IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 19, 2019, accepted December 31, 2019, date of publication January 6, 2020, date of current version January 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964067

An Efficient Network Method for Time Series Forecasting Based on the DC Algorithm and Visibility Relation

JUNYIN ZHAO[1], HONGMING MO[2], AND YONG DENG[1]
[1]Institute of Fundamental and Frontier Science, University of Electronic Science and Technology of China, Chengdu 610054, China
[2]Library, Sichuan Minzu College, Sichuan 626001, China

Corresponding author: Yong Deng (dengentropy@uestc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61973332, and in part by the General Natural Research Program of Sichuan Minzu College under Grant XYZB18013ZB.

ABSTRACT Recently time series prediction based on network analysis has become a hot research topic. However, how to more accurately forecast time series with good efficiency is still an open question. To address this issue, we propose an efficient time series forecasting method based on the DC algorithm and visibility relations on the vertexes set. Firstly, the time series is mapped into the network by the DC algorithm, which is a more efficient approach to generate the visibility graph. Then, we use the variation trends (slope) of those nodes that have visibility relation with the last node to get the preliminary predictive values. Afterward, the value of the last node is adopted to obtain the revised predictive values, which are assigned different weights according to node degree and time distance to get the final weighted result. To better demonstrate the prediction performance and applicability of the proposed method, the proposed method is applied to different time series data sets. The empirical results show that the proposed method could provide a higher level of forecasting accuracy than many methods with relatively lower time complexity.

INDEX TERMS Complex network, time series forecasting, node degree, visibility graph, variation trend.

I. INTRODUCTION

Time series refers to the series of values of the same statistical index in the order of their occurrence. In our daily life, there are many data sets in the form of time series, like temperature [1] and precipitation [2], and most of the economic data are given as time series, such as stock prices [3] and construction costs [4]. One of the primary purposes of time series analysis is to predict the future data based on the historical data, like forecasting financial [5]–[7] and functional time series [8], [9]. Therefore, how to improve the accuracy of time series prediction has attracted many researchers' attention.

To improve the accuracy of time series prediction, researchers have proposed various time series forecasting methods, such as simple moving average (SMA) [10] and Holt-exponential smoothing (Holt ES) [11]. Moreover, combining with autoregressive (AR) [12] and moving average (MA), an autoregressive integrated moving average (ARIMA) [13] forecasting model is built with good prediction accuracy. To handle the seasonality of time series, researchers have built the seasonal ARIMA [14] based on the ARIMA model. Besides, since the uncertainty and volatility of time series bring several errors for time series prediction [15], many time series prediction models [16]–[18] based on the fuzzy time series [19] have been proposed due to the effectiveness of dealing with uncertainty [20], [21] by fuzzification. In recent years, due to the ability of machine learning methods to use information efficiently, some of them, such as neural networks [22]–[24] and random forest [25], have been widely used in time series prediction models.

The experimental results of these methods have shown that they have good prediction accuracy. They also promote the development of time series researches. However, these methods have certain limitations. For example, although the SMA model is easy to implement, its accuracy is not high compared with other methods. ARIMA and ARIMA based models could indeed achieve a high level of forecasting accuracy, but the development of an ARIMA or ARIMA based model is cumbersome [26]. Besides, it takes a great deal of time to train these models' parameters, which also happens

The associate editor coordinating the review of this manuscript and approving it for publication was Donghyun Kim.

for these models applying machine learning methods [27]. And every time if new data are available, parameters of these models have to be trained again to ensure the high accuracy level. As for some fuzzy time series based forecasting methods, the establishment of fuzzy sets is a little complicated and subjective [28]. Therefore, it is challenging to build a forecasting model that has a good forecasting accuracy with relatively low time complexity.

Since there are some limitations in those methods, the proposed method develops the prediction model from a different perspective. Lately, complex networks has become a hot topic as a result of its wide application in different fields, such as synchronization [29], [30], fractal analysis [21], [31], identifying influential nodes [32]–[36] and forecasting [37], [38]. Some researchers have also put forward some network-based time series forecasting methods [39]–[41], and the experimental results of them have shown that it is an effective method to transform time series into the network and make time series predictions based on network analysis. Recently, Lacasa et al proposed the visibility algorithm to map time series into the network [42], [43]. Compared with other transformation methods [44], [45], it is simpler and faster, while some properties of time series could be inherited. Some time series prediction models based on the visibility graph have proved their validity. For example, many researchers combine with visibility graph and link prediction [46] to predict time series [47], [48]. Although these methods are more accurate than many existing methods, they take a large amount of time to iterate the local random walk process [49] and merely use a few historical data for prediction. Therefore, this paper aims to build a more efficient network-based time series forecasting model which can improve the prediction accuracy with relatively lower time complexity, at the same time, considering more historical information when forecasting.

In this paper, we propose an efficient network method based on the DC algorithm and visibility relation for time series prediction. First of all, a time series is converted into a network via the DC algorithm, which is a simpler and faster method to generate a visibility graph. Secondly, we use the visibility relations between the last node and its preceding nodes to make the preliminary predictions. In the third step, the average of the preliminary prediction value and the value of the last node is regarded as the revised predictive value. Next, considering the node degree and time distance, the weights of different revised prediction values are assigned to obtain the final weighted result. To examine the prediction performance of this method, we conduct different experiments based on three time series, including the Construction cost index (CCI) [26], the student enrollment of the University of Alabama [50], [51], and the market price of State Bank of India Share [28]. The prediction results under different conditions show that this method is more accurate in terms of some error measurements and less time-consuming than many compared methods.

The structure of this paper is as follows. The second section briefly introduces some basic theories contained in the proposed method, including graph theory and visibility graph. Section 3 describes the process of implementing this method. In Section 4, we show the empirical results and make a discussion about them. Section 5 concludes this paper.

II. PRELIMINARIES

In this section, some basic theories covered in this method will be introduced, including graph theory and visibility graph.

A. GRAPH THEORY

Graph theory deals with the topological structure, involving set, mapping, operation, and relation. It has a wide application in various fields, such as linguistics and computer science. Especially, many pieces of researches about complex networks apply some advanced knowledge of graph theory [52], [53].

A graph mainly includes two parts: the set of vertexes or nodes (V) and the set of edges (E). Normally, a graph is denoted as $G = (V, E)$. For example, there are 5 vertexes and 5 edges in Fig. 1, where solid points represent vertexes, and edges are the lines linking these points. The degree of a vertex $d(v)$ is the number of edges this vertex has. For instance, the degree of vertex v_2 in Fig. 1 is 2, which is denoted as $d(v_2) = 2$, since there are two edges (e_1 and e_2) that link v_2 with other vertexes. Note that $d(v_5)$ is 3 (a loop counts two). Generally, a vertex that has a higher degree is more important in the network [54].

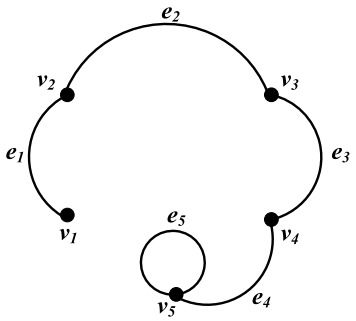


FIGURE 1. A graph consists of 5 vertexes and 5 edges. Solid points, denoted as v_1, v_2, v_3, v_4, and v_5, represent the vertexes in this graph. Edges (e_1, e_2, e_3, e_4, and e_5) are the lines connecting vertexes.

B. VISIBILITY GRAPH

Lacasa *et al.* first proposed the visibility algorithm (Eq. (1)) to map time series into the network [42], at the same time, some geometric properties of the time series could also be inherited in the generated visibility graph. Fig. 2 illustrates the schema of this algorithm. In the upper zone of Fig. 2 we plot a time series that includes 6 values, with every vertical bar representing one value, and two arbitrary bars (two points in this time series) are linked if they could ''see'' each other without any ''obstacles'' between them. One trick to judge whether one point could ''see'' another is that regarding every single bar as a landscape, so other bars are linked (blue

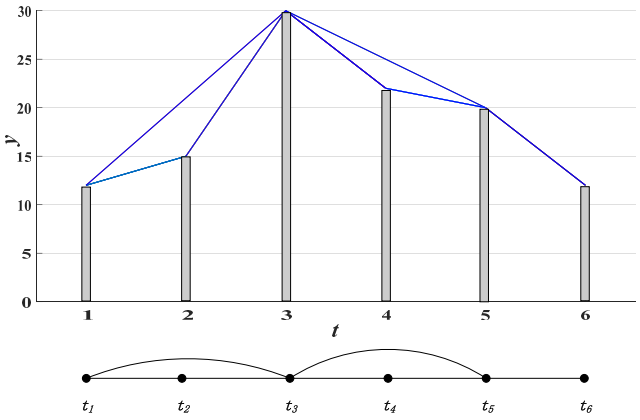


FIGURE 2. An example of a time series including six values that are represented by vertical bars. Every two bars will be linked and become connected nodes in the generated visibility graph if their data values satisfy the visibility algorithm.

lines) with it if these bars could be seen from the top of the considered one. As a result, the associated graph is obtained (shown in the lower part of Fig. 2). In the associated graph, every node corresponds to the series data in the same order. Two nodes are connected if they are visible to each other, in other words, if there is a line that links the series data, provided that this "visibility line" does not intersect any intermediate data height [55], [56].

To be more formally, the visibility algorithm can be described as follows: two arbitrary data values (t_a, y_a) and (t_b, y_b), as two nodes in the generated visibility graph, will be visible and connected with each other if any other data (t_c, y_c) between them $(t_a < t_c < t_b)$ fulfills Eq. (1) [42]:

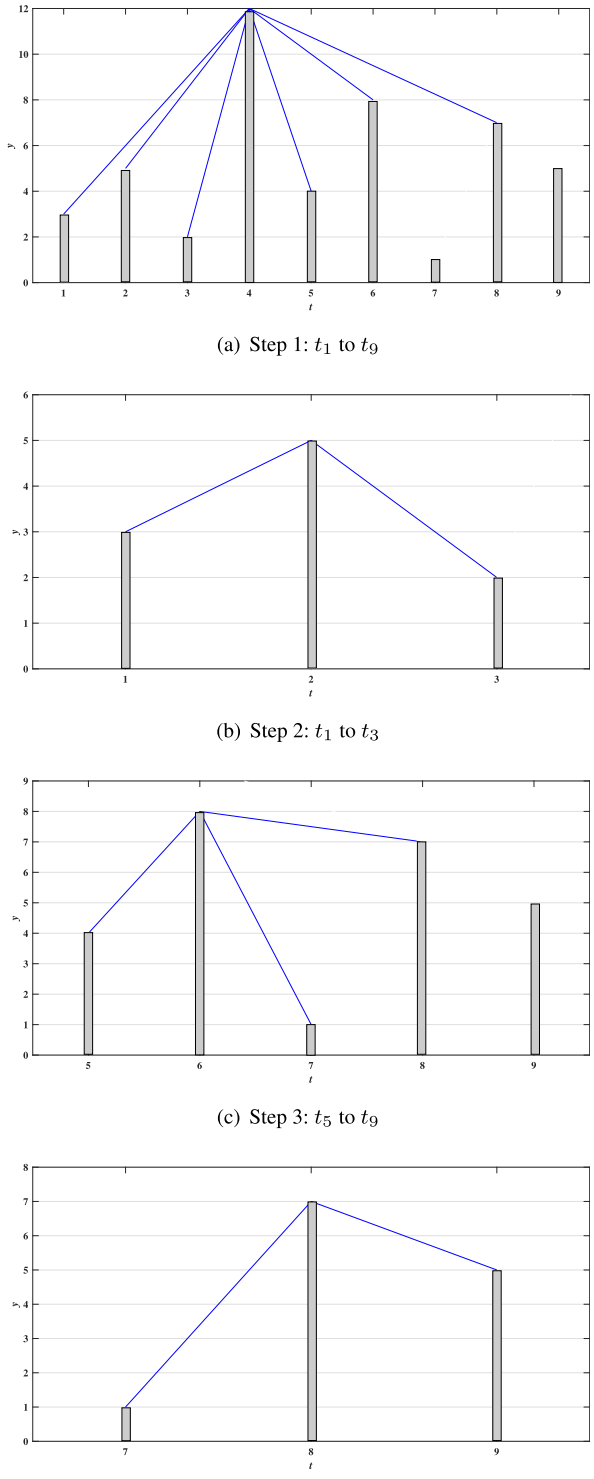
$$\frac{y_c - y_a}{t_c - t_a} < \frac{y_b - y_a}{t_b - t_a}. \tag{1}$$

III. THE PROPOSED METHOD

In this section, the proposed method will be introduced by 2 parts. Part 1 shows how to use DC algorithm to transform a time series *TS* with observed data $\{(t_1, y_1), (t_2, y_2), ..., (t_N, y_N)\}$ into network more quickly than by original visibility algorithm Eq. (1). Part 2 illustrates the steps of forecasting time series based on Part 1's result.

A. PART 1: MAP TIME SERIES INTO THE NETWORK

The visibility algorithm Eq. (1) is a good tool to map time series into the network. However, since we could regard series data as landscape and they could "see" other data if there is no "obstacle" between them. It is unnecessary to judge every two pairs of data whether they are visible to each other with time complexity of $O(n^2)$. It is more efficient to examine visibility based on the maximum local value since nodes on one side of the node that has the maximum local value could not see those on the other side. Therefore, derived from "*divide & conquer*" strategy [57], [58] and combining with the essence of visibility algorithm, Lan *et al.* have proposed the DC algorithm [59] to map time series into the network,



(a) Step 1: t_1 to t_9

(b) Step 2: t_1 to t_3

(c) Step 3: t_5 to t_9

(d) Step 4: t_7 to t_9

FIGURE 3. A schematic illustration of using the DC algorithm to generate a visibility graph from a time series that includes 9 data.

which saves much time when generating visibility graph especially for large-scale time series, decreasing the time complexity from $O(n^2)$ to $O(n \log n)$.

For illustration purpose, Fig. 3 shows that how to adopt DC algorithm to generate a visibility graph from a time series

with 9 data. Firstly, finding the node that has the maximum value in this time series, which is t_4. It is clear that t_4 could "see" and connect with nodes t_1, t_2, t_3, t_5, t_6, and t_8 (shown in Fig. 3(a)). But nodes on the left side of t_4 could not "see" and connect with those nodes on the right side of t_4, because t_4 blocks their "view". Next, based on the strategy mentioned above, the nodes on both sides of t_4 are divided into two distinct parts that will then be handled separately (Fig. 3(b) and Fig. 3(c)) . In Fig. 3(b), obviously node t_2 has the maximum local value, so it is linked to t_1 and t_3 in the visibility graph whereas t_1 and t_3 have no connection. Afterward, since there are only one node left on both sides of t_2, there is no need to get further division for them. These division and connection processes of the right part of (shown in Fig. 3(c)) node t_4 are similar with those of the left part, but one difference is that after linking node t_6 with those nodes that can be seen from it, the right side of t_6 has more than one node. Consequently, we repeat the division process for the right side of t_6. Finally, the whole implementing process ends with connecting t_8 with t_7 and t_9 (shown in Fig. 3(d)).

In summary, in each step of the DC algorithm, it is only necessary to connect the local maximum node with those nodes that could be seen from it. Next, the right side and left side of the maximum node are regarded as two parts and handled separately. Then, the previous steps are repeated in both divided parts until there is only one node on both sides of the last found local maximum node in these parts. Consequently, the DC algorithm avoids judging every pair of nodes to generate the visibility graph and saves much time. Algorithm 1 summarizes the DC algorithm.

Algorithm 1 DC Algorithm

```
function FASTERGENERATEVG(TS, left, right)
    // TS is the time series to be transformed and the time index of it could be 1 to n. left and right are two indexes denoting the occurrence time of nodes.
    // VG is the generated visibility graph
    if (left < right) then
        //there must be at least two points in the input time series
        k = the index of the maximum value of TS[left...right]
        for i = left → right do
            // check whether node k could "see" node i
            if node k could "see" node i then
                add edge={k, i} to VG
            end if
        end for
    end if
    FASTERGENERATEVG(TS, left, k − 1);// the left side of node k
    FASTERGENERATEVG(TS, k+1, right);// the right side of node k
end function
```

B. PART 2: MAKE PREDICTIONS

Step 1 (Identifying the Nodes That Are Visible to the Last Node): Previous data play a significant role in time series prediction. Inspired by the exponential smoothing (ES) model, the last observed data should get more attention under the assumption that the current time is the most important one. Moreover, to make full use of the historical data when making predictions, the relation between the last node t_N and previous $N-1$ nodes are considered. The generated visibility graph in Part 1 indicates that those nodes that are connected with t_N have a visibility relation with t_N. We denote those nodes as t_i^* with superscript $*$ denoting that this node has a visibility relation with t_N. Subscript i represents the time index of the node that is connected with t_N, and the value of i could take from 1 to $N-1$ (no loops in the visibility graph). In the following steps, this model will forecast future data based on the visibility relations between t_i^* and t_N.

Step 2 (Making the Preliminary Prediction Based on Different Slopes): From a geometric perspective, actually the process of forecasting the value of time point t_{N+1} is looking for a variation trend of t_N, namely the slope between t_N and t_{N+1}. Based on the visibility relations mentioned in Step 1 of Part 2, the variation trend of each t_i^* is seen as the variation trend for t_N. That is to say, the predictive variation trend of t_N is considered to be the slope between (t_i^*, y_i) and (t_{i+1}, y_{i+1}). Therefore, Eq. (2) is used to get the preliminary predictive value of t_{N+1}, and this value is denoted as y_i^*. Note that i denotes the time index of the previous nodes which have visibility relations with the last observed node t_N. If there is more than one node connecting with t_N in the visibility graph, then various y_i^* will be obtained based on (t_i^*, y_i) and (t_{i+1}, y_{i+1}).

$$y_i^* = \frac{y_{i+1} - y_i}{t_{i+1} - t_i^*}(t_{N+1} - t_N) + y_N \tag{2}$$

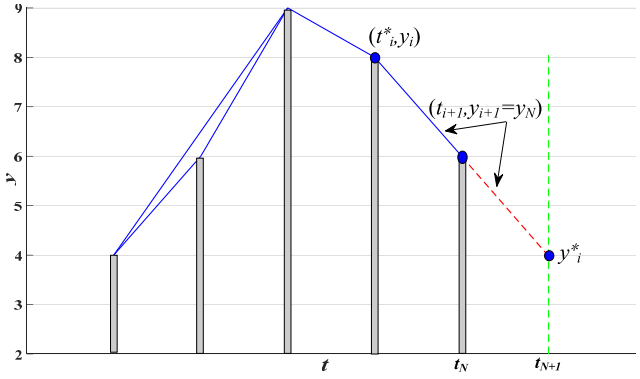


FIGURE 4. The illustration of Step 2 in Part 2: The green dotted line represents the future time t_{N+1}. The two lines pointed by arrows have the same slope. The slope between (t_i^*, y_i) and (t_{i+1}, y_{i+1}) is applied to link (t_N, y_N) and the green dotted line to obtain the preliminary predictive value y_i^*.

To efficiently illustrate this step, Fig. 4 shows how to obtain y_i^* when there is only one node having visibility relation

with t_N. The future time t_{N+1} is represented by the green dotted line, and the slope between (t_i^*, y_i) and (t_{i+1}, y_{i+1}) is adopted to connect t_N with t_{N+1} (red dotted line). As a result, the y coordinate of the junction point of this red line with the green dotted line is taken as the preliminary predictive value y_i^*.

Step 3 (Obtaining the Revised Predictive Value Based on the Last Observed Value): In Step 2 of Part 2, the variation trend of past node t_i^* is adopted to forecast the value of future time t_{N+1}. However, the variation trend is merely one factor that has an impact on predictive value. In this step, the current data (y_N) is considered as a basic value to revise the preliminary predictive value and Eq. (3) is adopted to get the revised predictive value $\bar{y}_i$ which is calculated as follows:

$$\bar{y}_i = \frac{1}{2}y_i^* + \frac{1}{2}y_N, \tag{3}$$

where y_i^* is the preliminary predictive value obtained in Step 1 and y_N is the value of t_N.

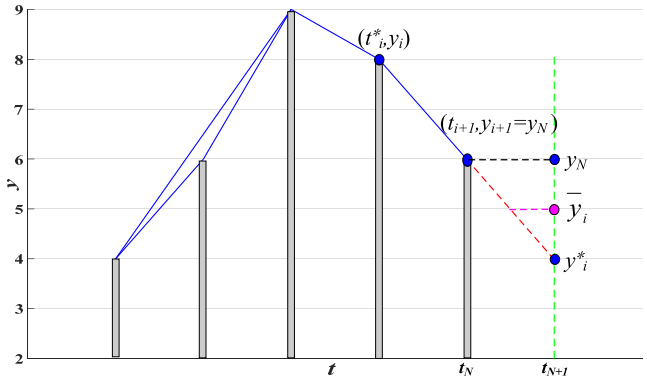


FIGURE 5. The illustration of Step 3 in Part 2: y_i^* denotes the preliminary predictive value of future time t_{N+1} and y_N is the last observed data. The midpoint (pink point) of segment $y_N y_i^*$ denotes the revised predictive value $\bar{y}_i$, which is the average of y_N and y_i^*.

Fig. 5 demonstrates this step more vividly. Similarly, there is only one node that is visible to t_N on the purpose of more efficiently illustrating this process. So the only one value $\bar{y}_i$ obtained is our revised predictive value for future time t_{N+1}.

Step 4 (Considering the Degree and Time Distance to Get the Final Result): For better illustrating the proposed method, the previous example only discusses the situation when there is only one node connecting with t_N. However, time series, in reality, is much more complex, and t_N may have visibility relations with many preceding nodes (denoted as t_i^*). In this case, various revised predictive values $(\bar{y}_i)$ are obtained. But obviously, the influence of t_i^* on future data are different. Consequently, the proposed method measures the effects of t_i^* on future data according to the degree of t_i^* and the time index of t_i^*.

On the one hand, as mentioned in the preliminaries, generally the bigger the degree a node has, the more important this node is in the network. On the other hand, if a node is far away from the last node, it means that the data of this node was recorded long time ago, so if there is no evident periodicity shown in the considered time series, the impact of this data on the future data is supposed to be small. Therefore, combined with these two factors, Eq. (4) is adopted to assign weights for different revised predictive values $\bar{y}_i$.

$$w_i = \frac{\frac{d(t_i^*)}{d_{iN}}}{\sum_{i=t_f}^{t_l} \frac{d(t_i^*)}{d_{iN}}}, \tag{4}$$

where $d(t_i^*)$ is the degree of node t_i^* and d_{iN} is the distance between t_i and t_N, namely $N - i$. t_f represents the time index of the first node that is visible to t_N while t_l represents the time index of the last node having visibility relation with t_N.

Note that since there is no loop in the visibility graph, if there is only one node in the visibility graph, namely $t_1 = t_N, d(t_i^*)$ is zero. In this case, the value of the single node is considered as the final predictive value. After the weights of revised predictive values $\bar{y}_i$ are determined, we use Eq. (5) to get the final predictive value $\hat{y}_{N+1}$.

$$\hat{y}_{N+1} = \sum_{i=t_f}^{t_l} w_i \bar{y}_i \tag{5}$$

The proposed method is summarized in Fig. 6.

IV. PREDICTION RESULTS AND DISCUSSION

To test the predictive performance of the proposed method, we apply the proposed method to predict the values of three different time series, including the Construction Cost Index (CCI), the student enrollment of the University of Alabama, and the price of SBI's share at BSE, India. At the same time, to further evaluate the forecasting performance of the proposed method, the predicted results of this method are compared with those of some forecasting models.

We use some common error indicators to measure the forecasting accuracy of different models, i.e. mean absolute difference (MAD), mean absolute percent error (MAPE), root mean square error (RMSE), and normalized root mean squared error (NRMSE). Except for error measurements, statistical parameters like correlation coefficient (R) and coefficient of determination (R^2) are also used to verify the predictive performance of these models. Eqs. (6-10) show the definitions of these error indicators and statistical parameters. Besides, the time complexity of some models are also calculated to evaluate their efficiency.

$$MAD = \frac{1}{N}\sum_{t=1}^{N}\left|\hat{y}(t) - y(t)\right| \tag{6}$$

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\frac{\left|\hat{y}(t) - y(t)\right|}{y(t)} \tag{7}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left[\hat{y}(t) - y(t)\right]^2} \tag{8}$$

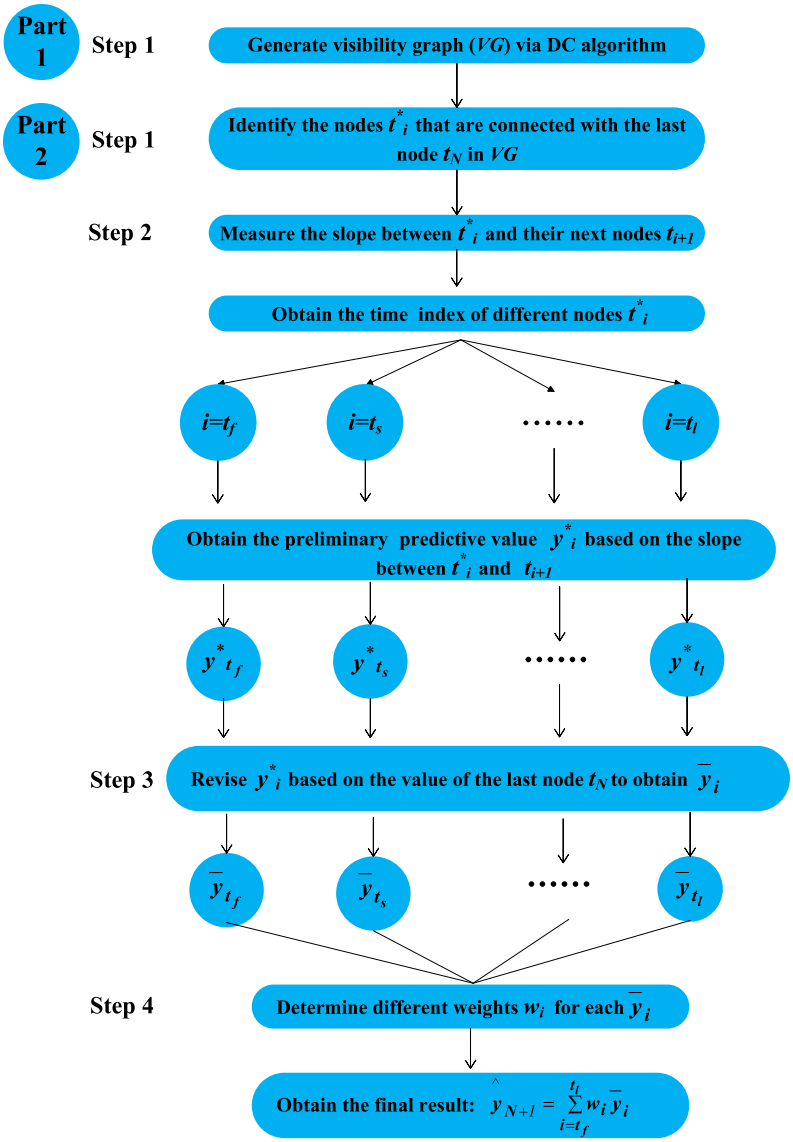


FIGURE 6. A brief summarization of the proposed method.

$$NRMSE = \frac{\sqrt{\frac{1}{N}\sum_{t=1}^{N}\left[\hat{y}(t) - y(t)\right]^2}}{y_{max} - y_{min}} \tag{9}$$

$$R = \frac{\sum_{t=1}^{N}(\hat{y}(t) - \hat{y}_o)(y(t) - y_o)}{\sqrt{\sum_{t=1}^{N}(\hat{y}(t) - \hat{y}_o)\sum_{t=1}^{N}(y(t) - y_o)}} \tag{10}$$

Note that N is the number of predictive values. $\hat{y}(t)$ denotes the predictive values and $y(t)$ denotes the actual values. $\hat{y}_o(t)$ and $y_o(t)$ are the mean values of the predictive and actual values respectively.

A. FORECASTING CCI BY THE PROPOSED METHOD

The cost construction index (CCI) is a weighted index reflecting price information of various materials in construction projects. It is published monthly on *Engineering News-Record* (ENR). Many violent variations shown in the observed CCI data set make it difficult to predict CCI accurately. Therefore, improving the prediction accuracy of CCI has become the focus of many researchers. [26], [60].

In this section, CCI from January 1990 to July 2014 are adopted as the universal set, and CCI from January 1990 to June 2014 are used to predict CCI between March 1990 and July 2014. The universal set contains 295 data, which are denoted as $CCI(i)$, i representing the time index. 293 predictive values are saved in set $\hat{C}CI(i)$. Note that at least two CCI values are used for forecasting the future CCI. For instance, when making the first prediction, there are only two CCI values (January 1990 and February 1990) in the training set, and these two values are used to forecast the CCI in March 1990. Then the actual CCI in March 1990 is added to the training set to forecast the CCI in April 1990. This process

TABLE 1. Error measurements and time complexity of different CCI forecasting models.

Models	MAD	MAPE(%)	RMSE	NRMSE(%)	T(n)
SMA (k=1) [26]	21.6450	0.3117	32.7870	0.6417	**O(1)**
Zhang *et al.* [48]	20.7571	0.3013	29.7315	0.5800	O(t*n^3)
Mao *et al.* [49]	20.6686	0.3006	29.4463	0.5700	O(t*n^3)
Proposed method	**19.9250**	**0.2882**	**29.0590**	**0.5600**	O(nlogn)

is repeated until CCI in July 2014 is obtained. Algorithm 2 further explains the steps to implement this experiment.

Algorithm 2 The CCI Forecasting Process

Input: The universal set $CCI(i)$: i from 1 to 295
Output: Predictive values and errors
1: **for** j=2 to 294 **do**
2: Training set = $\{x \mid x \in CCI(i), i = 1, 2, ..., j\}$
3: Obtain the predictive value $\hat{C}CI(j+1)$ with the proposed method
4: Calculate the error between actual value $CCI(j+1)$ and predictive value $\hat{C}CI(j+1)$
5: Add $CCI(j+1)$ to the training set
6: **end for**

In Fig. 7, the predictive and actual values are represented by the red and black curves, respectively. Generally, the red and black curves fit well, indicating that the predictive values are close to the actual values. Since the observed CCI data set shows that it was greatly affected by dramatic short-term variations, to better verify the predictive performance of the proposed method, we divide the forecasting results into three groups, with 80 data in each group. The prediction results of each group are depicted in Fig. 8. Likewise, the predictive and actual values of each group are close.

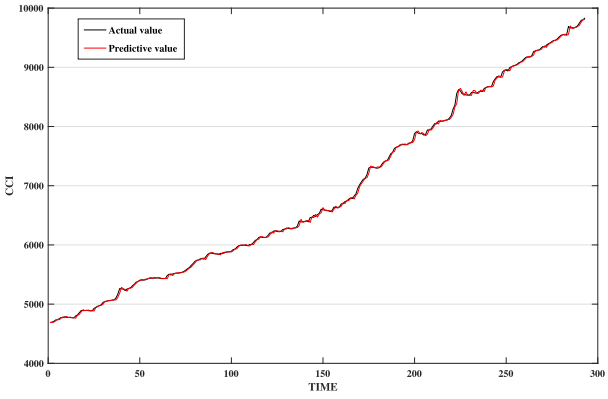


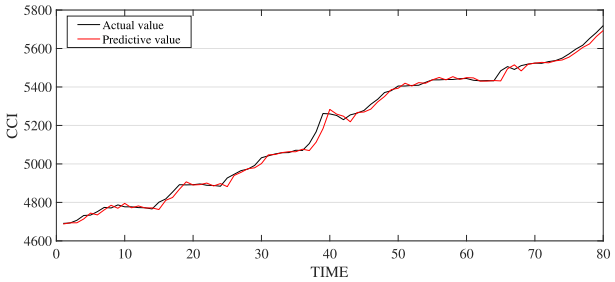
FIGURE 7. The result of the CCI forecasting experiment: On the whole, the red curve (predictve values) fits the black curve (actual values) well.

Zhang *et al.* have proposed a network-based CCI forecasting method [47]. According to their research, their CCI forecasting model has better prediction performance than many traditional and statistical time series forecasting models, such as SMA [10] and Seasonal ARIMA [14]. In this section, predictive results of the proposed method are compared with a traditional time series forecasting model (SMA) and two network-based forecasting models in 3 error indicators and time complexity T(n). Tables 1-2 show more details of all the comparison results. Note that in Table 1, n and t denote the number of nodes in the network and the number of iterations, respectively. Obviously, in general, the proposed method has a better prediction performance than compared methods when forecasting large and small scale CCI in terms of shown error measurements. Although the time complexity of our model is higher than that of the SMA model, compared with other network-based prediction models, the proposed method could save a great deal of time.

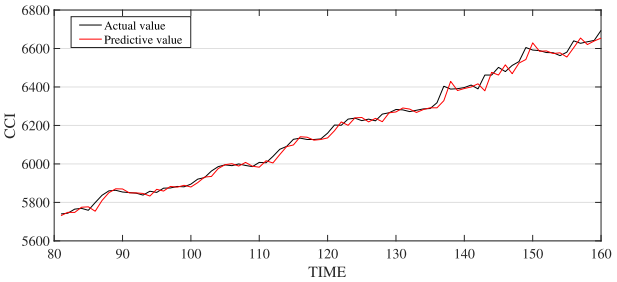
B. FORECASTING STUDENT ENROLLMENT BY THE PROPOSED METHOD

In this part, the proposed method is used to predict student enrollment at the University of Alabama. Specifically, student enrollments from 1971 to 1992 are adopted as the universal set, and the number of students enrolled in 1971-1991 are used to predict the number of students enrolled in 1972-1992. Since there is only one node in the network in the first forecasting experiment, the actual student enrollment in 1971 is used as the predicted student enrollment in 1972. Afterward, the student enrollments in 1971 and 1972 are used to predict student enrollment in 1973. Similarly, after every prediction process, the actual enrollment is added to the training set for the next prediction. This process is repeated until obtaining the predicted enrollment in 1992.

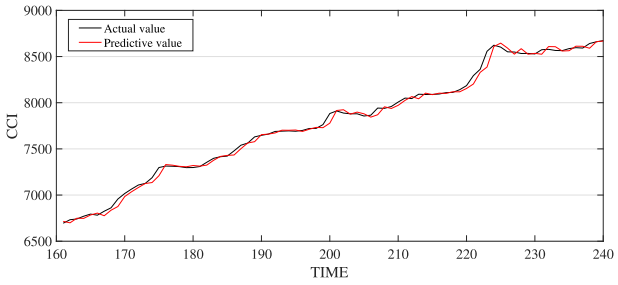
Actual student enrollment (blue curve) and predictive student enrollment (red curve) of the University of Alabama are depicted in Fig. 9. The predictive value fluctuates near the actual value and could conform to the variation trend of the actual value, which indicates that the proposed method could provide excellent performance for the present experiment. Also, to better evaluate the prediction performance of the



(a) The predictive results of group 1



(b) The predictive results of group 2



(c) The predictive results of group 3

FIGURE 8. **The CCI prediction results of three groups: The predictive values (red curve) and the actual values (black curve) are close in each group.**

TABLE 2. **Error measurements of CCI predicting results in three groups.**

Groups	Models	MAD	MAPE(%)	RMSE	NRMSE(%)
Group 1	SMA (k=1) [26]	28.0625	0.5402	39.0210	0.0406
	Zhang *et al.* [48]	14.2638	0.2760	19.2442	0.0187
	Mao *et al.* [49]	14.3694	0.2779	19.2540	0.0187
	Proposed method	**13.9483**	**0.2696**	19.6783	0.0191
Group 2	SMA (k=1) [26]	17.5875	0.2823	24.8850	0.0276
	Zhang *et al.* [48]	19.9201	0.3184	26.7975	0.0281
	Mao *et al.* [49]	20.1613	0.3221	27.2237	0.0285
	Proposed method	18.1254	0.2903	**24.3018**	**0.0255**
Group 3	SMA (k=1) [26]	30.0500	0.3877	44.7035	0.0227
	Zhang *et al.* [48]	28.5976	0.3663	39.0111	0.0197
	Mao *et al.* [49]	27.8579	0.3572	38.3157	0.0194
	Proposed method	**27.0050**	**0.3460**	**38.2594**	**0.0194**

proposed method in this experiment, some forecasting methods based on the fuzzy time series model [61]–[64] are taking into comparison. In this case, we do not compare the time complexity of these listed methods, since it is unreasonable to compare the time complexity of the proposed method with these fuzzy time series forecasting models. The reason is

TABLE 3. The precition errors of different models in student enrollment prediction.

Models	RMSE	MAPE(%)	R	R^2
Chou [62]	781.47	3.6075	0.8866	0.78
Yolcu *et al.* [63]	805.17	4.2886	0.9121	0.83
Pathak and Singh [64]	646.67	2.9883	0.9373	0.88
Gangwar and Kumar [65]	642.58	2.9672	0.9243	0.85
Proposed method	**562.89**	**2.9532**	**0.9569**	**0.91**

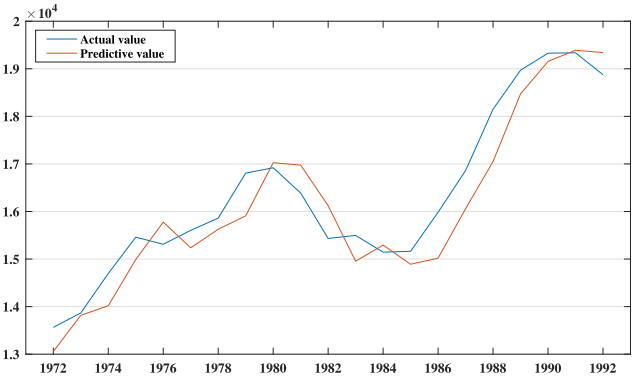


FIGURE 9. The result of the student enrollment forecasting experiment: actual enrollments are represented by the blue curve whereas the red curve represents predictive enrollment.

that one step contained in these fuzzy time series forecasting methods is to construct different fuzzy sets based on the division of the universe of discourse [19]. Then the fuzzy logic relationships between these fuzzy sets are established, and the time complexity of this establishing process is $O(n * n_s)$, where n and n_s are the number of observed data and fuzzy sets, respectively. But how to divide the universe of discourse to get different intervals (the number of fuzzy sets) is depended on the data sets and implementer's consideration in different situations, so n_s is not fixed in different experiments whereas the proposed method has a fixed time complexity of O(nlogn). Therefore, Table 3 shows the comparison results of different models in terms of error measurements and statistical parameters. The decrease of RMSE and MAPE shows that the proposed method could improve the accuracy of enrollment prediction in this experiment compared with these listed fuzzy time series based methods. Besides, the higher values of statistical parameters R and R^2 also confirms a better correlation between recorded and predicted enrollment.

C. FORECASTING MARKET PRICE OF STATE BANK OF INDIA SHARE BY THE PROPOSED METHOD

In this part, we apply the proposed method to predict the market prices of State Bank of India (SBI) share at BSE India. To be specific, the actual market prices of SBI share for each month from April 2008 to February 2010 are used to predict the market prices from May 2008 to March 2010. In this experiment, since at first there is only one node in the network (April 2008), the prediction process begins with regarding the market price in April 2008 as the predictive value in May 2008. The actual price in May 2008 is then added to the training set to predict the price in June 2008. Likewise, after each prediction, the actual price is added to the training set for the next prediction. These steps are iterated until the price in March 2010 is forecasted.

TABLE 4. Error measurement of different methods in the SBI forecasting experiment.

Models	MAPE(%)	MAD	NRMSE(%)	T(n)
Mao *et al.* [49]	10.1896	176.5820	0.1673	$O(t*n^3)$
Zhang *et al.* [48]	9.5127	166.0369	0.1621	$O(t*n^3)$
Proposed method	**8.7325**	**152.6564**	**0.1494**	**O(nlogn)**

Three error measurements are employed to assess the prediction accuracy of the proposed method in this experiment, and the experimental results and time complexity of two recently proposed network-based forecasting models are compared with those of the proposed method. The details are shown in Table 4. Note that n denotes the number of nodes (observed data) in the network, and t denotes the iteration number, which is very large [47]. As can be seen, compared with the other two methods, the method proposed in this paper is more accurate for SBI prediction in terms of these error measurements. Also, the time complexity of the proposed method is much lower than those of the compared methods.

D. DISCUSSIONS

According to the experiment results and some error analysis of these conducted experiments, we analyze the proposed method as follows.

1) Recently, many researchers have paid much attention to time series analysis, but how to build a network

model for time series forecasting with good accuracy and relatively low time complexity is still an unsolved problem. In this paper, an efficient time series forecasting method is proposed with good forecasting accuracy. The main steps of implementing this method are using the DC algorithm to map time series into the network and making predictions based on the visibility relation between historical data. What's more, the proposed method is explained and implemented in a geometric way, which is easy to understand and carry out.

2) When talking about the prediction performance of the proposed method, we apply the proposed method to predict three different time series, and the experimental results are compared with other methods. The comparison results show that although the SMA model has the lowest time complexity (O(1)), the prediction accuracy of the SMA model is not that high. Zhang *et al.* have proposed a network-based method [47], which is more accurate than many methods in CCI prediction, such as SMA and ARIMA. Based on Zhang *et al.*'s method, Mao *et al.* [48] have proposed an improved forecasting method that achieves a higher forecasting accuracy when forecasting CCI than Zhang *et al.*'s method. But these two methods need much time to iteration and only use a few historical data to forecast time series. As shown in Tables 1 and 4, the proposed method could further improve the forecasting accuracy than these methods with much lower time complexity, at the same time, more relations between the observed data are used to make predictions. Besides, the prediction performance of the proposed method is better than many fuzzy time series based methods (shown in Table 3) and avoid subjectiveness brought by constructing fuzzy sets.

V. CONCLUSION

In this paper, an efficient time series forecasting method is proposed based on network analysis. At first, a time series is transformed into the network by the DC algorithm, which provides a more efficient way to generate a visibility graph. Afterward, the previous nodes that are visible to the last node are identified, and then the variation trends (slope) of them are used to make preliminary predictions. Next, considering the effect of the last node, the revised predictive values are obtained. After the weights of these revised values are allocated according to node degree and time distance, we get the final weighted result.

To verify the prediction performance of this method, we adopt three different time series for prediction experiments. The good experimental results indicate that this method can improve prediction accuracy compared with many methods and has good applicability. At the same time, the time complexity of the proposed method is much lower, and more relations between historical data are used to make predictions.

In our future work, to further improve the forecasting accuracy, we will focus on taking more properties of time series, like seasonality and periodicity, into consideration when making predictions. For example, based on these properties, adding directions for the edges in the visibility graph.

REFERENCES

[1] J. Song, J. Lu, E. Wang, M. Lu, Z. An, Y. Liu, X. Zeng, W. Li, H. Li, D. Xu, S. Yao, and W. Wu, "Short-term effects of ambient temperature on the risk of premature rupture of membranes in Xinxiang, China: A time-series analysis," *Sci. Total Environ.*, vol. 689, pp. 1329–1335, Nov. 2019.

[2] A. Troncoso, P. Ribera, G. Asencio-Cortés, I. Vega, and D. Gallego, "Imbalanced classification techniques for monsoon forecasting based on a new climatic time series," *Environ. Model. Softw.*, vol. 106, pp. 48–56, Aug. 2018.

[3] S. M. Idrees, M. A. Alam, and P. Agarwal, "A prediction approach for stock market volatility based on time series data," *IEEE Access*, vol. 7, pp. 17287–17298, 2019.

[4] T. Moon and D. H. Shin, "Forecasting construction cost index using interrupted time-series," *KSCE J. Civil Eng.*, vol. 22, no. 5, pp. 1626–1633, May 2018.

[5] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on CEEMDAN and LSTM," *Phys. A, Stat. Mech. Appl.*, vol. 519, pp. 127–139, Apr. 2019.

[6] J. Li, P. Shang, and X. Zhang, "Financial time series analysis based on fractional and multiscale permutation entropy," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 78, Nov. 2019, Art. no. 104880.

[7] Y. Zhang, P. Shang, and H. Xiong, "Multivariate generalized information entropy of financial time series," *Phys. A, Stat. Mech. Appl.*, vol. 525, pp. 1212–1223, Jul. 2019.

[8] H. L. Shang and R. J. Hyndman, "Grouped functional time series forecasting: An application to age-specific mortality rates," *J. Comput. Graph. Statist.*, vol. 26, no. 2, pp. 330–343, Apr. 2017.

[9] Y. Gao, H. L. Shang, and Y. Yang, "High-dimensional functional time series forecasting: An application to age-specific mortality rates," *J. Multivariate Anal.*, vol. 170, pp. 232–243, Mar. 2019.

[10] F. R. Johnston, J. E. Boyland, M. Meadows, and E. Shale, "Some properties of a simple moving average when applied to forecasting a time series," *J. Oper. Res. Soc.*, vol. 50, no. 12, p. 1267, Dec. 1999.

[11] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *Int. J. Forecasting*, vol. 20, no. 1, pp. 5–10, Jan. 2004.

[12] W. Xu, H. Peng, X. Zeng, F. Zhou, X. Tian, and X. Peng, "Deep belief network-based AR model for nonlinear time series forecasting," *Appl. Soft Comput.*, vol. 77, pp. 605–621, Apr. 2019.

[13] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis, Forecasting and Control*. Hoboken, NJ, USA: Wiley, 2015.

[14] F.-M. Tseng, H.-C. Yu, and G.-H. Tzeng, "Combining neural network model with seasonal time series ARIMA model," *Technol. Forecasting Social Change*, vol. 69, no. 1, pp. 71–87, Jan. 2002.

[15] Abhishekh, S. S. Gautam, and S. R. Singh, "A new type 2 fuzzy time series forecasting model based on three-factors fuzzy logical relationships," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 27, no. 2, pp. 251–276, Apr. 2019.

[16] C. Cheng, G. Cheng, and J. Wang, "Multi-attribute fuzzy time series method based on fuzzy clustering," *Expert Syst. Appl.*, vol. 34, no. 2, pp. 1235–1242, Feb. 2008.

[17] G. Liu and F. Xiao, "Time series data fusion based on evidence theory and OWA operator," *Sensors*, vol. 19, no. 5, p. 1171, Mar. 2019.

[18] S. Kumar and S. S. Gangwar, "Intuitionistic fuzzy time series: An approach for handling nondeterminism in time series forecasting," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 6, pp. 1270–1281, Dec. 2016.

[19] Q. Song and B. S. Chissom, "Fuzzy time series and its models," *Fuzzy Sets Syst.*, vol. 54, no. 3, pp. 269–277, Mar. 1993.

[20] L. Yin and Y. Deng, "Toward uncertainty of weighted networks: An entropy-based model," *Phys. A, Stat. Mech. Appl.*, vol. 508, pp. 176–186, Oct. 2018.

[21] T. Wen, M. Song, and W. Jiang, "Evaluating topological vulnerability based on fuzzy fractal dimension," *Int. J. Fuzzy Syst.*, vol. 20, no. 6, pp. 1956–1967, Aug. 2018.

[22] R. Talavera-Llames, R. Pérez-Chacón, A. Troncoso, and F. Martínez-Álvarez, "Big data time series forecasting based on nearest neighbours distributed computing with spark," *Knowl.-Based Syst.*, vol. 161, pp. 12–25, Dec. 2018.
[23] J. Torres, A. Galicia, A. Troncoso, and F. Martínez-Álvarez, "A scalable approach based on deep learning for big data time series forecasting," *Integr. Comput.-Aided Eng.*, vol. 25, no. 4, pp. 335–348, Sep. 2018.
[24] P. Amini and M. Khashei, "A soft intelligent allocation-based hybrid model for uncertain complex time series forecasting," *Appl. Soft Comput.*, vol. 84, Nov. 2019, Art. no. 105736.
[25] A. Galicia, R. Talavera-Llames, A. Troncoso, I. Koprinska, and F. Martínez-Álvarez, "Multi-step forecasting for big data time series based on ensemble learning," *Knowl.-Based Syst.*, vol. 163, pp. 830–841, Jan. 2019.
[26] B. Ashuri and J. Lu, "Time series analysis of ENR construction cost index," *J. Construct. Eng. Manage.*, vol. 136, no. 11, pp. 1227–1237, Nov. 2010.
[27] Q. Zhu, H. Ma, and W. Lin, "Detecting unstable periodic orbits based only on time series: When adaptive delayed feedback control meets reservoir computing," *Chaos*, vol. 29, no. 9, Sep. 2019, Art. no. 093125.
[28] S. S. Gautam and Abhishekh, "A novel moving average forecasting approach using fuzzy time series data set," *J. Control, Automat. Elect. Syst.*, vol. 30, no. 4, pp. 532–544, Aug. 2019.
[29] B. Wei, F. Xiao, and Y. Shi, "Synchronization in Kuramoto oscillator networks with sampled-data updating law," *IEEE Trans. Cybern.*, to be published.
[30] B. Wei, F. Xiao, and Y. Shi, "Fully distributed synchronization of dynamic networked systems with adaptive nonlinear couplings," *IEEE Trans. Cybern.*, to be published.
[31] T. Wen and W. Jiang, "An information dimension of weighted complex networks," *Phys. A, Stat. Mech. Appl.*, vol. 501, pp. 388–399, Jul. 2018.
[32] Y. Wang, S. Wang, and Y. Deng, "A modified efficiency centrality to identify influential nodes in weighted networks," *Pramana*, vol. 92, no. 4, p. 68, 2019.
[33] T. Wen and W. Jiang, "Identifying influential nodes based on fuzzy local dimension in complex networks," *Chaos, Solitons Fractals*, vol. 119, pp. 332–342, Feb. 2019.
[34] L. Fei, Q. Zhang, and Y. Deng, "Identifying influential nodes in complex networks based on the inverse-square law," *Phys. A, Stat. Mech. Appl.*, vol. 512, pp. 1044–1059, Dec. 2018.
[35] B. Wei and Y. Deng, "A cluster-growing dimension of complex networks: From the view of node closeness centrality," *Phys. A, Stat. Mech. Appl.*, vol. 522, pp. 80–87, May 2019.
[36] H. Mo and Y. Deng, "Identifying node importance based on evidence theory in complex networks," *Phys. A, Stat. Mech. Appl.*, vol. 529, Sep. 2019, Art. no. 121538.
[37] C. Curme, H. E. Stanley, and I. Vodenska, "Coupled network approach to predictability of financial market returns and news sentiments," *Int. J. Theor. Appl. Finance*, vol. 18, no. 7, Nov. 2015, Art. no. 1550043.
[38] S. Begušić, Z. Kostanjčar, D. Kovač, H. E. Stanley, and B. Podobnik, "Information feedback in temporal networks as a predictor of market crashes," *Complexity*, vol. 2018, pp. 1–13, Sep. 2018.
[39] Y. Teng and P. Shang, "Transfer entropy coefficient: Quantifying level of information flow between financial time series," *Phys. A, Stat. Mech. Appl.*, vol. 469, pp. 60–70, Mar. 2017.
[40] X. Zhao and P. Shang, "Principal component analysis for non-stationary time series based on detrended cross-correlation analysis," *Nonlinear Dyn.*, vol. 84, no. 2, pp. 1033–1044, Apr. 2016.
[41] X. Zhang, M. Chen, M. Wang, Y. Ge, and H. Stanley, "A novel hybrid approach to baltic dry index forecasting based on a combined dynamic fluctuation network and artificial intelligence method," *Appl. Math. Comput.*, vol. 361, pp. 499–516, Nov. 2019.
[42] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nuño, "From time series to complex networks: The visibility graph," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 13, pp. 4972–4975, Apr. 2008.
[43] L. Lacasa, B. Luque, J. Luque, and J. C. Nuño, "The visibility graph: A new method for estimating the Hurst exponent of fractional Brownian motion," *Europhys. Lett.*, vol. 86, no. 3, pp. 30001–30005, May 2009.
[44] T. T. Zhou, N. D. Jin, and Z.-K. Gao, "Limited penetrable visibility graph for establishing complex network from time series," *Acta Phys. Sin.*, vol. 61, Mar. 2012.
[45] Z.-K. Gao, Q. Cai, Y.-X. Yang, W.-D. Dang, and S.-S. Zhang, "Multiscale limited penetrable horizontal visibility graph for analyzing nonlinear time series," *Sci. Rep.*, vol. 6, Oct. 2016, Art. no. 35622.
[46] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Phys. A, Stat. Mech. Appl.*, vol. 390, no. 6, pp. 1150–1170, Mar. 2011.
[47] R. Zhang, B. Ashuri, and Y. Deng, "A novel method for forecasting time series based on fuzzy logic and visibility graph," *Adv. Data Anal. Classification*, vol. 11, no. 4, pp. 759–783, Dec. 2017.
[48] S. Mao and F. Xiao, "Time series forecasting based on complex network analysis," *IEEE Access*, vol. 7, pp. 40220–40229, 2019.
[49] W. Liu and L. Lü, "Link prediction based on local random walk," *Europhys. Lett.*, vol. 89, no. 5, Mar. 2010, Art. no. 58007.
[50] Q. Song and B. S. Chissom, "Forecasting enrollments with fuzzy time series—Part I," *Fuzzy Sets Syst.*, vol. 54, no. 1, pp. 1–9, 1993.
[51] Q. Song and B. S. Chissom, "Forecasting enrollments with fuzzy time series—Part II," *Fuzzy Sets Syst.*, vol. 62, no. 1, pp. 1–8, Feb. 1994.
[52] Z.-K. Gao, Z.-B. Wang, Y.-X. Yang, S. Li, W.-D. Dang, and X.-Q. Mao, "Wavelet multiresolution complex network for decoding brain fatigued behavior from P300 signals," *Phys. A, Stat. Mech. Appl.*, vol. 506, pp. 221–228, Sep. 2018.
[53] R. Zhang, Y. Zou, J. Zhou, Z.-K. Gao, and S. Guan, "Visibility graph analysis for re-sampled time series from auto-regressive stochastic processes," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 42, pp. 396–403, Jan. 2017.
[54] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, p. 378, 2000.
[55] F. Liu and Y. Deng, "A fast algorithm for network forecasting time series," *IEEE Access*, vol. 7, pp. 102554–102560, 2019.
[56] P. Xu, R. Zhang, and Y. Deng, "A novel visibility graph transformation of time series into weighted networks," *Chaos, Solitons Fractals*, vol. 117, pp. 201–208, Dec. 2018.
[57] C. E. Leiserson, R. L. Rivest, T. H. Cormen, and C. Stein, *Introduction to Algorithms*, vol. 6. Cambridge, MA, USA: MIT Press, 2001.
[58] G. Brassard and P. Bratley, *Fundamentals of Algorithmics*, vol. 524. Englewood Cliffs, NJ, USA: Prentice-Hall, 1996.
[59] X. Lan, H. Mo, S. Chen, Q. Liu, and Y. Deng, "Fast transformation from time series to visibility graphs," *Chaos*, vol. 25, no. 8, Aug. 2015, Art. no. 083105.
[60] R. Zhang, B. Ashuri, Y. Shyr, and Y. Deng, "Forecasting construction cost index based on visibility graph: A network approach," *Phys. A, Stat. Mech. Appl.*, vol. 493, pp. 239–252, Mar. 2018.
[61] M. T. Chou, "Long-term prediction value interval with the fuzzy time series," *J. Mar. Sci. Technol.*, vol. 19, no. 5, pp. 509–513, 2011.
[62] U. Yolcu, E. Egrioglu, V. R. Uslu, M. A. Basaran, and C. H. Aladag, "A new approach for determining the length of intervals for fuzzy time series," *Appl. Soft Comput.*, vol. 9, no. 2, pp. 647–651, Mar. 2009.
[63] H. K. Pathak and P. Singh, "A new bandwidth interval based forecasting method for enrollments using fuzzy time series," *Appl. Math.*, vol. 02, no. 04, pp. 504–507, 2011.
[64] S. S. Gangwar and S. Kumar, "Probabilistic and intuitionistic fuzzy sets–based method for fuzzy time series forecasting," *Cybern. Syst.*, vol. 45, no. 4, pp. 349–361, 2014.