ is the process transformation matrix, $Q(k)$ is the process noise covariance matrix, and H is the observation transformation matrix.

From (2)–(4), we can see that the system parameters of $\phi(k+\delta, k)$, $Q(k)$, H, and R are important to the Kalman filter, in which $\phi(k+\delta, k)$ and $Q(k)$ are called the process models and H and R are called measurement models. To capture the dynamic characters, researchers have given many models for estimation. Notes 2 and 3 will give more specific information about dynamic models.

Note 1 (the selection about δ). The initial δ is set as 0 and assume Sa is a uniform distribution random vector with N dimension, where $\mathrm{Sa}(i) \in (0, 1)$, $i = 1, 2, \ldots, N$. Then we introduce a constant named A, where $A \in (0, 1)$ corresponding to CSR. For example, A is 0.7 means the CSR value is $(1-0.7) * 100\% = 30\%$.

We obtain the interval value by comparing $\mathrm{Sa}(i)$ and A. Only the ith data with $\mathrm{Sa}(i) > A$ is picked up and δ is calculated by two adjacent picked pieces of data. We give an example for the relation of A and δ and how to calculate δ is shown in Figure 4. In Figure 4, A is set as 0.7, and 20 points $\mathrm{Sa}(i) \in (0, 1)$, $i = 1, 2, \ldots, 20$, are created by a uniform distribution random vector. We can see that only the 3rd, 6th, 12th, 13th, and 18th are larger than 0.7; therefore we can get $\delta = 3, 3, 6, 1$, and 5. The flow chart is shown in Figure 5.

Note 2 (the selection of process models). The process model described the changing relations about the main trend. Some inertia model had been developed by the researchers, such as constant-velocity (CV) model, constant-acceleration (CA) model, Singer model, the "current" model, and the adaptive model. CV [21] assumes that the acceleration is a Wiener process or, more generally and precisely, the acceleration is a process with independent increments, which is not necessarily a Wiener process. It is simply referred to as CA or more precisely "nearly-constant-acceleration model" [22]. The Singer model in [23] assumes the acceleration as a first-order semi-Markov process with zero mean, which in essence is a priori model since it does not use online information about the target maneuver, and it can be made adaptive through some parameters.

An acceleration model, called the "current" model [23], is in essence a Singer model with an adaptive mean, that is, a Singer model modified to have a nonzero mean of the acceleration. The "current" model can use the online information and replace the a priori (unconditional) probability density

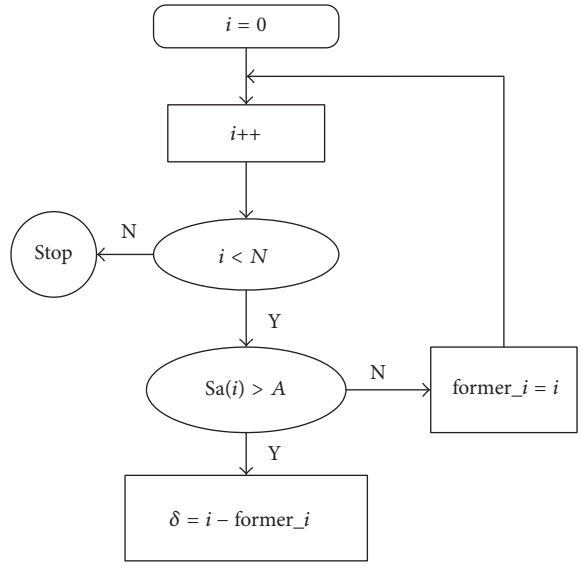


Figure 5: The flow chart of obtaining δ.

of acceleration in Singer model by a conditional density, that is, Rayleigh density. Clearly, this conditional density carries more accurate information than a priori density.

The above models all need prior hypothesis. Based on the statistical relation between the autocorrelation function and the covariance of Markov random processing, [24] develops a model which can adaptively adjust system parameter to dynamics characters of time series online, but this process is complex to compute. In practice, we should choose the appropriate system parameters to suit the data dynamics characters. In our experiments, we use several models for the developed algorithm and discuss the estimation performance.

Note 3 (the selection of measurement models). We give the measurement model for the main trend of the series data as

$$y(k) = H(k)\, x(k) + v(k), \tag{5}$$

where $x(k)$ is the main trend to be extracted. The extraction matrix $H(k)$ can be set as $H(k) = [1 \;\; 0 \;\; 0]$ if the state is defined as a three-dimensional vector. The covariance of the extraction noise $v(k)$, which we denote as R, can be decided by the difference between the main trend and the original data as

$$R = \frac{\sum_{k=k_0}^{N_t+k_0} [y(k) - x(k)]^2}{N_t}, \tag{6}$$

where $y(k)$ is the original data, $x(k)$ is the expected main trend, k_0 is the positive constant, $k_0 \in (0, N)$, and N_t is the number of the series to be estimated; $N_t \leq N - k_0$.

To better illustrate the influence of R, we choose a segment of the dirty data in Figure 1 with N_t samples to calculate R (named as R_{small}), where $N_t \leq N$ (in the first subplot in Figure 6, the red part is the data selected to be estimated). We also choose the whole dirty time series to calculate R (named as R_{large}), shown in the third subplot in Figure 6, with the "red" part showing the data for estimating. Because of the influence of outliers, R_{small} is smaller than R_{large}. In the second and the fourth subplots in Figure 6, the red lines show the estimation results by R_{small} and R_{large}, respectively. To be more clear, Figure 7 gives an enlarged subplot about the estimated main trend with R_{small} and R_{large} from 2600th to 3000th samples. From Figure 7, we can conclude that R_{small} can get the less estimated point with errors than R_{large} and such peak value can be removed by the following average step. Therefore, in practice, N_t is chosen as $N/3$ so as to remove some of the "dirty" data and receive the better results.

As to the expected main trend $x(k)$, we should choose it based on the practical applications. In general, we know what the main trend should be in practice, as we have mentioned that the "dirty" data of a data center shown in Figure 1 should have the main trend as in Figure 2, where the data engineers give this suggestion based on the knowledge about the data center.

2.2. The Cubic Spline Interpolation Reconstruction Method. By the compression and estimation method in Section 2.1, we can obtain the useful information of main trend based on part of the original storage volume series. But the compressed series cannot be used for the further prediction directly because its sampling is irregular. Thanks to δ parameter, the specific intervals between two pieces of input data are retained and can be used to reconstruct the whole time series.

The polynomial of cubic spline interpolation $S(x)$ is a piecewise function of $x_0, x_1, \ldots, x_{n-1}, x_n$ data.Each cubic polynomial value determines the parameters at each mini

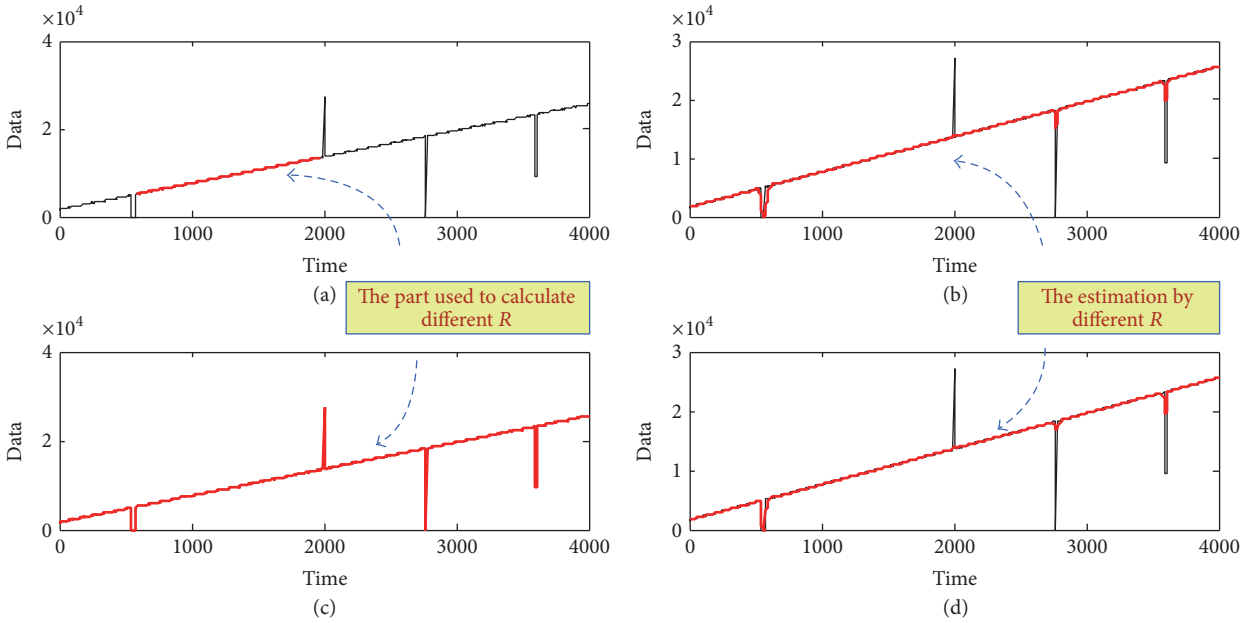


Figure 6: The main trend of storage volume time series.

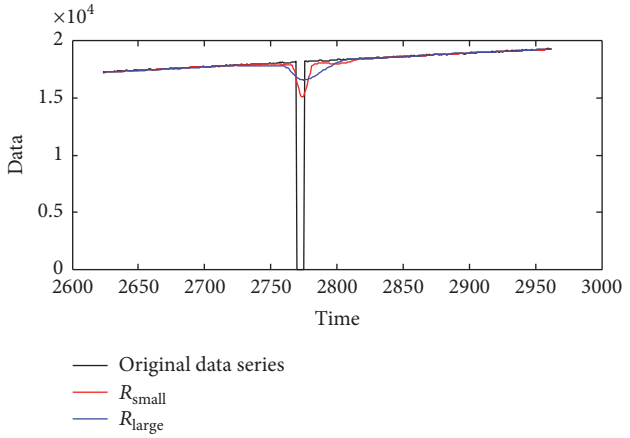


Figure 7: The comparison of different R from 2600th to 3000th samples.

zone $[x_{i-1}, x_i]$ and the node x_i satisfies $a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$. We have

$$S(x) = \frac{1}{6h_i}\left[(x_i - x)^3 \lambda_{i-1} + (x - x_{i-1})^3 \lambda_i\right] + \left(y_{i-1} - \frac{h_i^2}{6}\lambda_{i-1}\right)\frac{x_i - x}{h_i} + \left(y_i - \frac{h_i^2}{6}\lambda_i\right)\frac{x - x_{i-1}}{h_i}, \quad (7)$$

where $x \in [x_{i-1}, x_i]$, $i = 1, 2, \ldots, N \cdot CSR$.

Define

$$\mu_i = \frac{h_i}{h_i + h_{i+1}},$$
$$\lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}} = 1 - \mu_i,$$
$$d_i = \frac{6}{h_i + h_{i+1}}\left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i+1}}{h_i}\right) = 6f\left(x_{i-1}, x_i, x_{i+1}\right). \quad (8)$$

Equation (8) satisfies the following $n - 1$ equations:

$$\mu_i \lambda_{i-1} + 2\lambda_i + \lambda_i \lambda_{i+1} = d_i, \quad i = 1, 2, \ldots, n-1. \quad (9)$$

Equation (7) has $n + 1$ unknown quantity; if we set $\lambda_0 = \lambda_n = 0$, the value of $\lambda_i\,(1 \le i \le n - 1)$ and trend(x) can be obtained. In our reconstruction, we get cubic spline interpolation values in each $[x_{i-1}, x_i]$, where x_{i-1} and x_i are the adjacent two sampled input storage time series.

2.3. The Averaging Method. As mentioned previously, we select the part of the original dirty storage volume series to estimate. Under the assumption that the outliers are little, we can guarantee that most of the selected data can contribute to the main trend data and most of useful information can be extracted when outliers can be discarded and missing data can be handled. However, some of the outliers or the missing data of the real storage volume time series may be "lucky" enough and be selected. In fact, the "lucky" selected data will

distort the main trend in the interpolation reconstruction step and decrease the extraction performance. Decreasing the CSR value is surely a way to reduce the possibility of "lucky" one, but too low CSR value may cause some useful information to be lost. So we further use the average method to achieve the main trend after reconstruction step.

The averaging method can be detailed as follows. At first, we use the irregular estimation method M times to get M main trend series. As to the reconstruction results, the selected "lucky" outliers or missing data are always the max. or min. value among M reconstructed series with the same column. Thus, the average method is used to calculate the mean value by discarding the max. and min. value. Repeat several cycle times until the max. and min. are similar to the mean. For simplicity, instead of numerical comparisons, we use the number of cycles to control the end of the cycle. The detailed algorithm is as follows:

(1) Calculate the max. and min. values of each column in $M \times N$ reconstructed time series matrix, where N is the number of the time series.

(2) Set the max. and min. value of the row by zeros.

(3) Calculate the mean value of each column in the replaced $M \times (N - 2)$ matrix.

(4) Use the calculated mean value replacing the "zeros."

(5) Repeat the above steps II cycle times, where II is a positive constant.

3. Experiments and Discussion

In this section, two parameters are used to measure the performance of the developed main trend estimation method: Covariance (Cov) is introduced to evaluate the quality of the estimated main trend and Time of Programming (TP) is applied to measure the calculation cost of different dynamic models:

$$\widehat{y}(k) - y(k) = e(k),$$
$$\text{Cov} = \frac{\sum_{k=1}^{N} e(k)}{N}, \tag{10}$$

where $y(k)$ represents the expected main trend, while $\widehat{y}(k)$ represents the estimated main trend by developed algorithm. N is the total number of the original storage volume time series.

Section 3.1 gives several reference methods, such as the Piecewise Median Underlying method, the Local regression smoothing (Loess) method, and the Moving Average (MA) smoothing method, whose results have been discussed. Section 3.2 discussed the developed method of performance based on different models, including CV, CA, Singer, current model, and adaptive model.

3.1. Several Reference Approaches for Extracting Main Trend. In this section, the Piecewise Median Underlying (PMU) method [5], the Local regression (Loess) smoothing method [3], and the Moving Average (MA) smoothing method [4] are used for extracting the main trend of the storage volume. Table 1 gives the Cov result based on the three methods under different window size and Figure 8 shows the extraction result of the three methods.

Table 1: The Cov of main trend extraction result based on PMU, Loess, and MA.

Cov	PMU	Loess	MA
10	704090	1307400	915220
50	323010	672790	309860
100	**32780**	427980	167570
150	71339	314470	114510
200	124360	**244900**	**85857**
250	174170	206780	71517

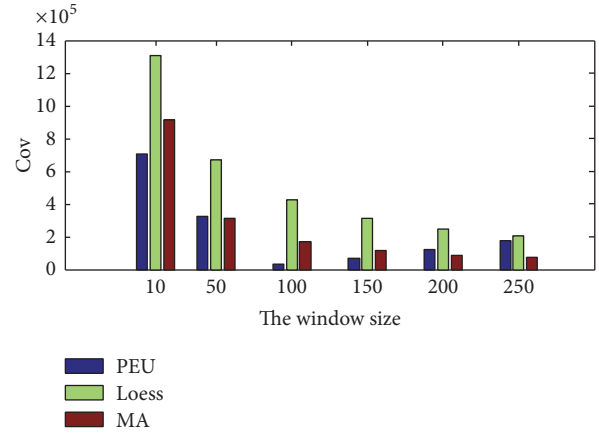


Figure 8: The extraction result by PMU, Loess, and MA.

From Table 1 and Figure 8, we can see that the window size can influence greatly the result of main trend extraction by PMU, Loess, and MA methods. For PMU method, there exists a tradeoff between the window size and the extraction performance. In fact, small window size means a better approximation of the expected main trend, while large window size means a more poor approximation result. So the Cov value should be larger along with larger window size. But the existence of missing data and outliers make Cov value very large when the window size is small. That is the reason why a tradeoff (323010) exists in PMU extraction method. For Loess and MA methods, it is easy to find that larger window size means better extraction result. But when the window size is too large, the time delay problem is very obvious. Thus we have to find a tradeoff for these two methods too, for example, the window size with 200. Figure 9 gives the main trend extraction result of the three methods under their tradeoff. And Figure 10 gives the detailed information of the four windows in Figure 9.

From Figures 9 and 10, we can see that the Loess and MA methods can remove white noise very well, but they cannot deal with the outliers and missing data as well as PMU. Though the PMU has a low Cov value, it depends on the

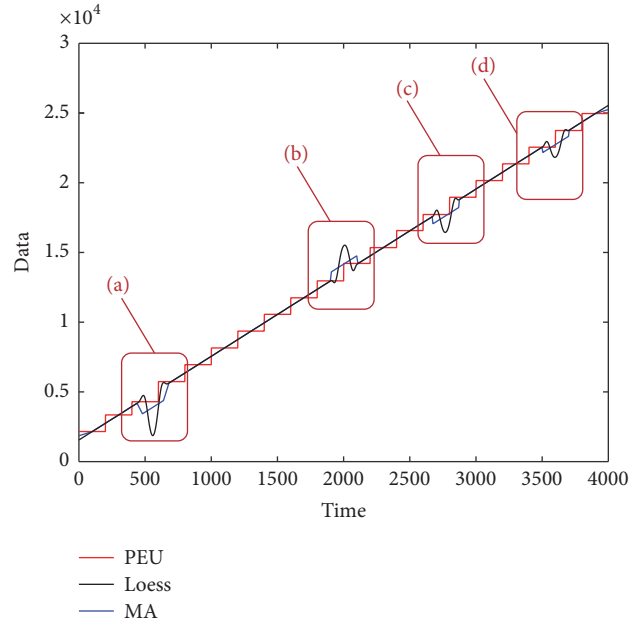


Figure 9: The extracted main trend by PEU, Loess, and MA methods.

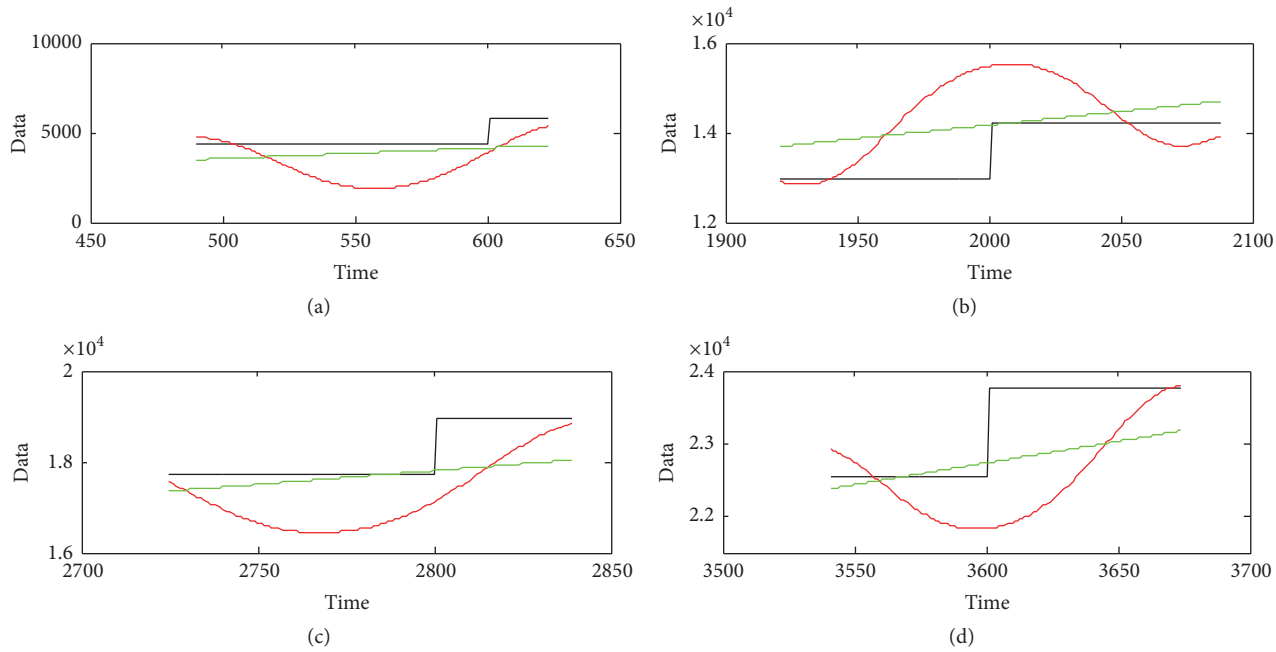


Figure 10: The detailed extraction result by PMU, Loess, and MA methods.

values of the calculating window. What is more, there exists a long lag effect for its trend if the last few points are less than the window.

3.2. Extract Main Trend Based on Different System Models Based on Different System Models. In this section, we will use different system model for the developed extracting method to compare with PEU, Loess, and MA methods; meanwhile we will analyze the performance of developed method based on different system models. Table 2 gives the extraction Cov with the different system dynamic models and CSR. Figures 11 and 12, respectively, give the Cov and TP values about different dynamic models and different CSR values. Comparing Tables 1 and 2, we can conclude that our developed method is better than PMU, Loess, and Moving Average smoothing methods, because the Cov value is smaller, especially when the CSR value is small. We admit that the computation cost of the developed method is larger

Table 2: The Cov/TP main trend estimation result.

CSR	CV	CA	Singer	Current	Jerk	Adaptive
1%	46214/246.60	21834/243.11	**11278/244.60**	47011/245.23	15680/248.69	14839/237.18
3%	119460/285.06	163920/250.55	107880/252.01	137070/251.68	168470/252.27	140280/242.28
5%	243250/355.91	247660/261.56	287760/261.34	233420/261.83	192730/264.86	288430/250.64
7%	382880/483.28	488200/268.93	360140/269.89	436360/272.67	365600/270.34	348420/260.81
9%	610100/588.35	433290/279.60	570830/281.79	507100/287.21	492320/280.91	534560/273.98

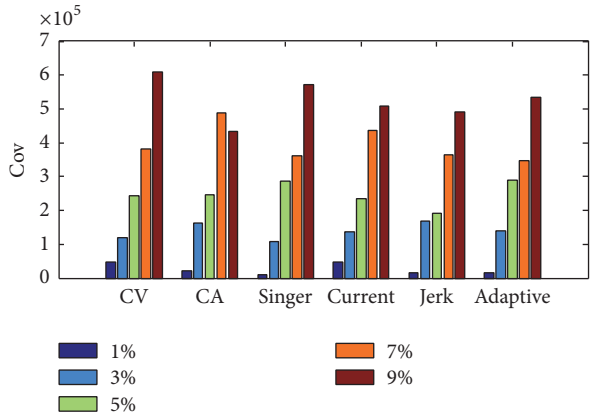


Figure 11: The estimation Cov result of different models.

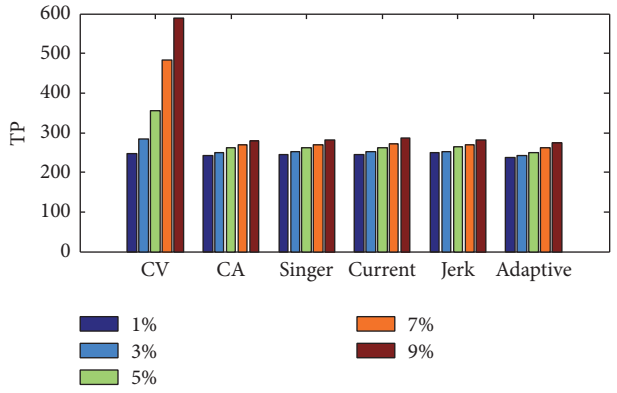


Figure 12: The estimation TP result of different models.

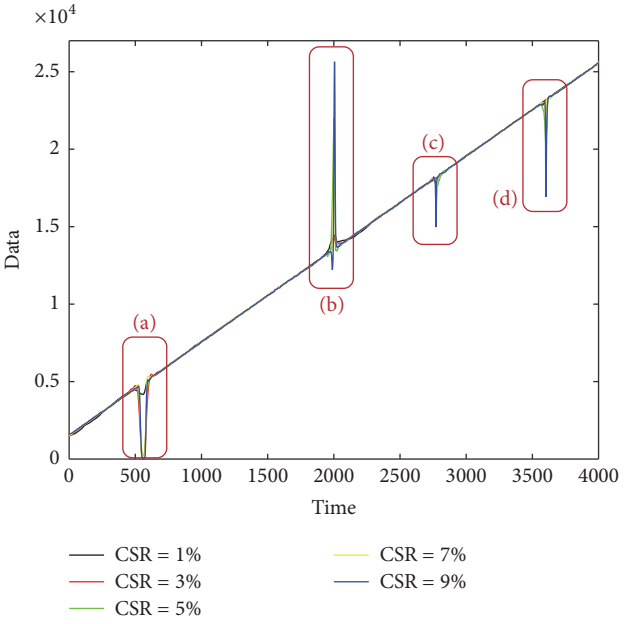


Figure 13: The estimation result by Singer model under different CSR.

than other three methods, but the computing time cost will still be within the limitation of practical application.

From Table 2, we can see that smaller CSR means better main trend estimation result when the dynamic model is determined. It is about half time down of Cov values when the CSR is 2% smaller. The reason is that small CSR can reduce the probability of selecting "lucky" dirty data.

Figure 13 gives the estimated trends with the Singer model for different CSR values, and Figure 14 gives the enlarged details of the windows in Figure 13. From the figures, we can note that lower CSR value is helpful to achieve better main trend.

4. Conclusions

The estimation result may be different by different dynamic models with the set CSR value. Figure 15 gives the estimation result about different models when CSR is 1%. And Figure 16

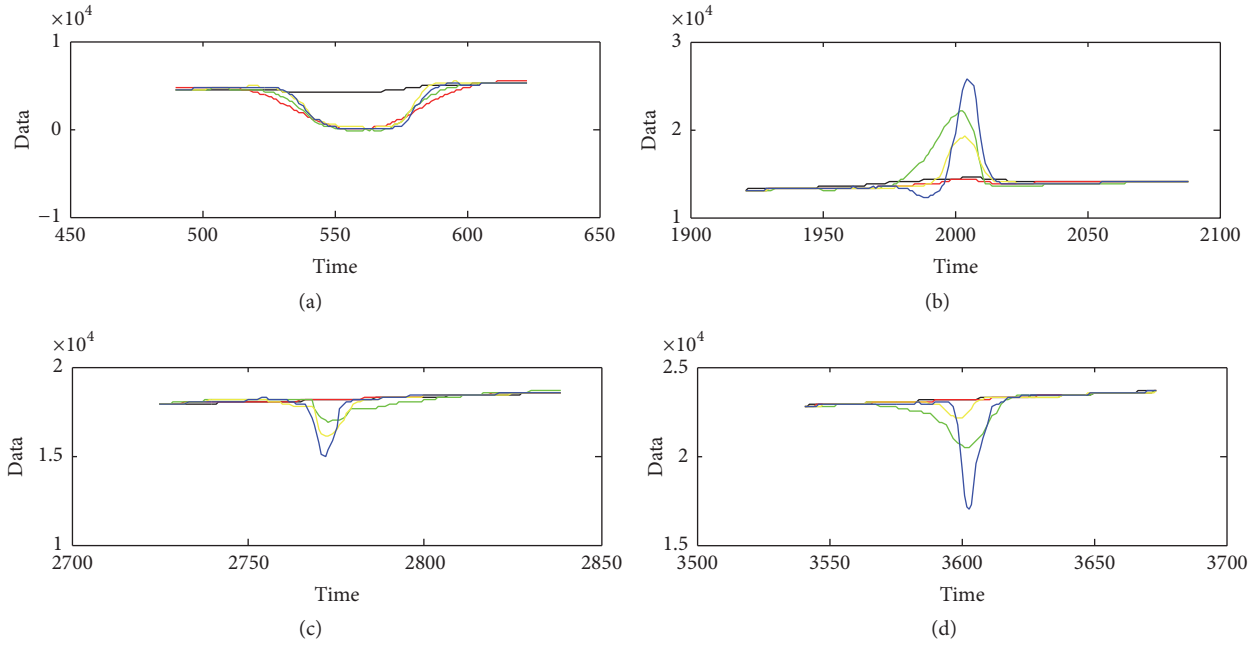


Figure 14: The enlarged details of Figure 13.

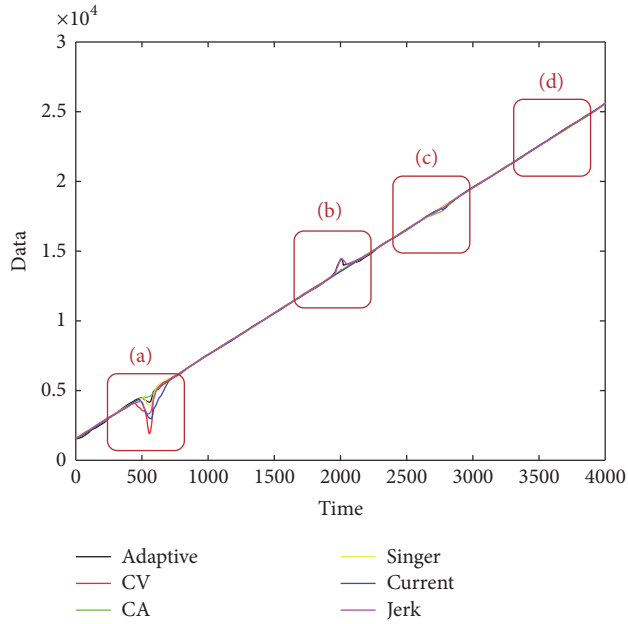


Figure 15

gives the detailed estimation information where there exists large block of outliers. By the different estimation covariance in Table 2, it can be concluded that different model can achieve different estimation results. On the other hand, less CSR means less TP value, which means less computing time cost. We also note that less CSR can also result in less Cov values of the estimation. This excellent characteristic helps in choosing an appropriate CSR for good estimation when the Cov and TP values are small. From Table 2, we can conclude that the developed method can achieve the best main trend when CSR is 1% with the Singer model, because it can achieve the lowest Cov value with the lowest TP.

This paper gives a method to estimate the main trend for the "dirty" time series of the storage volume series of the data center. We combine the irregular compressions based on Kalman estimation method together as well as the cubic spline interpolation reconstruction algorithm to extract the useful main information and then the average method is used to get the final main trend. We test this developed method on

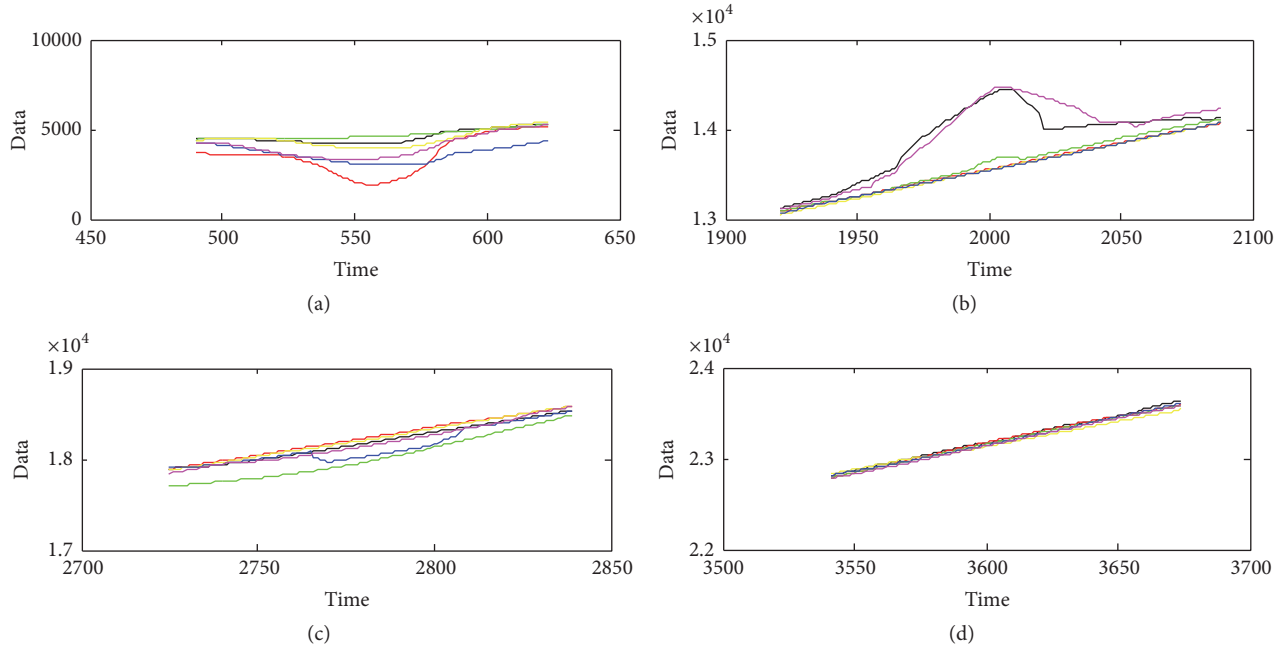


Figure 16

a storage volume series offered by an Internet company. It can be found that our developed method can estimate the main trend of a storage volume time series more accurately than PMU, Loess, and MA methods. And the accurate main trend is helpful for predicting the future storage volume value. We would like to mention that the developed algorithm has been used in practice and, together with the prediction algorithms, it has received high accuracy in practice.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Authors' Contributions

Beibei Miao wrote the manuscript and designed the experiments and Chao Dou implemented experiments. Xuebo Jin conceived the idea for the manuscript and contributed with the theoretical analysis.

Acknowledgments

This work is partially supported by NSFC (Grant nos. 61273002 and 61673002), the Third Baidu Campus Cooperation Project (181415PO1914), and the Key Science and Technology Project of Beijing Municipal Education Commission of China (no. KZ201510011012).

References

[1] T. C. Redman, "The impact of poor data quality on the typical enterprise," *Communications of the ACM*, vol. 41, no. 2, pp. 79–82, 1998.

[2] W. E. Wayne, "Data Quality and the Bottom Line: Achieving Business Success through a Commitment to High Quality Data," The Data warehouse Institute (TDWI) report, 2004, http://www.dw-institute.com.

[3] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "A seasonal-trend decomposition procedure based on loess," *Journal of Official Statistics*, vol. 6, pp. 3–73, 1990.

[4] O. Vallis, J. Hochenbaum, and A. Kejariwal, "STL: a novel technique for long-term anomaly detection in the cloud," in *Proceedings of the 6th USENIX Conference on Hot Topics in Cloud Computing*, p. 15, 2014.

[5] S. Tsani, "On the relationship between resource funds, governance and institutions: evidence from quantile regression analysis," *Resources Policy*, vol. 44, pp. 94–111, 2015.

[6] J.-C. Shen, C.-H. Chang, S.-J. Wu, C.-T. Hsu, and H.-C. Lien, "Real-time correction of water stage forecast using combination of forecasted errors by time series models and Kalman filter method," *Stochastic Environmental Research and Risk Assessment*, vol. 29, no. 7, pp. 1903–1920, 2015.

[7] P. J. Sherman, T. Jónsson, and H. Madsen, "A Kalman filter based DSP method for prediction of seasonal financial time series with application to energy spot price prediction," in *Proceedings of the IEEE Statistical Signal Processing Workshop (SSP '11)*, pp. 33–36, IEEE, Nice, France, June 2011.

[8] W. Kleynhans, J. C. Olivier, K. J. Wessels, F. van den Bergh, B. P. Salmon, and K. C. Steenkamp, "Improving land cover class separation using an extended Kalman filter on MODIS NDVI time-series data," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 2, pp. 381–385, 2010.

[9] F. Lu and J. Li, "Application of Kalman filter method based on time series model in the landslide deformation forecast," in *Proceedings of the 2nd Conference on Environmental Science and Information Application Technology (ESIAT '10)*, pp. 95–98, IEEE, Zhumadian, China, July 2010.

[10] K. Krishnamurthy, "A method for estimating the lyapunov exponents of chaotic time series corrupted by random noise using extended Kalman filter," in *Mathematical Modelling and Scientific Computation*, pp. 237–244, Springer, 2012.

[11] X. Wu and Z. Song, "Online chaotic time-series prediction with the derivative-free extended Kalman filter," in *Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA '08)*, pp. 2360–2364, IEEE, Chongqing, China, June 2008.

[12] J. V. T. Sørensen and H. Madsen, "Water level prediction skill of an operational marine forecast using a hybrid Kalman filter and time series modeling approach," in *Proceedings of the Celabrating the Past... Teaming Toward the Future*, vol. 5, September 2003.

[13] X. Wu and Y. Wang, "Extended and Unscented Kalman filtering based feedforward neural networks for time series prediction," *Applied Mathematical Modelling*, vol. 36, no. 3, pp. 1123–1131, 2012.

[14] Y. Wen and H. Wang, "Fuzzy prediction of time series based on Kalman filter with SVD decomposition," in *Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '09)*, vol. 4, pp. 458–462, Tianjin, China, August 2009.

[15] M. Xu and M. Han, "Online prediction for multivariate time series by echo state network based on square-root cubature Kalman filter," in *Proceedings of the 33rd Chinese Control Conference (CCC'14)*, pp. 5065–5070, July 2014.

[16] A. Gopalakrishnan, N. S. Kaisare, and S. Narasimhan, "Incorporating delayed and infrequent measurements in Extended Kalman Filter based nonlinear state estimation," *Journal of Process Control*, vol. 21, no. 1, pp. 119–129, 2011.

[17] I. Peñarrocha, R. Sanchis, and J. A. Romero, "State estimator for multisensor systems with irregular sampling and time-varying delays," *International Journal of Systems Science*, vol. 43, no. 8, pp. 1441–1453, 2012.

[18] X.-B. Jin, X.-F. Lian, Y. Shi, and L. Wang, "Data driven modeling under irregular sampling," in *Proceedings of the 32nd Chinese Control Conference (CCC '13)*, pp. 4731–4734, IEEE, Xi'an, China, July 2013.

[19] X.-B. Jin, J.-J. Du, and J. Bao, "Target tracking of a linear time invariant system under irregular sampling," *International Journal of Advanced Robotic Systems*, vol. 9, no. 5, article 219, 2012.

[20] B.-B. Miao, C. Dou, and X.-B. Jin, "Main trend extraction of the storage volume for internet data center," in *Proceedings of the International Symposium on Computer, Consumer and Control (IS3C '16)*, pp. 990–993, Xi'an, China, July 2016.

[21] M. Hashirao, T. Kawase, and I. Sasase, "A Kalman filter merging CV and acceleration estimation model using mode probabilities," in *Proceedings of the RADAR 2002*, October 2002.

[22] X. R. Li and V. P. Jilkov, *Survey of Maneuvering Target Tracking: Dynamic Models*, AeroSense, 2000.

[23] X. L. Chen, Y. J. Pang, Y. Li, and D. Li, "AUV sensor fault diagnosis based on STF-singer model," *Chinese Journal of Scientific Instrument*, vol. 31, no. 7, pp. 1502–1508, 2010.

[24] X.-B. Jin, J.-J. Du, and J. Bao, "Maneuvering target tracking by adaptive statistics model," *The Journal of China Universities of Posts and Telecommunications*, vol. 20, no. 1, pp. 108–114, 2013.

Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2016, Article ID 3045254, 8 pages
http://dx.doi.org/10.1155/2016/3045254

Research Article

Artificial Neural Network and Genetic Algorithm Hybrid Intelligence for Predicting Thai Stock Price Index Trend

Montri Inthachot,[1] Veera Boonjing,[2] and Sarun Intakosum[1]

[1]*Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand*
[2]*International College, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand*

Correspondence should be addressed to Montri Inthachot; 56605006@kmitl.ac.th

Received 10 August 2016; Accepted 17 October 2016

Academic Editor: Jorge Reyes

This study investigated the use of Artificial Neural Network (ANN) and Genetic Algorithm (GA) for prediction of Thailand's SET50 index trend. ANN is a widely accepted machine learning method that uses past data to predict future trend, while GA is an algorithm that can find better subsets of input variables for importing into ANN, hence enabling more accurate prediction by its efficient feature selection. The imported data were chosen technical indicators highly regarded by stock analysts, each represented by 4 input variables that were based on past time spans of 4 different lengths: 3-, 5-, 10-, and 15-day spans before the day of prediction. This import undertaking generated a big set of diverse input variables with an exponentially higher number of possible subsets that GA culled down to a manageable number of more effective ones. SET50 index data of the past 6 years, from 2009 to 2014, were used to evaluate this hybrid intelligence prediction accuracy, and the hybrid's prediction results were found to be more accurate than those made by a method using only one input variable for one fixed length of past time span.

1. Introduction

Stock index, trend, and market predictions present a challenging task for researchers because movement of stock index is the result of many possible factors such as a company's growth and profit-making capacity, local economic, social, and political situations, and global economic situation. Good predictions are crucial for minimizing investment risk and maximizing return.

There are 2 kinds of stock analyses: fundamental and technical. The first kind is an analysis of the intrinsic value of a stock based on consideration of basic factors such as a company's growth and profit-making capacity, the growth of its industrial group, and the economic trend. The second kind, on the other hand, is a mathematical analysis based on past stock index records. The simplest analysis of this kind is to make prediction by observing stock movement trend in a graph. More sophisticated analyses employ complex statistical methods and machine learning algorithms.

Artificial Neural Network (ANN) is one of the popular machine learning algorithms that has been applied for time series forecasting and a widely accepted method for predictions of stock index, trend, and market [1, 2]. Kimoto et al. [3] were the first, in 1990, to apply a modular neural network machine learning algorithm to predict the movement of stock index of Tokyo Stock Exchange and the best times to buy and sell its stocks. Later on, ANN was developed and widely applied to stock analysis. For example, Wu and Lu [4] used ANN to predict S&P 500 stock index, compared its prediction results with those made by a Box-Jenkins model, and reported that ANN made more accurate predictions, while Zhang and Wu [5] used Improved Bacterial Chemotaxis Optimization (IBCO) with Backpropagation Neural Network (BPNN) to predict the same index. Birgul et al. [6] used ANN to predict ISE index. Bollen et al. [7] used data posted on Twitter to predict Dow Jones index. Guresen et al. [8] used 4 models—ANN Multilayer Perceptron (MLP), Dynamic Architecture for Artificial Neural Network (DAN2), GARCH-MLP, and GARCH-DAN2—to predict NASDAQ index and found that MLP was the most accurate. Wang et al. [9] combined Elman recurrent neural networks with stochastic time effective function to predict SSE, TWSE, KOSPI, and Nikkei225. Not

only for predicting established stock markets, ANN was also used for predicting emerging ones. For example, Kara et al. [2] used ANN and SVM to predict the movement of Turkish ISE 100 index by importing several technical indicators and found that ANN's predictions were accurate. Patel et al. [10] proposed a preparation of trend deterministic data of technical indicators prior to import into models and found that it gave better prediction results than those given by a conventional import procedure when the indicators were imported into 4 models—ANN, SVM, Random Forest, and Naïve-Bayes classifier models—that were used to analyze CNX Nifty and S&P Bombay Stock Exchange markets. Manish and Thenmozhi [11] used ANN, SVM, logit, and Random Forest to predict the daily movement of direction of S&P CNX NIFTY Index and found that SVM outperformed the other models. In all of the works above, either ANN or SVM was the top performer. Most recently, Inthachot et al. [12] imported 10 technical indicators into ANN and SVM, used the models to predict the movement of Thailand's SET50 index, and found that ANN was more accurate than SVM, but both were still low in accuracy and needed to be further developed.

The Stock Exchange of Thailand (SET) is an emerging stock market in the TIP group (Thailand, Indonesia, and the Philippines) that has attracted the attention of Asian and global investors alike. When SET started to operate on April 30, 1975, only 16 public companies were registered; in 2015, the number exceeded 500. SET50 index is an index calculated from the stock prices of the top 50 companies registered in SET in terms of large market capitalization and high liquidity. Accurate prediction of SET50 index trend especially helps short-term investors to reduce risk and make profit from SET50 Futures and SET50 Index Options of the TFEX Futures markets.

As mentioned that the accuracy of SET50 index predictions based on technical indicators calculated from one past time span was still low, this study proposes importing technical indicators of which each is represented by 4 input variables based on 4 past time spans of different lengths—3-, 5-, 10-, and 15-day spans before the day of prediction—in order to generate more diverse subsets of input which is then culled down to a manageable number of effective ones by Genetic Algorithm (GA) and passed onto ANN to make prediction of SET50 index trend. Our contribution to application of GA to ANN was to use GA for finding a manageable number of effective subsets of input into ANN in order to improve the hybrid overall trend prediction accuracy.

The remainder of this paper is organized into the following sections: Section 2 is a literature review; Section 3 describes the methodology, research data, preprocessing of the data, prediction models, and measurement accuracy; Section 4 shows the experimental results and discussion; and Section 5 concludes the study.

2. Literature Review

This review focuses on several studies that have applied ANN to predict stock price and index in both established and emerging markets. Leung et al. [13] used various types of models based on multivariate classification method to predict stock index trend and reported that classification models (linear discriminant analysis, logit, probit, and probabilistic neural network) outperformed level estimation models (exponential smoothing, multivariate transfer function, vector autoregression with Kalman filter, and multilayered feedforward neural network) in terms of prediction accuracy of stock market movement direction and maximum return of investment trading. Chen et al. [14] used probabilistic neural network (PNN) to predict Taiwan Stock Exchange movement direction and applied the prediction to formulating trading strategies. They found that the prediction results obtained from PNN were more accurate than those obtained from GMM-Kalman filter and random walk. Altay and Satman [15] used ANN and linear regression to predict an emerging market movement direction and found that ANN gave more accurate predictions: 57.8%, 67.1%, and 78.3% for daily, weekly, and monthly data, respectively. Kara et al. [2] used ANN and SVM to predict Istanbul Stock Exchange (ISE) movement direction based on stock index data of 1997–2007 and employed 10 technical indicators as input variables—simple moving average, weighted moving average, momentum, stochastic K%, stochastic D%, RSI, moving average convergence divergence (MACD), Williams' R%, A/D oscillator, and CCI. The prediction accuracies of their ANN model were found to be 99.27% for the training data set and 76.74% for the test data set, while those of the SVM model were 100% for the training set but only 71.52% for the test data set. Chang et al. [16] used an evolving partially connected neural networks (EPCNNs) model and technical indicator input variables to predict the stock price movement of Taiwan Stock Exchange (TSE). The architecture of EPCNNs was different from that of ANN: connections between neurons were random; more than one hidden layer was accommodated; and weights were trained and adjusted with GA. They found that their proposed model gave more accurate predictions than those obtained from BPN, TSK fuzzy system, and multiple regression analysis. Patel et al. [10] proposed using deterministic input variables with ANN, SVM, Random Forest, and Naïve-Bayes models to predict Indian stock market index trend. They constructed a layer for converting 10 continuous input variables employed in a study by Kara et al. [2] into deterministic input variables before incorporating them into the models. The prediction results obtained from ANN, SVM, Random Forest, and Naïve-Bayes were 86.69%, 89.33%, 89.33%, and 90.19% accurate, respectively, which were higher than those obtained from models using continuous variables, the highest of which was 83.56% obtained from Random Forest model.

For the case of Thailand's stock market, Sutheebanjard and Premchaiswadi [17] used backpropagation neural network (BPNN) to forecast SET Index movement during July 2–December 30, 2004 (124 days), and obtained predictions with a mean square error (MSE) of 234.68 and a mean absolute percentage error (MAPE) of 1.96%. Inthachot et al. [12] applied ANN and SVM to predicting Thailand's SET50 index movement by employing the same 10 technical indicators that Kara et al. [2] used and the index data of 2009–2014 and found that year-by-year one-day prediction results by ANN were

Table 1: The number of up and down movements of SET50 index during 2009–2014.

Year	Up (times)	Up (%)	Down (times)	Down (%)	Total
2009	137	56.38	106	43.62	243
2010	138	57.02	104	42.98	242
2011	119	48.77	125	51.23	244
2012	140	57.14	105	42.86	245
2013	126	51.43	119	48.57	245
2014	135	55.10	110	44.90	245
Total	795	54.30	669	45.70	1,464

more accurate than those by SVM. ANN's average accuracy for this study was quite low, at 56.30%, when compared with the accuracy of the prediction results of ISE stock index movement which was most likely due to considerably wilder fluctuation of SET50 index values.

Readers who would like to have a comprehensive overview of recent stock market forecasting research studies should consult a review paper by Atsalakis and Valavanis [1].

3. Methodology

3.1. Data Preparation and Preprocessing. This work used a data set of daily SET50 index at closing time between January 5, 2009, and December 30, 2014 (1,464 days). During this period, the stock index moved up 795 times (54.30%) and down 669 times (45.70%), as shown in Table 1.

The data set was divided into 5 groups for 5-fold cross-validation runs, as shown in Table 2. A total of 5 runs were made in which each run used one group of data as a test data set and the other 4 groups as training data sets so that every group was used as a test data set exactly once.

From all of the widely accepted 11 technical indicators for stock price and index forecast [2, 10, 12, 18], each input variable was calculated by the equation of its corresponding indicator shown in Table 3. Four input variables were derived from each technical indicator in which each of the four variables was calculated based on one of these 4 past time span lengths: 3, 5, 10, and 15 days, making up a total of $11 \times 4 = 44$ input variables.

All input variables were normalized to $[-1, 1]$ so that they all had the same weight. The only output variable could take a value of either 0 or 1—a value of 0 means that the predicted next-day SET50 index was lower than the prediction day index (down trend) while a value of 1 means that the predicted next-day index was higher than the prediction day index (up trend).

3.2. Prediction Models

3.2.1. Artificial Neural Network (ANN). ANN (introduced by McCulloch and Pitts [19]) is a machine learning model that mimics an aspect of human learning from past experience to predict a future outcome. ANN is widely adopted in research studies on stock price and index forecast [1, 2, 8, 16, 20]. It has already been used for predicting SET50 index trend [12] in a study and found to make more accurate predictions than support vector machine (SVM). However, its absolute accuracy was still not very good. This study attempted to develop it further to make it more accurate for predicting next-day SET50 index movement. Our ANN model was a three-layered feedforward model consisting of an input layer, a hidden layer, and an output layer. Past stock trading data were represented by 11 technical indicators. Each technical indicator was imported into ANN as 4 variables based on 4 different past time length spans, making up a total of 44 variables in the input layer. The number of nodes in the hidden layer was set to 100, following the optimum number used in a study by Inthachot et al. [12]. The transfer function between nodes in the input layer and the hidden layer and between nodes in the hidden layer and the output layer was tan sigmoid. The output layer had 1 neuron with log sigmoid transfer function. The computed output could take a value between 0 and 1 where a value equal to or less than 0.5 indicates a downward index movement and a value higher than 0.5 indicates an upward movement. A weight was assigned between each pair of connected nodes. Initially, all of the weights were randomly generated; then they were adjusted during the training period by a gradient descent with momentum method.

The model parameters that needed to be set were the number of hidden layer neurons (n), the learning rate (lr), the momentum constant (mc), and the number of iterations for learning (ep). They were set to $n = 100$, lr = 0.1, mc = 0.1, and $ep = 8{,}000$ following those that gave the best accuracy in the study by Inthachot et al. [12] mentioned above.

3.2.2. A Hybrid Intelligence of ANN and Genetic Algorithm (GA). ANN has several disadvantages such as long training time, unwanted convergence to local instead of global optimal solution, and large number of parameters; therefore, there have been attempts to remedy some of these disadvantages by combining ANN with another algorithm that can take care of a specific problem. An algorithm that has frequently been hybridized with ANN is GA. In 1990, Whitley et al. [21] began to use GA to optimize weighted connections and find a good architecture for neural network connections. In 2006, Kim [22] proposed a hybrid model of ANN with GA that performs instance selection to reduce dimensionality of data. In 2012, Karimi and Yousefi [23] used GA to find a set of weights for connections to each node in an ANN model and determine correlation of density in nanofluids. Sangwan et al. [24] proposed an integrated ANN and GA for predictive modeling and optimization of turning parameters to minimize surface roughness. Some other successful examples of ANN-GA hybrid applications are network intrusion detection [25] and cancer patient classification [26]. Inspired by these successes, this study attempted to use GA to solve a feature selection problem—to find effective subsets of input into ANN.

The rationale behind our idea of using a hybrid intelligence of ANN and GA was that it should be better to use, first, multiple input variables (4 in this study) for each technical indicator based on different past time spans (3, 5, 10, and 15 days) and, second, a small number of effective subsets of input variables that would be imported. Since the number of subsets

Table 2: The number of up and down movements of the whole set of daily index in 5 cross-validation runs.

Year	Five runs of cross-validation										Total
	1st run		2nd run		3rd run		4th run		5th run		
	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down	
2009	27	22	31	18	24	25	32	16	23	25	243
2010	28	20	30	18	37	11	17	32	26	23	242
2011	26	23	25	24	27	22	22	27	19	29	244
2012	30	19	22	27	30	19	24	25	34	15	245
2013	28	21	27	22	23	26	24	25	24	25	245
2014	26	23	23	26	36	13	19	30	31	18	245
Total	165	128	158	135	177	116	138	155	157	135	1,464

Table 3: Technical indicators used in this study and their equations [2, 18].

Indicator name	Equation	Level (n)	Total
Simple n-day moving average	$\frac{C_t + C_{t-1} + \cdots + C_{t-n-1}}{n}$	3, 5, 10, 15	4
Weighted n-day moving average	$\frac{(n)C_t + (n-1)C_{t-1} + \cdots + C_{t-(n-1)}}{n + (n-1) + \cdots + 1}$	3, 5, 10, 15	4
Momentum	$C_t - C_{t-n}$	3, 5, 10, 15	4
Stochastic K%	$\frac{C_t - LL_{t-(n-1)}}{HH_{t-(n-1)} - LL_{t-(n-1)}} \times 100$	3, 5, 10, 15	4
Stochastic D%	$\frac{\sum_{i=0}^{n-1} K_{t-i}\%}{n}$	3, 5, 10, 15	4
Relative Strength Index (RSI)	$100 - \frac{100}{1 + (\sum_{i=0}^{n-1} (\mathrm{UP}_{t-i}/n))/(\sum_{i=0}^{n-1} (\mathrm{DW}_{t-i}/n))}$	3, 5, 10, 15	4
Moving Average Convergence Divergence (MACD)	$\mathrm{MACD}(n)_{t-1} + \frac{2}{n+1} \times (\mathrm{DIFF}_t - \mathrm{MACD}(n)_{t-1})$	3, 5, 10, 15	4
Larry William's R%	$\frac{H_n - C_t}{H_n - L_n} \times -100$	3, 5, 10, 15	4
Commodity Channel Index (CCI)	$\frac{M_t - SM_t}{0.015 D_t}$	3, 5, 10, 15	4
Rate of change	$\frac{C_t - C_{t-n}}{C_{t-n}} \times 100$	3, 5, 10, 15	4
Average Directional Index (ADX)	$\mathrm{SMA}\left(\frac{+\mathrm{DI}_n - (-\mathrm{DI}_n)}{+\mathrm{DI}_n + (-\mathrm{DI}_n)}\right)$	3, 5, 10, 15	4
Total			44

Note: n is n-day period times ago; C_t is closing price; L_t is low price at time t; H_t is high price at time t; DIFF = $\mathrm{EMA}(12)_t - \mathrm{EMA}(26)_t$; EMA is exponential moving average; $\mathrm{EMA}(k)_t = \mathrm{EMA}(k)_{t-1} + \propto (C_t - \mathrm{EMA}(k)_{t-1})$; $\propto$ is smoothing factor = $2/(1+k)$; $k = 10$ in $k-$day exponential moving average; LL_t and HH_t are the lowest low and highest high in the last t days, respectively; $M_t = (H_t + L_t + C_t)/3$; $SM_t = \sum_{t=1}^{n} M_{t-i+1}/n$; $D_t = \sum_{i=1}^{n} |M_{t-i+1} - SM_t|/n$; UP_t is upward index change at time t, DW_t is downward index change at time t; $+\mathrm{DI}_n$ is plus directional indicator and $-\mathrm{DI}_n$ is minus directional indicator.

of 44 variables is astronomical 2^{44}, it would take too much computation time to process them. GA took care of that. GA is an algorithm that is especially powerful at feature selection, so we used it to find better subsets of input variables.

GA, a search algorithm based on concepts of natural selection and genetics, was officially introduced by Holland in the 1990s [27]. The underlying principles of GA are to generate an initial population of chromosomes (search solutions) and then use selection and recombination operators generate a new, more effective population which eventually will have the fittest chromosome (optimal value) among them.

The 10 steps of operation of ANN and GA hybrid intelligence are as follows.

Step 1 (initialization of population). Generate an initial population of chromosomes which are bit strings of randomly

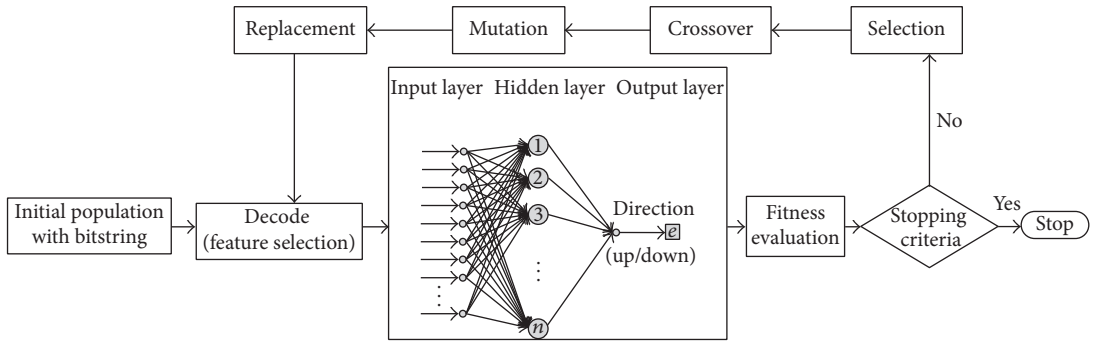


Figure 1: Steps of operation of ANN and GA hybrid intelligence.

generated binary values. The chromosome and population sizes that we used were 44 and 10, respectively.

Step 2 (decoding). Decode chromosomes (bit strings) to find which input variables will be selected.

Step 3 (ANN). Run three-layered feedforward ANN model to make prediction of next-day SET50 index. The parameters in the model that we used were the same as those reported by Inthachot et al. [12].

Step 4 (fitness evaluation). Take the prediction accuracy of each chromosome from ANN as its fitness value for GA.

Step 5 (stopping criterion). Determine whether to continue or exit the loop. The stopping criterion was not more than 10 generations.

Step 6 (selection). Select chromosomes to cross over using tournament selection technique. A tournament selection involves running several tournaments on a few chromosomes chosen at random from the population. The winner of each tournament is selected for crossover.

Step 7 (crossover). Apply an arithmetic crossover operator that defines a linear combination of two chromosomes.

Step 8 (mutation). Inject new genes into the population with uniform mutation operator and generate a random slot number of the crossed-over chromosome as well as flip the binary value in that slot.

Step 9 (replacement). Replace old chromosomes with two best offspring chromosomes for the next generation.

Step 10 (loop). Go to Step 2.

All of the steps are shown in Figure 1.

3.3. Fitness Evaluation. We used accuracy to determine chromosome selection (subsets of input variables)—chromosomes that would generate the next generation—as well as to measure the performance of the prediction model. Fitness values in GA were taken as the accuracy values that can be calculated as below:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (1)$$

where TP is true positive, FP is false positive, TN is true negative, and FN is false negative.

4. Results and Discussion

A hybrid intelligence of ANN and GA models was developed for predicting SET50 index movement during 2009–2014. Each year's trading data during this period were converted into 11 technical indicators, each of which represented by 4 input variables based on different lengths of past time spans, and hence 44 input variables. All input variables were identically normalized and subsets of them were selected by GA and imported into ANN that would use them to make stock index movement prediction. Fivefold cross-validation runs were made to guarantee reliability. This hybrid intelligence was coded and run in a MATLAB software environment.

Results from successful runs are presented in Figures 2–7 and Table 4. Figures 2–7 illustrate the best fitness value (which reflects the prediction accuracy) achieved in each generation for each year of prediction.

The trend prediction accuracy performance of our proposed method was compared to another study [12] only because even though there have been several works on SET, all of them reported either the mean square error of stock price or stock price index [17, 28–30]. Moreover, it was not compared to those of any binary choices model because it has been widely reported in the literature that their prediction performances were inferior to ANN in regard to stock market prediction [2, 11, 12].

Table 4 compares the accuracies achieved by the model that Inthachot et al. [12] used in their study and those achieved by our hybrid intelligence of ANN and GA models.

Table 4: Prediction performances of Inthachot et al. [12] model and this study's model.

Year	Accuracy		
	Inthachot et al. [12]	This study	Percentage increase
2009	0.5602	0.6293	12.3349%
2010	0.5257	0.6000	14.1335%
2011	0.5986	0.6887	15.0518%
2012	0.5592	0.6041	8.0293%
2013	0.5714	0.6531	14.2982%
2014	0.5796	0.6408	10.5590%
Average	0.5658	0.6360	12.4011%

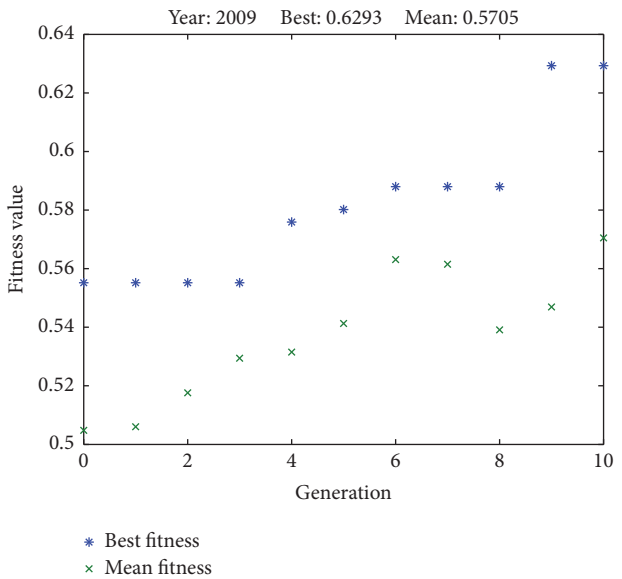


Figure 2: Fitness values in each generation (predicting SET50 index in 2009).

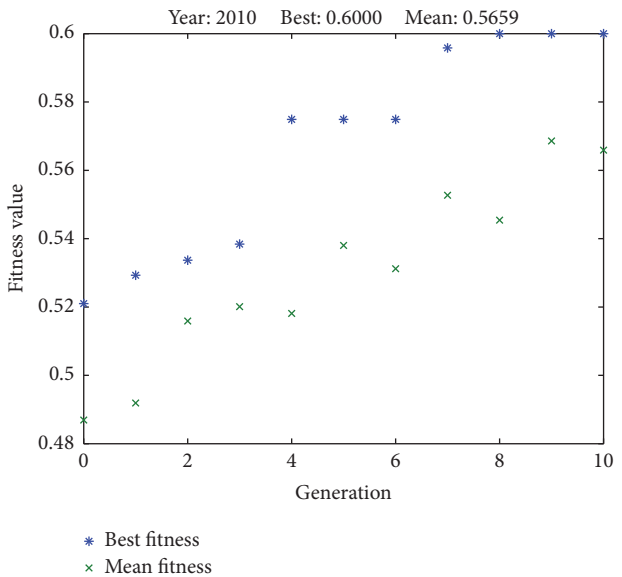


Figure 3: Fitness values in each generation (predicting SET50 index in 2010).

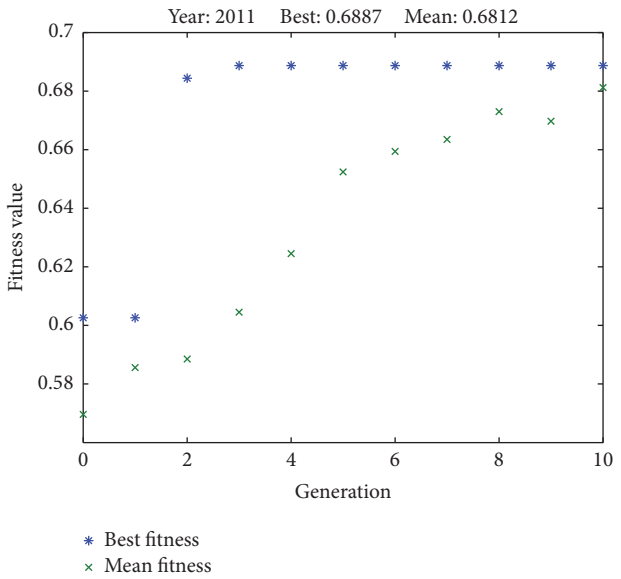


Figure 4: Fitness values in each generation (predicting SET50 index in 2011).

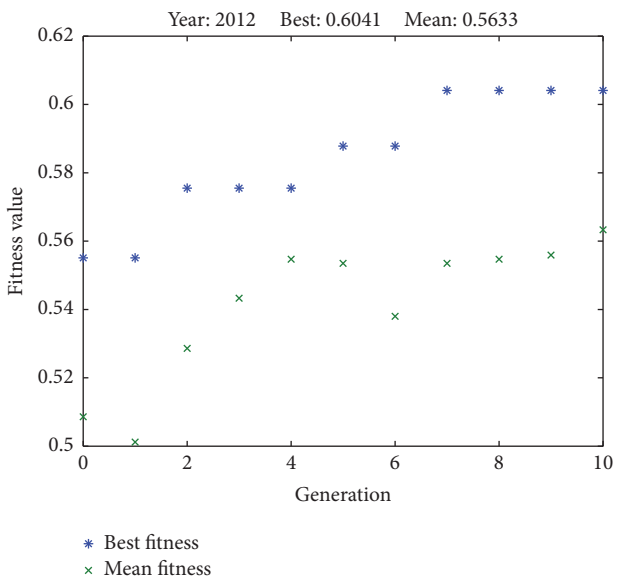


Figure 5: Fitness values in each generation (predicting SET50 index in 2012).

It can be seen that the model that Inthachot et al. [12] used achieved its lowest prediction accuracy of 52.57% for the year 2010, highest accuracy of 59.86% for the year 2011, and average accuracy of 56.58%. On the other hand, our hybrid intelligence achieved its lowest prediction accuracy of 60.00% for the year 2010, highest accuracy of 68.87% for the year 2011, and average accuracy of 63.60%.

This study's hybrid intelligence predicted more accurately than the model used by Inthachot et al. [12] for every year during the selected period with the lowest percentage improvement of 8.0293%, the highest of 15.0518%, and the average improvement of 12.4011%. In order to confirm this

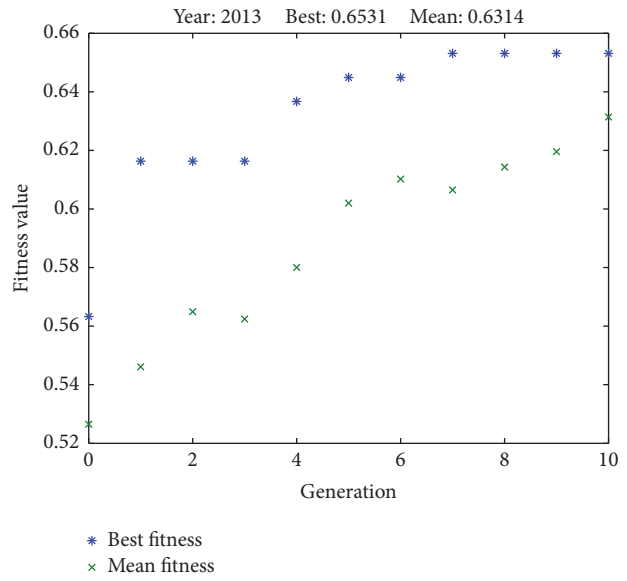


Figure 6: Fitness values in each generation (predicting SET50 index in 2013).

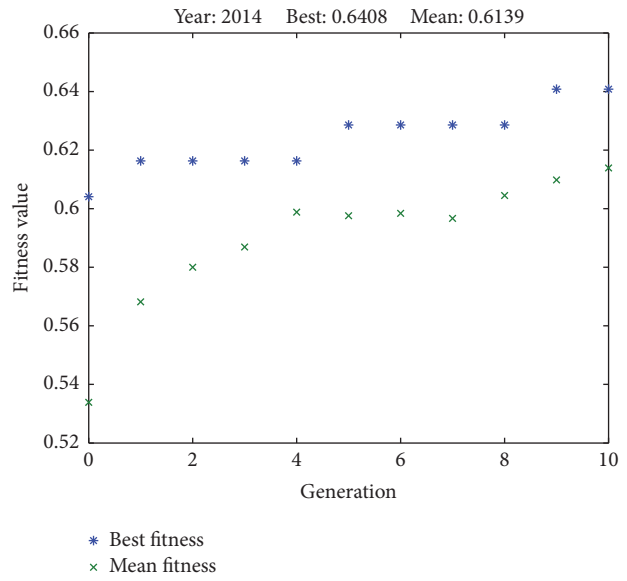


Figure 7: Fitness values in each generation (predicting SET50 index in 2014).

conclusion statistically, we compared them using a t-test at 0.05 level of significance and found that the P value of the right tail was 0.0009; hence the conclusion is valid.

5. Conclusion

In this study, we developed a hybrid intelligence of ANN and GA models for predicting SET50 stock index movement and tested it on a large set of past stock trading data. The purpose of the development was to achieve a better prediction accuracy than that obtained by a previous ANN model that we have developed [12]. Test results show that the hybrid intelligence has accomplished this purpose, gaining an average improvement of 12.4011%. It is 63.60% average prediction accuracy; however, it was still not very high and we are looking into combining ANN with other machine learning models in order to gain a higher prediction accuracy.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] G. S. Atsalakis and K. P. Valavanis, "Surveying stock market forecasting techniques—part II: soft computing methods," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5932–5941, 2009.

[2] Y. Kara, M. Acar Boyacioglu, and Ö. K. Baykan, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul Stock Exchange," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5311–5319, 2011.

[3] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka, "Stock market prediction system with modular neural networks," in *Proceedings of the 1990 International Joint Conference on Neural Networks (IJCNN '90)*, vol. 1, pp. 1–6, Washington, DC, USA, June 1990.

[4] S.-I. Wu and R.-P. Lu, "Combining artificial neural networks and statistics for stock-market forecasting," in *Proceedings of the 21st Annual ACM Computer Science Conference*, pp. 257–264, New York, NY, USA, February 1993.

[5] Y. Zhang and L. Wu, "Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network," *Expert Systems with Applications*, vol. 36, no. 5, pp. 8849–8854, 2009.

[6] E. Birgul, M. Ozturan, and B. Badur, "Stock market prediction using artificial neural networks," in *Proceedings of the in 3rd Hawaii International Conference on Business*, 2003.

[7] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, 2011.

[8] E. Guresen, G. Kayakutlu, and T. U. Daim, "Using artificial neural network models in stock market index prediction," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10389–10397, 2011.

[9] J. Wang, J. Wang, W. Fang, and H. Niu, "Financial time series prediction using elman recurrent random neural networks," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 4742515, 14 pages, 2016.

[10] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques," *Expert Systems with Applications*, vol. 42, no. 1, pp. 259–268, 2015.

[11] K. Manish and M. Thenmozhi, *Forecasting Stock Index Movement: A Comparison of Support Vector Machines and Random Forest*, Social Science Research Network, Rochester, NY, USA, 2005.

[12] M. Inthachot, V. Boonjing, and S. Intakosum, "Predicting SET50 index trend using artificial neural network and support vector machine," in *Current Approaches in Applied Artificial*

Intelligence, M. Ali, Y. S. Kwon, C.-H. Lee, J. Kim, and Y. Kim, Eds., pp. 404–414, Springer, Berlin, Germany, 2015.

[13] M. T. Leung, H. Daouk, and A.-S. Chen, "Forecasting stock indices: a comparison of classification and level estimation models," *International Journal of Forecasting*, vol. 16, no. 2, pp. 173–190, 2000.

[14] A.-S. Chen, M. T. Leung, and H. Daouk, "Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index," *Computers and Operations Research*, vol. 30, no. 6, pp. 901–923, 2003.

[15] E. Altay and M. H. Satman, "Stock market forecasting: artificial neural network and linear regression comparison in an emerging market," SSRN Scholarly Paper ID 893741, Social Science Research Network, Rochester, NY, USA, 2005.

[16] P.-C. Chang, D.-D. Wang, and C.-L. Zhou, "A novel model by evolving partially connected neural network for stock price trend forecasting," *Expert Systems with Applications*, vol. 39, no. 1, pp. 611–620, 2012.

[17] P. Sutheebanjard and W. Premchaiswadi, "Stock Exchange of Thailand index prediction using back propagation neural networks," in *Proceedings of the 2nd International Conference on Computer and Network Technology (ICCNT '10)*, pp. 377–380, April 2010.

[18] K.-J. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.

[19] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[20] S. Banik, A. F. M. Khodadad Khan, and M. Anwer, "Hybrid machine learning technique for forecasting dhaka stock market timing decisions," *Computational Intelligence and Neuroscience*, vol. 2014, Article ID 318524, 6 pages, 2014.

[21] D. Whitley, T. Starkweather, and C. Bogart, "Genetic algorithms and neural networks: optimizing connections and connectivity," *Parallel Computing*, vol. 14, no. 3, pp. 347–361, 1990.

[22] K.-J. Kim, "Artificial neural networks with evolutionary instance selection for financial forecasting," *Expert Systems with Applications*, vol. 30, no. 3, pp. 519–526, 2006.

[23] H. Karimi and F. Yousefi, "Application of artificial neural network-genetic algorithm (ANN-GA) to correlation of density in nanofluids," *Fluid Phase Equilibria*, vol. 336, pp. 79–83, 2012.

[24] K. S. Sangwan, S. Saxena, and G. Kant, "Optimization of machining parameters to minimize surface roughness using integrated ANN-GA approach," in *Proceedings of the 22nd CIRP Conference on Life Cycle Engineering (LCE '15)*, vol. 29, pp. 305–310, Sydney, Australia, April 2015.

[25] J. Tian and M. Gao, "Network intrusion detection method based on high speed and precise genetic algorithm neural network," in *Proceedings of the International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC '09)*, vol. 2, pp. 619–622, April 2009.

[26] F. Ahmad, N. A. Mat-Isa, Z. Hussain, R. Boudville, and M. K. Osman, "Genetic Algorithm-Artificial Neural Network (GA-ANN) hybrid intelligence for cancer diagnosis," in *Proceedings of the 2nd International Conference on Computational Intelligence, Communication Systems and Networks (CICSYN '10)*, pp. 78–83, 2010.

[27] J. H. Holland, *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.

[28] S. Rimcharoen, D. Sutivong, and P. Chongstitvatana, "Prediction of the Stock Exchange of Thailand using adaptive evolution strategies," in *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '05)*, pp. 232–236, IEEE, Hong Kong, November 2005.

[29] S. Chaigusin, C. Chirathamjaree, and J. Clayden, "The use of neural networks in the prediction of the stock exchange of Thailand (SET) index," in *Proceedings of the International Conference on Computational Intelligence for Modelling Control Automation*, pp. 670–673, Vienna, Austria, 2008.

[30] W. Sirijunyapong, A. Leelasantitham, S. Kiattisin, and W. Wongseree, "Predict the stock exchange of Thailand—set," in *Proceedings of the 4th Joint International Conference on Information and Communication Technology, Electronic and Electrical Engineering (JICTEE '14)*, pp. 1–4, March 2014.

Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2016, Article ID 5329870, 6 pages
http://dx.doi.org/10.1155/2016/5329870

Research Article

A Forecasting Model for Feed Grain Demand Based on Combined Dynamic Model

Tiejun Yang, Na Yang, and Chunhua Zhu

School of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, China

Correspondence should be addressed to Chunhua Zhu; zhuchunhua@haut.edu.cn

Received 6 April 2016; Revised 28 June 2016; Accepted 14 July 2016

Academic Editor: Jorge Reyes

In order to improve the long-term prediction accuracy of feed grain demand, a dynamic forecast model of long-term feed grain demand is realized with joint multivariate regression model, of which the correlation between the feed grain demand and its influence factors is analyzed firstly; then the change trend of various factors that affect the feed grain demand is predicted by using ARIMA model. The simulation results show that the accuracy of proposed combined dynamic forecasting model is obviously higher than that of the grey system model. Thus, it indicates that the proposed algorithm is effective.

1. Introduction

The grain used in feeding is the second largest grain used in China; its quantity and proportion of the total grain consumption grow stably. It is of great significance to ensure food security in our country by exploring the changes of feed grain demand and its influencing factors. However, the special research of China's feed grain demand is scattered, which lacks objective statistics and always exists in projections of the total grain consumption. The forecasting methods of feed grain demand in existing literature can be divided into two kinds: one is using some quantitative methods such as time series regression, model of consumer demand system, and farming grain consumption, based on the analysis about the situation of the feeding food consumption over the past few years to analysis and forecast [1, 2]; the other is from the perspective of nutrition standards analysis of meat, eggs, milk, per capita consumption of aquatic products to predict the future demand for animal products and then use the ratio of feed to meat (i.e., the conversion rate of feed grains) to predict the feed grain demand [3, 4]. Actually, the feed grain demand is affected by population growth, urbanization level, per capita income (urban residents per capita income and rural ones per capita income), and other factors [5, 6], which suggest that there should be a comprehensive survey about correlation degree between the feed grain demand and its influence factors for improving the prediction accuracy, and the corresponding prediction model should be generalized. In this paper, the correlation coefficients of feed grain demand and its influence factors are calculated quantitatively on the basis of the second kind of forecasting method; then the major factors have been chosen; finally the dynamic prediction of influence factors and feed grain demand can be realized by using the ARIMA model and multiple regression model, respectively.

2. Relational Coefficient Analysis of Influence Factors to Feed Grain Demand

2.1. Grey Relational Analysis. The essence of grey relational degree is to make a geometric comparison in the data series which are responded to the changing characteristics of all factors. The closer the curves are, the greater the relational grade of the corresponding series is and vice versa. The use of the grey relational analysis can define the changing trend of all factors in this system and find out the main factors which affect the further development of the system so as to grasp the main features of things and the principal contradiction, promote, and guide the system to rapid, health, and efficient development [7]. The basic steps of grey relational analysis are as follows.

Step 1. Assume that the reference sequence is $x_0(k)$ and related comparison sequences are $x_i(k)$. They are expressed as $x_0(k) = \{x_0(1), x_0(2), \ldots, x_0(n)\}$ and

$$x_i(k) = \{x_i(1), x_i(2), \ldots, x_i(n)\}, \quad (i = 1, 2, \ldots, m). \quad (1)$$

Step 2. Dis-dimension treatment to the data sequence [8]. Here, we illustrate the initiating. Then it can get the reference sequence $y_0(k)$ and comparison sequences $y_i(k)$ ($i = 1, 2, \ldots, m$; $k = 1, 2, \ldots, n$).

Step 3. The absolute difference sequences $\Delta_{0i}(k)$ between reference sequence $y_0(k)$ and comparison sequences $y_i(k)$ are calculated by the formula

$$\begin{aligned}\Delta_{0i}(k) &= |y_0(k) - y_i(k)| \\ &= \{\Delta_i(1), \Delta_i(2), \ldots, \Delta_i(n)\}, \\ &\qquad (i = 1, 2, \ldots, m).\end{aligned} \quad (2)$$

Step 4. Identify the absolute maximum $\Delta_{\max}$ and minimum $\Delta_{\min}$ from absolute difference sequence.

Step 5. Calculate the grey relational coefficient. The formula is

$$L_{0i}(k) = \frac{(\Delta_{\max} + \Delta_{\min})}{(\Delta_{0i}(k) + \Delta_{\max})}. \quad (3)$$

Step 6. Calculate correlation degree.

$$\begin{aligned}R_{0i}(k) &= \frac{1}{n}\sum_{k=1}^{n} L_{0i}(k) \\ &= \frac{1}{n}\{L_{0i}(1) + L_{0i}(2) + \cdots + L_{0i}(n)\}.\end{aligned} \quad (4)$$

2.2. Prediction for the Feed Grain Demand by Using Multiple Linear Regression. According to grey relational analysis, the domestic population, urbanization level, and per capita income of urban and rural residents are the main factors affecting the feed grain demand. Based on the modeling principle of multiple regression model, the linear regression model of the feed grain demand is set up, the structure form of the model [9]:

$$y_0 = U_0 + U_1 x_1 + U_2 x_2 + U_3 x_3 + \varepsilon, \quad \varepsilon \sim N(0, \delta^2). \quad (5)$$

In the formula, U_1, U_2, and U_3 are the undetermined parameters (regression parameters), with ε for unobservable random error.

2.3. Prediction for Main Factors That Influence the Feed Grain Demand. The ARIMA model from literature is adopted to predict the change trend of impact factors [10]. Suppose that ω_t is the predictive value in t time of various influence factors and $\omega_{t-1}, \omega_{t-2}, \ldots, \omega_{t-p}$ are actual values of various impact factors in past p years. Setting $\omega_t = (1 - L)^d y_t$, among it, y_t is a single integer sequence with d order; ω_t is the stationary series [11]; thus the general model of the ARMA model can be expressed as

$$\begin{aligned}\omega_t = {} & \varphi_1\omega_{t-1} + \varphi_2\omega_{t-2} + \cdots + \varphi_p\omega_{t-p} + \varepsilon_t + \theta_1\varepsilon_{t-1} + \cdots \\ & + \theta_q\varepsilon_{t-q}.\end{aligned} \quad (6)$$

In the formula, p and q are, respectively, called autoregressive order number and average order number. Suppose L as the lag operator; then

$$\begin{aligned}L\omega_t &= \omega_{t-1}, \\ L^p\omega_t &= \omega_{t-p}.\end{aligned} \quad (7)$$

Equation (6) can be rewritten as

$$\varphi(L)\omega_t = \Theta(L)\varepsilon_t. \quad (8)$$

Among it, $\varphi(L) = 1 - \varphi_1 L - \varphi_2 L^2 - \cdots - \varphi_p L^p$ and $\Theta(L) = 1 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_q L^q$.

ARMA(p, q) model in formula (7) can be expressed as ARIMA(p, d, q) after d order difference transformation

$$\varphi(L)(1 - L)^d y_t = \Theta(L)\varepsilon_t. \quad (9)$$

ε_t is a white noise process with its mean value which is 0 and variance is σ^2 [12].

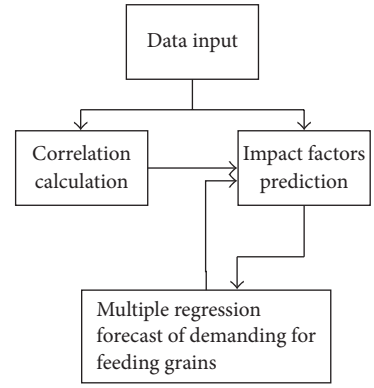


Figure 1: Dynamic prediction simulation process of feed grain demand.

3. Simulation Analysis

The dynamic simulation process based on the ARIMA model and multiple regression model to predict feed grain demand is shown in Figure 1.

The dynamic prediction algorithm of feed grain demand is shown in Figure 1; define the year of 1981 as $t = 1$ and thus 2007 as $t = 27$. The feed grain demand of urban and rural population is, respectively, expressed as $y_0(t)$ and $y_1(t)$; the three factors are, respectively, defined as $x_1(t)$, $x_2(t)$, and $x_3(t)$. According to the simulation process shown in Figure 1, the forecast process of feed grain demand in this paper is shown in the following:

(1) When t = 1~27, calculate the correlation degree and relational sequence, respectively, between $y_0(t)$ and $y_1(t)$ and $x_1(t)$, $x_2(t)$, and $x_3(t)$.

(2) Use ARIMA model to predict $x_1(t)$, $x_2(t)$, and $x_3(t)$ when $t > 27$.

(3) Use multiple regression method to predict urban feed grain demand $y_0(t)$ ($t = 28$) and rural feed grain demand $y_1(t)$ ($t = 28$).

(4) Repeat (3). Urban and rural long-term prediction of feed grain demand can be completed.

3.1. Correlation Calculation. The data about the feed grain demand, urban and rural population, urbanization level, and urban and rural residents per capita income between 1981 and 2007 are selected from *Rural China Statistical Yearbook* [13] as the training data; meanwhile the data from 2008 to 2012 are selected as the precision test data as shown in Table 1. The feed grain demand can be got by the sum of per capita meat, egg, milk, and aquatic product consumption multiplied by the urban and rural population, respectively, and then according to the conversion ratio of feed grain to meat which is 3.7 to 1, the conversion ratio to egg which is 2.7 to 1, the conversion ratio to milk which is 0.5 to 1, and the conversion ratio to aquatic material which is 0.4 to 1 to get the final result [14, 15].

The correlation degree and relational order are obtained by using the grey correlation analysis method, while the data about the feed grain demand are calculated in Table 1 as reference sequence; at the same time urban and rural population, urbanization level, and urban and rural residents per capita income are calculated as comparative sequence. The results are shown in Table 2.

As shown in Table 2, the correlation degree and relational order of various factors which affected the urban and rural feed grain demand are not completely the same; on the basis of that, it will be able to improve the prediction accuracy by predicting towns and rural feed grain demand separately.

3.2. Impact Factors Prediction. ARIMA(p, d, q) model described in Section 2.3 is adopted to predict the three factors including urban and rural population, urbanization level, and urban and rural residents per capita income. The prediction of impact factors for urban feed grain demand in 2008 is taken as an example in this paper, and the results are shown in Table 3. The forecast data will be used to forecast feed grain demand in 2008.

3.3. Prediction for Feed Grain Demand by Using Multiple Regression. The multiple regression model of urban and rural demand for feed grain demands is set up, respectively, in 2008 by using EVIEWS statistical software, while three factors mentioned above are taken as independent variables and China's urban and rural residents' feed grain demand is taken as the dependent variable. The models are shown as follows:

$$y_0 = -4240163 + 151.53x_{01} + 210419.4x_{02} - 195.0006x_{03}, \quad (10)$$

$$y_1 = -21283643 + 232.867x_{11} + 392070.5x_{12} - 543.328x_{13}. \quad (11)$$

Among them, y_0 and y_1 represent the urban and rural feed grain demand, respectively, x_{01} is urban population, and x_{11} is rural population. x_{02} and x_{12} represent urbanization level, x_{03} is urban residents per capita income, and x_{13} is rural residents per capita income. The predicted value of three factors in 2008 was typed in (10) and (11), respectively; then the value of urban and rural feed grain demand in 2008 can be calculated; the results are 9807134 tons and 6663724.9 tons.

In the above multivariate regression model of urban and rural feed grain demand, the model prediction coefficient of different years will change dynamically as the change of correlation of feed grains and affecting factors; then it forms a dynamic forecast system.

3.4. Simulation Results. The value of feed grain demand in 2008–2012 can be predicted according to (10) and (11); the result is shown in Table 4. A grey forecasting model by using residual error correction on the feed grain demand in literature [16] is also given in Table 4.

From Table 4 and combined with the feed grain demand between urban and rural areas since 1981, it can be seen that the basic trend of feed grain demand overall present rises steadily [17, 18]. The feed grain demand increased by 4 times, and the average annual growth rate is 14.8% from 1981 to 2007. Analysis shows that the income level of our country residents is low, and the consumption structure is unitary, mainly grain consumption before the reform and open policy. In recent years, the demand for animal products structure is changing and it mainly displays in the increasing demand for meat, eggs, milk, and aquatic products because people's living standards have been continuously improved.

In addition, compared with the grey system model in literature [16], the joint dynamic prediction model in this paper can track the change of impact factors, so it can achieve good long-term forecasts. Meanwhile the mean relative error of proposed model is 0.46% and has higher superiority in forecasting precision compared with traditional grey forecasting model of which the mean relative error is 6.4%. It is fully illustrated that the dynamic impact factor regression analysis method used to predict the feed grain demand is feasible.

4. Conclusion

The dynamic influence factors in combination with multivariate regression analysis method are used in this paper to forecast the feed grain demand in China since 2008. Prediction results show that China's demand for feed grains will increase year by year in the next 10 years, and the average relative error between the actual and predicted value by using the dynamic impact factor regression model is 0.46%, superior to the traditional grey system model. At present, China's feed grain demand represents more than 30% of the total demand for grain; the proportion of which feed grain demand on total demand for grain increased year by year shows the increasing influence of feed grains on food security,

Table 1: Statistical data of various impact factors.

Year	Meat		Egg		Milk		Aquatic product		Population (ten thousand people)		Per capita income (yuan)		Urbanization level (%)
	Urban	Rural	Urban	Rural	Urban	Rural	Urban	Rural	Urban	Rural	Urban	Rural	
1981	20.5	9.4	5.2	1.3	4.1	0.7	7.3	1.3	20171	79901	500.4	223.4	14.2
1982	21	9.9	5.9	1.4	4.5	0.7	7.7	1.3	21480	80174	535.3	270.1	14.4
1983	22.5	10.8	6.9	1.6	4.6	0.8	8.1	1.6	22274	80734	564.6	309.8	14.6
1984	22.8	11.5	7.6	1.8	5.2	0.8	7.8	1.7	24017	80340	652.1	355.3	14.7
1985	24	12	8.8	2.1	6.4	0.8	7.8	1.6	25094	80757	739.1	397.6	14.8
1986	25.3	12.9	7.1	2.1	4.7	1.4	8.2	1.9	26366	81141	900.9	423.8	15
1987	25.4	12.9	6.6	2.3	5.5	1.1	7.9	2	27674	81626	1002.1	462.6	15.1
1988	23.7	12	6.9	2.3	5.1	1.1	7.1	1.9	28661	82365	1180.2	544.9	15.3
1989	23.9	12.3	7.1	2.4	4.2	1	7.6	2.1	29540	83164	1373.9	601.5	15.4
1990	25.2	12.6	7.3	2.4	4.6	1.1	7.7	2.1	30195	84138	1510.2	686.3	15.5
1991	26.6	13.5	8.3	2.7	4.7	1.3	8	2.2	31203	84620	1700.6	708.6	15.9
1992	26.5	13.3	9.5	2.9	5.5	1.5	8.2	2.3	32175	84996	2026.6	784	16.2
1993	26	13.3	8.9	2.9	5.4	0.9	8	2.8	33173	85344	2577.4	921.6	16.5
1994	24.3	12.6	9.7	3	5.3	0.7	8.5	3	34169	85681	3496.2	1221	16.8
1995	23.6	13.1	9.7	3.2	4.6	0.6	9.2	3.4	35174	85947	4283	1577.7	17.2
1996	25.8	14.8	9.6	3.4	4.8	0.8	9.25	3.7	37304	85085	4838.9	1926.1	18.4
1997	25.5	15.1	11.1	4.1	5.1	1	9.3	3.8	39449	84177	5160.3	2090.1	19.6
1998	25.5	15.5	10.2	4.1	6.2	0.9	9.84	3.7	41608	83153	5425.1	2162	20.8
1999	26.7	16.4	10.9	4.3	7.9	1	10.3	3.8	43748	82038	5854	2210.3	22
2000	25.4	18.3	11.2	4.8	9.9	1.1	11.7	3.9	45906	80837	6280	2253.4	23.2
2001	26.5	18.2	10.4	4.7	11.9	1.2	10.33	4.1	48064	79563	6859.6	2366.4	24.4
2002	32.5	18.6	10.6	4.7	15.7	1.2	13.2	4.4	50212	78241	7702.8	2475.6	25.8
2003	32.9	19.7	11.2	4.8	18.6	1.7	13.4	4.7	52376	76851	8472.2	2622.2	27.2
2004	29.3	19.2	10.4	4.6	18.8	2	12.5	4.5	54283	75705	9421.6	2936.4	28.9
2005	32.9	22.4	10.4	4.7	17.9	2.9	12.6	4.9	56212	74544	10493	3254.9	30.7
2006	32.1	22.3	10.4	5	18.3	3.1	13	5	58288	73160	11759.5	3587	32.5
2007	31.8	20.5	10.3	4.7	17.8	3.5	14.2	5.4	60633	71496	13785.8	4140.4	34.3
2008	31.2	20.2	10.7	5.4	15.2	3.4	11.9	5.2	62403	70399	15780.8	4760.6	36
2009	34.7	21.5	10.6	5.3	14.9	3.6	12.2	5.3	64512	68938	17174.7	5153.2	37.7
2010	34.7	22.2	10	5.1	14	3.6	15.2	5.2	66978	67113	19109.4	5919	38.8
2011	35.2	23.3	10.1	5.4	13.7	5.2	14.6	5.4	69079	65656	21809.8	6977.3	40.6
2012	35.7	23.5	10.5	5.9	14	5.3	15.2	5.4	71182	64222	24564.7	7916.6	42.4

Note: (1) unit: per capita consumption in kilograms; (2) the data are from *Rural China Statistical Yearbook*.

Table 2: The grey correlation analysis about each influencing factor in 1981–2007 of urban and rural feed grain demand.

Influencing factor	Urban		Rural	
	Correlation degree	Relational order	Correlation degree	Relational order
(Urban/rural) population	0.9370	1	0.9255	2
Urbanization level	0.9047	2	0.9641	1
(Urban/rural) per capita income	0.7236	3	0.6881	3

Table 3: Predicted value of various influencing factors in 2008.

Influencing factor	Model	Adjusted R^2	Predicted value in 2008
Urban population	ARIMA(3, 2, 6)	0.956	62965.45
Urbanization level	ARIMA(7, 2, 2)	0.848	36.1254
Urban residents per capita income	ARIMA(4, 2, 5)	0.876	15872.19

Table 4: The comparison between the actual value and predicted value of feed grain demand under different prediction models (unit: ten thousand tons).

	Year	Actual value	Predicted value	Relative error	Mean relative error
Combined dynamic forecasting model	2008	16332.1	16470.9	0.8%	0.46%
	2009	17665.2	17795.3	0.7%	
	2010	17981.0	17928.4	0.2%	
	2011	18687.2	18798.5	0.5%	
	2012	19267.6	19243.8	0.1%	
Grey forecasting model	2008	16332.1	17925.4	9.7%	6.4%
	2009	17665.2	18426.9	4.3%	
	2010	17981.0	19246.7	7.0%	
	2011	18687.2	19875.1	6.4%	
	2012	19267.6	20144.8	4.6%	

so it has become a necessary work to research the feed grain demand deeply for ensuring food security.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research was financially supported by the National Food Industry Commonweal Special Scientific Research Projects (no. 201413001).

References

[1] T. Weiming and J. Chudleigh, *China's Feed Grain Market :development and Prospect*, AARC Working Paper Series, 1998.

[2] X. Yu and D. Abler, "The demand for food quality in rural China," *American Journal of Agricultural Economics*, vol. 91, no. 1, pp. 57–69, 2009.

[3] L. R. Brown, *Who Will Feed China? Wake up Call for a Small Planet*, World Watch Norton and Co, New York, NY, USA, 1995.

[4] M. Gao, Q. Luo, Y. Liu, and J. Mi, "Grain consumption forecasting in China for 2030 and 2050: volume and varieties," in *Proceedings of the 3rd International Conference on Agro-Geoinformatics (Agro-Geoinformatics '14)*, pp. 1–6, Beijing, China, August 2014.

[5] X. Xin, W. Tian, and Z. Zhou, "Changing patterns of feed grain production and marketing in China," Agribusiness Perspectives Paper 47, 2001.

[6] N. Minot and F. Goletti, "Rice market liberalization and poverty in Vietnam," IFPRI Research Report 114, IFPRI, Washington, DC, USA, 2000.

[7] M. Hao and L. Xiang, "Grey relational analysis for impact factors of micro-milling surface roughness," in *Proceedings of the 12th IEEE International Conference on Electronic Measurement & Instruments (ICEMI '15)*, pp. 109–113, IEEE, Qingdao, China, July 2015.

[8] R. Sallehuddin, S. M. H. Shamsuddin, and S. Z. Mohd Hashim, "Application of grey relational analysis for multivariate time series," in *Proceedings of the 8th International Conference on Intelligent Systems Design and Applications (ISDA '08)*, pp. 432–437, Kaohsiung, Taiwan, November 2008.

[9] Q. Wang, F. Xia, and X. Wang, "Integration of grey model and multiple regression model to predict energy consumption," in *Proceedings of the International Conference on Energy and Environment Technology (ICEET '09)*, pp. 194–197, Guilin, China, October 2009.

[10] T. Yang, N. Yang, and C. Zhu, "Investigation of grain output prediction based on ARIMA model," *Journal of Henan University of Technology (Natural Science Edition)*, vol. 36, no. 5, pp. 24–27, 2015.

[11] K. K. Suresh and S. R. Krishna Priya, "Forecasting sugarcane yield of Tamilnadu using ARIMA models," *Sugar Tech*, vol. 13, no. 1, pp. 23–26, 2011.

[12] X. Xu, L. Cao, J. Zhou, and F. Su, "Study and application of grain yield forecasting model," in *Proceedings of the 4th International Conference on Computer Science and Network Technology (ICCSNT '15)*, pp. 652–656, Harbin, China, December 2015.

[13] National Bureau of Statistics of China, *China Rural Statistical Yearbook*, China Statistics Press, Beijing, China, 2014.

[14] J. van Zyl, *Grain Worth Gold: "Feeding Animals Increases Demand and Pushes up Prices"*, Business Strategy, 2007.

[15] F. Qin and X. Chen, *Chinese Farmers' Food Consumption Research*, China Agriculture Press, Beijing, China, 2007.

[16] M. Li and H. Lei, *Study on China's food security in the new situation [Ph.D. thesis]*, Huazhong Agricultural University, Wuhan, China, 2005.

[17] C. Aubert, "Food security and consumption patterns in China: the grain problem," *China Perspectives*, vol. 2008, no. 2, pp. 5–23, 2014.

[18] X. Yu and D. Abler, "The demand for food quality in Rural China," *American Journal of Agricultural Economics*, vol. 91, no. 1, pp. 57–69, 2009.

Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2016, Article ID 6459873, 7 pages
http://dx.doi.org/10.1155/2016/6459873

Research Article

A Long-Term Prediction Model of Beijing Haze Episodes Using Time Series Analysis

Xiaoping Yang,[1] Zhongxia Zhang,[1] Zhongqiu Zhang,[2] Liren Sun,[1] Cui Xu,[1] and Li Yu[1]

[1]*School of Information, Renmin University of China, Beijing 100872, China*
[2]*School of Computer Science, Northeastern University, Shenyang 110819, China*

Correspondence should be addressed to Li Yu; buaayuli@ruc.edu.cn

Received 20 April 2016; Revised 28 June 2016; Accepted 10 July 2016

Academic Editor: Francisco Martínez-Álvarez

The rapid industrial development has led to the intermittent outbreak of pm2.5 or haze in developing countries, which has brought about great environmental issues, especially in big cities such as Beijing and New Delhi. We investigated the factors and mechanisms of haze change and present a long-term prediction model of Beijing haze episodes using time series analysis. We construct a dynamic structural measurement model of daily haze increment and reduce the model to a vector autoregressive model. Typical case studies on 886 continuous days indicate that our model performs very well on next day's Air Quality Index (AQI) prediction, and in severely polluted cases (AQI $\geq$ 300) the accuracy rate of AQI prediction even reaches up to 87.8%. The experiment of one-week prediction shows that our model has excellent sensitivity when a sudden haze burst or dissipation happens, which results in good long-term stability on the accuracy of the next 3–7 days' AQI prediction.

1. Introduction

Industry of developing countries is mainly centralized around big cities, accompanied by a large population, consumption, and pollution. Together with Tianjin city and Hebei province, Northern China has become one of the most prosperous and polluted areas on Earth. By 2013, the transient population of Beijing was 37.5 million, and the intermittent outbreak of air pollution has greatly impacted every citizen's life: physiological diseases [1, 2], depression, and poor visibility in traffic [3, 4]. The main component of haze is pm2.5 (particulate matters less than 2.5 μm in aerodynamic diameter), and the concentration of pollution is described with Air Quality Index (AQI, the concentration of pm2.5). The Chinese Government began to monitor and record pm2.5 concentrations for major cities since 2013 [5]. According to the report of Quan et al. [6], the AQI reached 600 in Beijing during the haze event in January 2013. In recent years, more and more papers have referred to the haze episodes and the consequences in Northern China [7–11]. Researchers pointed out that, over the coming years, haze episodes would continue to burst frequently in Northern China [12].

This paper presents an AQI prediction model of Beijing based on time series analysis. We collected Beijing's AQI data of 29 continuous months since 2013 and constructed a dynamic structural prediction model. Statistical methods are used to obtain the maximum likelihood estimation of the prediction model. And both short-term and long-term experiments are carried out to test the accuracy and robustness of our model.

The remainder of this paper is organized as follows. In Section 2, we introduce recent related work. Section 3 presents our prediction model and proves our model to be a vector autoregressive model. Experiments and evaluations are reported in Section 4. We conclude the paper in Section 5 with future works.

2. Related Work

Generally, pm2.5, or haze, is born mainly through anthropogenic factors [13–16] and eliminated by natural diffusion. Several days after emission, secondary pm2.5 is produced through photochemical reactions among indiffusible pollutants. Secondary pm2.5 is the principal component in most

severe haze episodes in China [17]. A typical way of haze prediction is to use pollutant emission data (CO, SO_2, and NO_x) in the simulation [5, 18]. Huang et al. [14] analyzed the chemical compositions of pm2.5 and used chemical mass balance to identify the emission sources. Other more complex models are proposed to introduce the atmospheric features, chemistry components, and transport factors [15]. But the more common case is that pollutant emission data usually increase or decrease synchronously with AQI. Sun [19] took population, car ownership, and GDP into consideration and proposed a statistical index system of average annual haze episode days. They found that although most factors contribute to predicting pm2.5, the annual average of NO_x is negatively correlated with average severely polluted days. The paper [12] established a cubic exponential smoothing model by introducing dust emission into haze prediction. Liang et al. pointed out that there are various distribution and transmission patterns of pm2.5 [20]. In fact, Wang et al. mentioned that the government control policy should be considered in model simulations [9].

Many researches use backpropagation neural network as the simulation model [19, 21]. Statistical time series analysis is rarely used in haze prediction, so long-term haze prediction is difficult for current methods to accomplish [22]. Multiple linear regression models also perform well on daily scale prediction [23, 24]. However, the test data of existing researches is not ample; for example, [21] tested the prediction accuracy on only 3 days. Besides, Zhang et al. pointed out that pm2.5 accumulation in previous days significantly affects the present daily pm2.5 concentration, which should also be a concern in the modeling process [22].

Considering the above points, this paper presents a new AQI prediction model integrated with natural factor, humanity factor, and self-evolution factor.

3. The Prediction Model of Beijing's Daily AQI

3.1. The Parameters and Architecture of the Prediction Model. The change of daily pm2.5 concentration depends on two factors: daily overall production of pm2.5 by human activities P_t and daily overall natural diffusion or overall natural accumulation of pm2.5 C_t. The production of haze depends a lot on the control policies of the government toward the emission of industry fuels I_t. The diffusion of haze mainly depends on the airflow W_t. Besides, complex chemical changes could occur between pm2.5 and other pollutants; thus, previous day's pm2.5 concentration also affects the AQI, which could be seen as the evolution result of previous day's pm2.5 and is represented by Y_t. Apparently, $P_t - C_t$ could be directly observed. P_t is generated by a semimanual method. P_t is mainly related to daily human activities, and we calculate P_t from AQI sequences of no less than five consecutive sunny and windless days. Special circumstances are also considered. In winter, P_t will be larger because the heating system is on. The car usage restrictions and temporary stoppage of factories during Beijing APEC 2014 are also taken into consideration. C_t is then calculated as $P_t - (P_t - C_t)$. Sometimes, C_t is greater than zero, which means pm2.5 accumulates because of nonhuman factors.

Thus, the daily net growth of pm2.5 ($P_t - C_t$) is a function of the evolution result Y_t, the industry control index I_t, and the forecast of wind power W_t. Consider this problem as a dynamic structural model, and our model can be described as

$$P_t - C_t = \beta_0 + \beta_1 Y_t + \beta_2 W_t + \beta_3 I_t + \beta_4 \left(P_{t-1} - C_{t-1}\right) + \mu_t^D. \tag{1}$$

Parameters β_1, β_2, and β_3, respectively, represent the effect caused by the pm2.5 of the previous day, the wind power, and the industry control index. The net growth of previous day's pm2.5 partly affects present day's pm2.5 and partly affects the next day's pm2.5. The parameter β_4 represents this "partial adjustment." The disturbance μ_t^D represents other factors which affect present day's pm2.5.

3.2. Complexity Reduction of the Prediction Model. In order to facilitate the research and modeling process, we have proved that this model could be reduced to a vector autoregressive model.

Proposition 1. *Formula (1) is a vector autoregressive model.*

Proof. Assume that there exists sequence autocorrelation in formula (1). The autocorrelation is

$$\mu_t^D = \rho \mu_{t-1}^D + v_t^D \tag{2}$$

in which v_t^D is white noise. Here, we apply the Cochrane-Orcutt iteration to rewrite formula (2):

$$\left(1 - \rho L\right) \mu_t^D = v_t^D, \tag{3}$$

where L is the lag operator ($LV_t \equiv V_{t-1}$), which can convert the last phase to current value in a time series.

The next work is to find the most satisfying value of ρ through successive iteration method. Specifically, this method uses residual error to estimate the unknown ρ.

Assume that we use previous p days' AQI to predict present day's AQI. Multiply $(1 - \rho L)$ on both sides of formula (1); the expansion formula will be as follows:

$$\begin{aligned} P_t = {} & k_1 + \beta_{12}^0 C_t + \beta_{13}^0 I_t + \beta_{14}^0 Y_t + \beta_{15}^0 W_t + \beta_{11}^1 P_{t-1} \\ & + \beta_{12}^1 C_{t-1} + \beta_{13}^1 I_{t-1} + \beta_{14}^1 Y_{t-1} + \beta_{15}^1 W_{t-1} \\ & + \beta_{11}^2 P_{t-2} + \beta_{12}^2 C_{t-2} + \beta_{13}^2 I_{t-2} + \beta_{14}^2 Y_{t-2} \\ & + \beta_{15}^2 W_{t-2} + \cdots + \beta_{11}^p P_{t-p} + \beta_{11}^p C_{t-p} + \beta_{11}^p I_{t-p} \\ & + \beta_{11}^p Y_{t-p} + \beta_{11}^p W_{t-p} + v_t^D. \end{aligned} \tag{4}$$

In the substitution process, many assumptions are neglected. But the ordinary least square method (OLS estimation) should not be used in the estimation of formula (4), because OLS can only illustrate the relationship between daily pm2.5 production and the policy control index, the accumulation of history pm2.5, and the wind power. The net

growth of previous day's pm2.5 is only one reason of the correlation of these variables.

The government could make policies to control pm2.5 production of industry to obtain "satisfying" daily production of pm2.5; that is, I_t is an endogenous variable. And the policy control index depends on present day's and previous p days' accumulation of history pm2.5, the wind power, the daily production of pm2.5, and daily diffusion of pm2.5:

$$\begin{aligned} I_t = {} & k_3 + \beta_{31}^0 P_t + \beta_{32}^0 C_t + \beta_{33}^0 Y_t + \beta_{34}^0 W_t + \beta_{31}^1 P_{t-1} \\ & + \beta_{32}^1 C_{t-1} + \beta_{33}^1 Y_{t-1} + \beta_{34}^1 W_{t-1} + \beta_{35}^1 I_{t-1} \\ & + \beta_{31}^2 P_{t-2} + \beta_{32}^2 C_{t-2} + \beta_{33}^2 Y_{t-2} + \beta_{34}^2 W_{t-2} \\ & + \beta_{35}^2 I_{t-2} + \cdots + \beta_{31}^p P_{t-p} + \beta_{32}^P C_{t-p} + \beta_{33}^P Y_{t-p} \\ & + \beta_{34}^P W_{t-p} + \beta_{35}^P I_{t-p} + \upsilon_t^C, \end{aligned} \tag{5}$$

where υ_t^C represents the influence brought about by other policies.

The net growths of previous days' pm2.5 and policy control index also have an effect on daily accumulation of pm2.5:

$$\begin{aligned} Y_t = {} & k_4 + \beta_{41}^0 P_t + \beta_{42}^0 C_t + \beta_{43}^0 I_t + \beta_{44}^0 W_t + \beta_{41}^1 P_{t-1} \\ & + \beta_{42}^1 C_{t-1} + \beta_{43}^1 I_{t-1} + \beta_{44}^1 W_{t-1} + \beta_{45}^1 Y_{t-1} \\ & + \beta_{41}^2 P_{t-2} + \beta_{42}^2 C_{t-2} + \beta_{43}^2 I_{t-2} + \beta_{44}^2 W_{t-2} \\ & + \beta_{45}^2 Y_{t-2} + \cdots + \beta_{41}^p P_{t-p} + \beta_{42}^p C_{t-p} + \beta_{43}^p I_{t-p} \\ & + \beta_{44}^p W_{t-p} + \beta_{45}^p Y_{t-p} + \upsilon_t^A, \end{aligned} \tag{6}$$

where υ_t^A represents other factors that influence daily accumulation of pm2.5.

Analogized from formulas (4), (5), and (6), C_t and W_t can both be written in a similar form. Join formulas (4), (5), and (6) together, and rewrite them into vector form:

$$B_0 x_t = K + B_1 x_{t-1} + B_2 x_{t-2} + \cdots + B_p x_{t-p} + \upsilon_t \tag{7}$$

in which

$$\begin{aligned} x(t) &= \left(P_t, C_t, I_t, Y_t, W_t\right)^T, \\ \upsilon_t &= \left(\upsilon_t^D, \upsilon_t^S, \upsilon_t^C, \upsilon_t^A, \upsilon_t^H\right)^T, \\ K &= \left(k_1, k_2, k_3, k_4, k_5\right), \\ B_0 &= \begin{bmatrix} 1 & -\beta_{12}^0 & -\beta_{13}^0 & -\beta_{14}^0 & -\beta_{15}^0 \\ -\beta_{21}^0 & 1 & -\beta_{23}^0 & -\beta_{24}^0 & -\beta_{25}^0 \\ -\beta_{31}^0 & -\beta_{32}^0 & 1 & -\beta_{34}^0 & -\beta_{35}^0 \\ -\beta_{41}^0 & -\beta_{42}^0 & -\beta_{43}^0 & 1 & -\beta_{45}^0 \\ -\beta_{51}^0 & -\beta_{52}^0 & -\beta_{53}^0 & -\beta_{54}^0 & 1 \end{bmatrix}. \end{aligned} \tag{8}$$

In B_0, the parameters in the 1st, 2nd, 3rd, 4th, and 5th row, respectively, relate P_t, C_t, I_t, Y_t, and W_t to the other variables. Every B_s is a $5 * 5$ matrix. Premultiply formula (7) by B_0^{-1} (the inverse matrix of B_0):

$$x_t = c + \Psi_1 x_{t-1} + \Psi_2 x_{t-2} + \cdots + \Psi_p x_{t-p} + \varepsilon_t \tag{9}$$

in which

$$\begin{aligned} c &= B_0^{-1} K, \\ \Psi_s &= B_0^{-1} B_s, \\ \varepsilon_t &= B_0^{-1} \upsilon_t. \end{aligned} \tag{10}$$

This is the standard form of vector autoregressive model. So it is proved that our prediction model (formula (1)) is in fact a vector autoregressive model. □

The regression parameters of our haze prediction model can be obtained as follows.

Let

$$\begin{aligned} -\Gamma &= \begin{bmatrix} K & B_1 & B_2 & \cdots & B_p \end{bmatrix}, \\ y_t &= \begin{bmatrix} 1 & x_{t-1} & x_{t-2} & \cdots & x_{t-p} \end{bmatrix}^T. \end{aligned} \tag{11}$$

The dynamic structural system (formula (7)) isss

$$B_0 x_t = -\Gamma x_t + \upsilon_t. \tag{12}$$

Assume that the disturbance terms are not sequence correlated or correlated to each other, which means

$$E\left(\upsilon_t \upsilon_\tau^T\right) = \begin{cases} D, & t = \tau, \\ 0, & t \neq \tau. \end{cases} \tag{13}$$

D is a main diagonal matrix. Formula (12) could be written as

$$x_t = \Phi^T y_t + \varepsilon_t \tag{14}$$

in which

$$\begin{aligned} \Phi^T &= B_0^{-1} - \Gamma, \\ \varepsilon_t &= B_0^{-1} \upsilon_t. \end{aligned} \tag{15}$$

Let Ω be the variance-covariance matrix of ε_t:

$$\Omega = E\left(\varepsilon_t \varepsilon_t^T\right) = B_0^{-1} E\left(\upsilon_t \upsilon_t^T\right) \left(B_0^{-1}\right)^T = B_0^{-1} D \left(B_0^{-1}\right)^T. \tag{16}$$

Suppose B_0 is a lower triangular matrix, in which all main diagonal elements are assigned 1, and D is a main diagonal matrix. The parameters (B_0, Γ, D) can be obtained through the maximum likelihood estimation of complete information. The maximum likelihood estimation of Ω can be obtained by the variance-covariance matrix of regression residual.

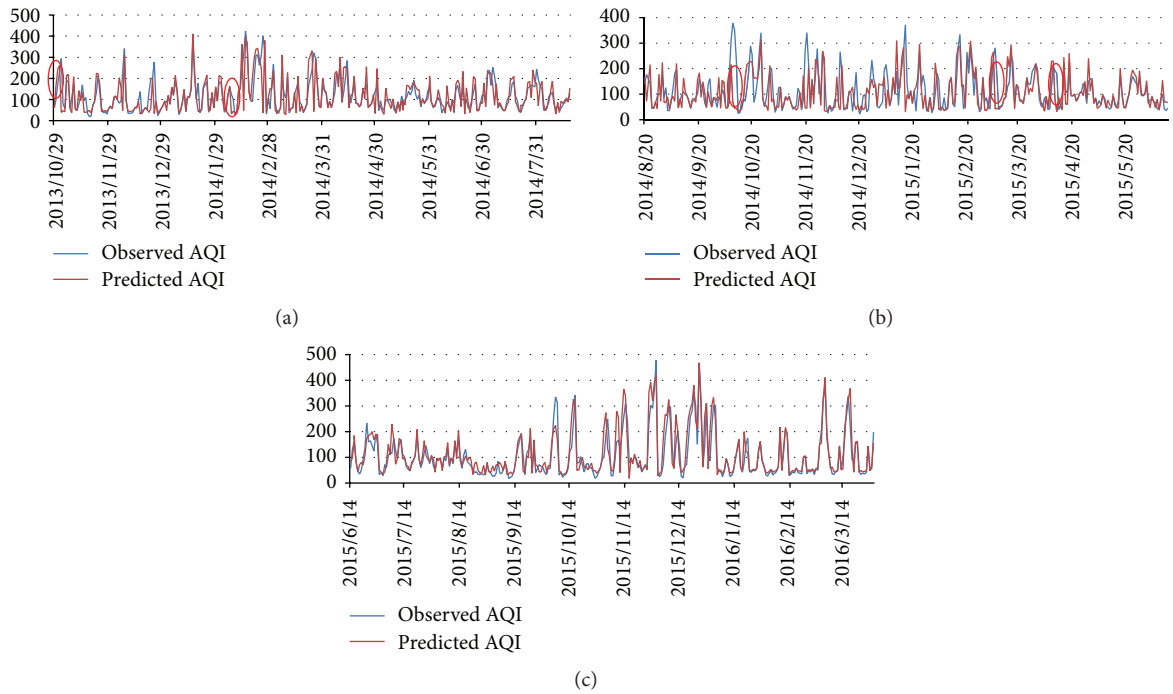


Figure 1: (a, b, c) Next day's AQI prediction on 886 continuous days.

Finally, $\widehat{B}_0^{-1}$ and D are calculated through triangular decomposition of $\widehat{\Omega}$; thus, Γ can be evaluated.

Above all, the prediction model of Beijing AQI has considered factors including industry emission and policy control, together with the chemical changes of previous days' pollution accumulation and the diffusion conditions. This model also takes the correlations between these factors into consideration and introduces time series haze features into the dynamic structural model. The policy control index is simulated by the record of 4 severe haze episodes during this period. The diffusion is evaluated by weather record of daily wind power.

4. Model Evaluations

We collected the daily AQI and daily weather information from 28 Oct. 2013 to 31 Mar. 2016. This complete sequence is used to test the accuracy of the prediction model. The next day's AQI prediction experiment (Section 4.1) and long-term AQI prediction experiment (Section 4.2) are both implemented. The next day's AQI prediction is evaluated from two perspectives: the accuracy of daily prediction and the accuracy of statistical analysis.

4.1. Next Day's AQI Prediction. The next day's weather forecast information is applied in next day's AQI prediction. The observed and predicted daily mean AQI in Beijing are illustrated in Figure 1. The simulation result shows that the predicted value matched the observed value very well on the whole sequence of 886 days. Sometimes, there is severe deviation from the observed value; for example, on 19 Feb. 2014, the observed AQI was 89, while our model gives a prediction of 209, with an offset of 135%. But the fact is, in the afternoon of 19 Feb., the wind of Beijing suddenly changed from northeasterly to southwesterly, and by 19:00 the AQI has reached already up to 170, which could be interpreted as our model successfully forecasted a severe haze outbreak several hours in advance; in the coming 7 days, the average daily AQI of Beijing is 305. The occasional occurrence of this "foreseeing" phenomenon is caused by coarse time granularity (daily), and this phenomenon is marked with red ellipse in Figure 1. These marks indicate that our model could "foresee" the sharp change of both outbreaks and diffusions. Most haze outbreaks and diffusions could be accurately simulated; some could be foreseen but could never be delayed.

Figure 2(a) is a scatter diagram of daily AQI, including both observed value and predicted value. Most points lie close to $y = x$ (the red line). But some points lie in a queue at the bottom part, which means the observed AQI exceeds 200, while the predicted value is less than 50. There are altogether 15 such outliers, 7 of which "foresee" haze diffusion, while the other 8 bug points could not be well interpreted. All the 15 points are checked and listed in Table 1. "√" means a "foreseeing" phenomenon, and "?" represents bug points. Figure 2(b) is a scatter diagram of annual AQI (sum of daily AQI in a certain year). Our data covers only 2 months of 2013 and 3 months of 2016, so, in this diagram, these 2 points lie in the lower left corner.

The pie chart in Figure 3 shows the distribution of prediction accuracy. The deviation of predicted and observed AQI is obtained through the following formula:

$$\text{Dev}_t = \frac{|\text{PredictedValue} - \text{ObservedValue}|}{\text{ObservedValue}} \times 100\%. \quad (17)$$

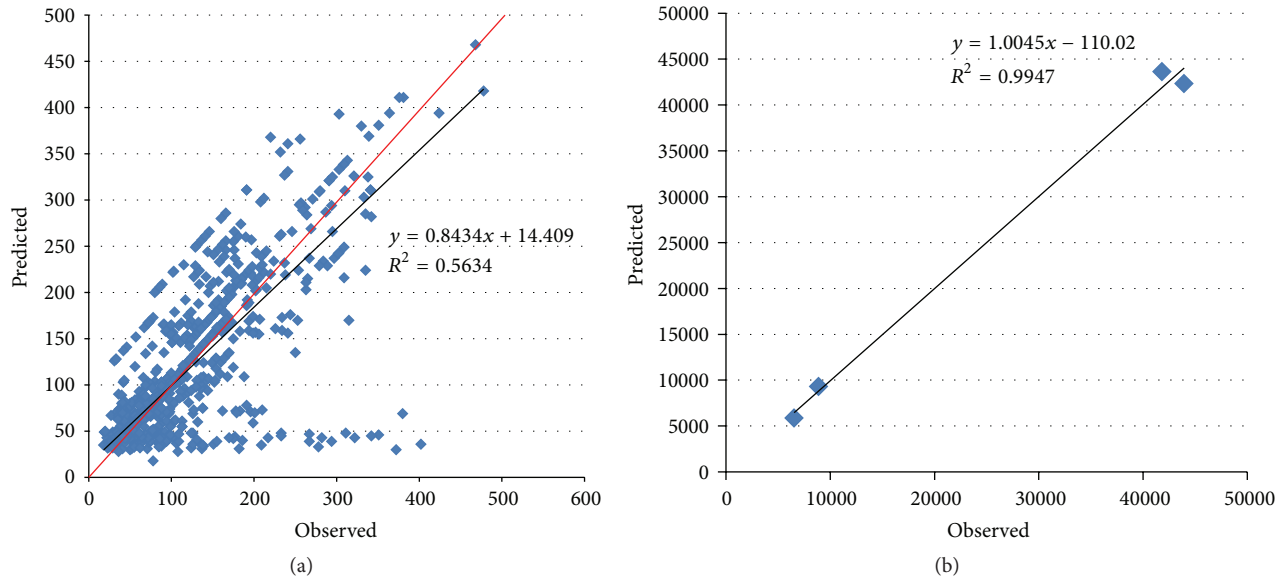


Figure 2: (a) Daily AQI of the 886 days. (b) Annual AQI from 2013 to 2016.

Table 1: All the 15 outliers in Figure 2(a).

Date of outlier	Label
Nov. 2, 2013	√
Dec. 7, 2013	?
Dec. 25, 2013	√
Feb. 14, 2014	?
Feb. 25, 2014	?
Mar. 26, 2014	?
Oct. 10, 2014	√
Oct. 11, 2014	√
Nov. 19, 2014	?
Nov. 20, 2014	?
Nov. 30, 2014	√
Dec. 9, 2014	?
Jan. 4, 2015	?
Jan. 15, 2015	√
Mar. 7, 2015	√

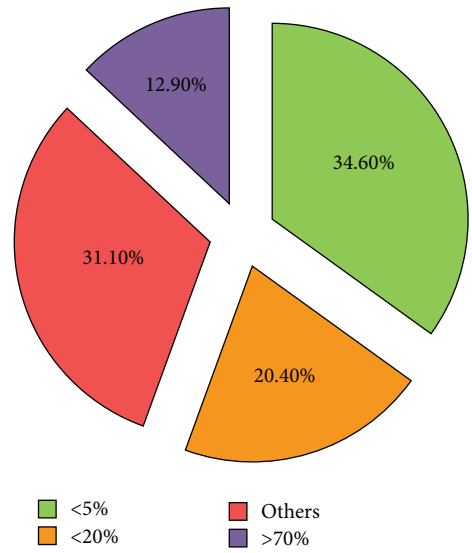


Figure 3: The deviation of predicted and observed AQI.

Figure 3 shows that 55% predictions match the observed values very well (<20% deviation). The purple part is mainly caused by the "foreseeing" phenomenon. Most samples of the red part come from less-polluted days. For example, on 12 Jan. 2016, the AQI prediction is 40 while the observed AQI is 29, which makes a deviation of 37.9%. In fact, statistics also indicate that our model performs better in worse air conditions (Figure 4). A sample is correctly predicted if the deviation of a sample is less than 20% or the predicted air quality level matches the observed level.

4.2. Long-Term AQI Prediction. In the long-term prediction, we use history haze data sequence and weather forecast information to predict the next 7 days' AQI. A sample is correctly predicted if the deviation of a sample is less than 20% or the predicted air quality level matches the observed level. From 26 Dec. 2015 to 31 Mar. 2016, we predict the AQI in the next 7 days and check the accuracy of n-day predictions. Figure 5 shows the accuracy of long-term prediction in the 91 days' experiment. Figure 5 shows that the accuracy stays stable on the next 3, 4, 5, 6, and 7 days' AQI prediction, which indicates that our model has excellent robustness on the task of long-term prediction. The next day's prediction accuracy surprisingly reaches 79.1%, which is far better than the experiment in Section 4.1. The reason is that, during the 91 days, 6 haze episodes attacked Beijing. These frequent attacks did contribute a lot to the overall performance because our model is very sensitive to sudden changes of AQI, including

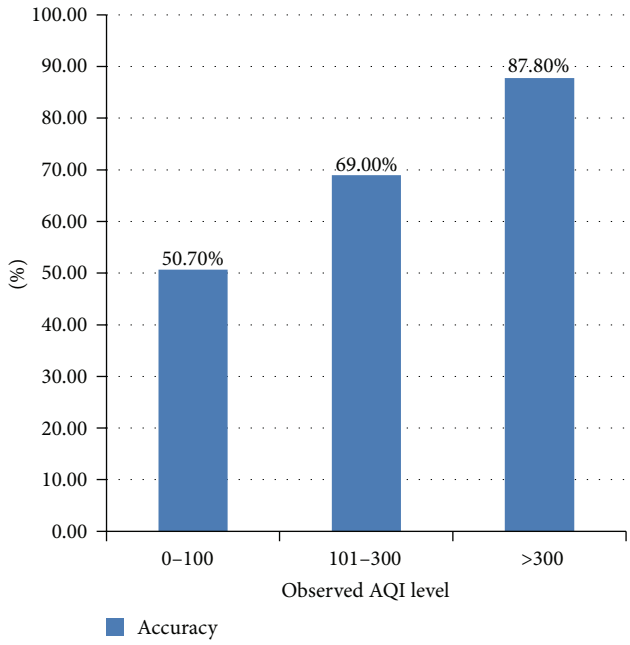


Figure 4: Prediction accuracies of different air qualities.

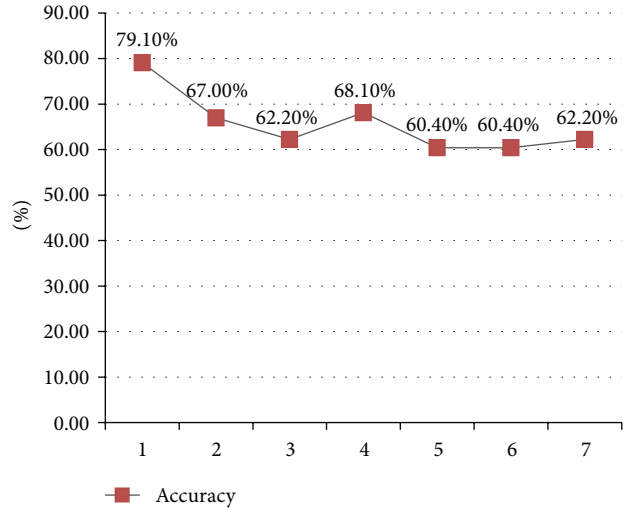


Figure 5: The accuracy of long-term AQI prediction.

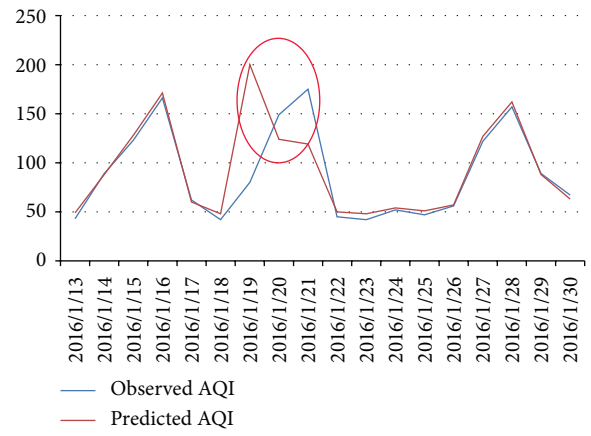


Figure 6: Three haze episodes in Jan. 2016.

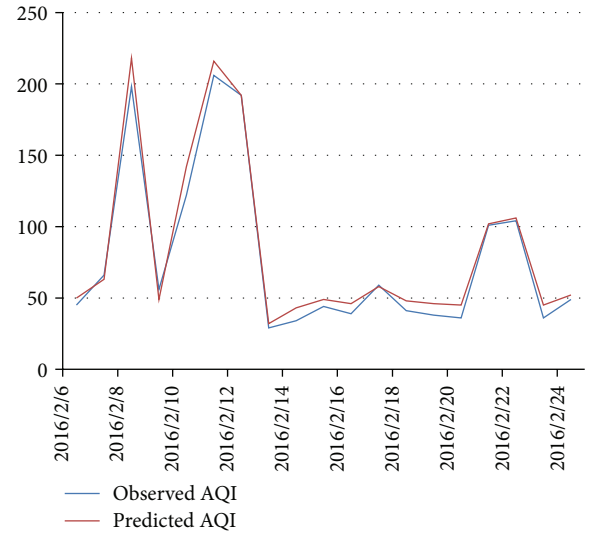


Figure 7: Three haze episodes in Feb. 2016.

outbreaks and diffusions (Section 4.1; Figure 4). Figures 6 and 7 show several haze episodes during the 91 days. Both figures show a pm2.5 change process of more than 2 weeks. Figure 6 also shows a "foreseeing" phenomenon caused by coarse time granularity, marked by a red ellipse.

5. Conclusion and Future Work

This paper presented a dynamic structural model to predict Beijing's daily AQI. This model integrated natural factor, humanity factor, and self-evolution factor into the time series model. This dynamic structural measurement model of daily haze increment is proven to be a vector autoregressive model. Experiments reflected two highlights of this model. First, our model is very sensitive to and performs very well on predicting sudden changes of AQI, including both outbreaks and diffusions. Second, the model has great robustness on the task of long-term AQI prediction. Lastly, limited by the coarse time granularity, our model sometimes "foresees" but never delays or misses any sudden changes of haze.

Many researchers use simple backpropagation neural network to accomplish nonlinear prediction models. But since methods based on time series are proven to be effective in haze prediction modeling, we believe that recurrent neural networks give better performances in such a prediction task. Although the related factors are limited in existing models, the overfitting problem should still be concerned, because, in long-term prediction, a deviation could spread and be exaggerated in the following days' predictions.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grant 71271209, Beijing Municipal Natural Science Foundation under Grant 4132052, and Humanity and Social Science Youth Foundation of Ministry of Education of China under Grant 11YJC630268.

References

[1] L. Gao, G. Jia, R. Zhang et al., "Visual range trends in the Yangtze River Delta Region of China, 1981–2005," *Journal of the Air and Waste Management Association*, vol. 61, no. 8, pp. 843–849, 2011.

[2] Q. Liu, J. Baumgartner, Y. Zhang, and J. J. Schauer, "Source apportionment of Beijing air pollution during a severe winter haze event and associated pro-inflammatory responses in lung epithelial cells," *Atmospheric Environment*, vol. 126, pp. 28–35, 2016.

[3] X. Tie, G. P. Brasseur, C. Zhao et al., "Chemical characterization of air pollution in Eastern China and the Eastern United States," *Atmospheric Environment*, vol. 40, no. 14, pp. 2607–2625, 2006.

[4] J. Deng, T. Wang, Z. Jiang et al., "Characterization of visibility and its affecting factors over Nanjing, China," *Atmospheric Research*, vol. 101, no. 3, pp. 681–691, 2011.

[5] MEPC (Ministry of Environmental Protection of China), *China Environment Yearbook*, China Environment Yearbook Press, Beijing, China, 2014.

[6] J. Quan, X. Tie, Q. Zhang et al., "Characteristics of heavy aerosol pollution during the 2012-2013 winter in Beijing, China," *Atmospheric Environment*, vol. 88, pp. 83–89, 2014.

[7] X. J. Zhao, P. S. Zhao, J. Xu et al., "Analysis of a winter regional haze event and its formation mechanism in the North China Plain," *Atmospheric Chemistry & Physics*, vol. 13, no. 11, pp. 5685–5696, 2013.

[8] H. Wang, J. An, L. Shen et al., "Mechanism for the formation and microphysical characteristics of submicron aerosol during heavy haze pollution episode in the Yangtze River Delta, China," *Science of the Total Environment*, vol. 490, pp. 501–508, 2014.

[9] Y. Wang, L. Li, C. Chen et al., "Source apportionment of fine particulate matter during autumn haze episodes in Shanghai, China," *Journal of Geophysical Research Atmospheres*, vol. 119, no. 4, pp. 1903–1914, 2014.

[10] D. Ji, L. Li, Y. Wang et al., "The heaviest particulate air-pollution episodes occurred in northern China in January, 2013: insights gained from observation," *Atmospheric Environment*, vol. 92, pp. 546–556, 2014.

[11] J. Hu, Y. Wang, Q. Ying, and H. Zhang, "Spatial and temporal variability of PM2.5 and PM10 over the North China Plain and the Yangtze River Delta, China," *Atmospheric Environment*, vol. 95, pp. 598–609, 2014.

[12] Q. Hou and H. Yang, "Analysis and forecasting of haze weather based on the cubic exponential smoothing model," *Environmental Protection Science*, vol. 6, pp. 73–77, 2014.

[13] S. Guo, M. Hu, M. L. Zamora et al., "Elucidating severe urban haze formation in China," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 111, no. 49, pp. 17373–17378, 2014.

[14] R.-J. Huang, Y. Zhang, C. Bozzetti et al., "High secondary aerosol contribution to particulate pollution during haze events in China," *Nature*, vol. 514, no. 7521, pp. 218–222, 2015.

[15] M. Vieno, M. Heal R, S. Hallsworth et al., "The role of long-range transport and domestic emissions in determining atmospheric secondary inorganic particle concentrations across the UK," *Atmospheric Chemistry & Physics*, vol. 14, pp. 8435–8447, 2014.

[16] G. Kiesewetter, J. Borkenkleefeld, W. Schöpp et al., "Modelling street level PM 10 concentrations across Europe: source apportionment and possible futures," *Atmospheric Chemistry & Physics*, vol. 14, pp. 18315–18354, 2014.

[17] R. Zhang, L. Wang, A. F. Khalizov et al., "Formation of nanoparticles of blue haze enhanced by anthropogenic pollution," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 42, pp. 17650–17654, 2009.

[18] J. M. Wooldridge, *Econometric Analysis of Cross Section and Panel Data*, MIT Press, Cambridge, Mass, USA, 2nd edition, 2010.

[19] Y. Sun, *Research on the Forecast of Haze Days Based on the Principal Component-BP Neural Network Model*, Jiangsu University, 2015.

[20] X. Liang, T. Zou, B. Guo et al., "Assessing Beijing's PM2.5 pollution: severity, weather impact, APEC and winter heating," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 471, no. 2182, 2015.

[21] H. Ai and Y. Shi, "Study on prediction of haze based on BP neural network," *Computer Simulation*, vol. 32, pp. 402–405, 2015.

[22] X. Zhang, Y. Wu, and B. Gu, "Characterization of haze episodes and factors contributing to their formation using a panel model," *Chemosphere*, vol. 149, pp. 320–327, 2016.

[23] L. Li, J. Qian, C.-Q. Ou, Y.-X. Zhou, C. Guo, and Y. Guo, "Spatial and temporal analysis of Air Pollution Index and its timescale-dependent relationship with meteorological factors in Guangzhou, China, 2001–2011," *Environmental Pollution*, vol. 190, pp. 75–81, 2014.

[24] G. Tian, Z. Qiao, and X. Xu, "Characteristics of particulate matter (PM10) and its relationship with meteorological factors during 2001-2012 in Beijing," *Environmental Pollution*, vol. 192, pp. 266–274, 2014.