Hindawi
Complexity
Volume 2019, Article ID 4132485, 12 pages
https://doi.org/10.1155/2019/4132485

Research Article

Stock Price Pattern Prediction Based on Complex Network and Machine Learning

Hongduo Cao, Tiantian Lin, Ying Li, and Hanyu Zhang

Business School, Sun Yat-sen University, Guangzhou 510275, China

Correspondence should be addressed to Hongduo Cao; caohd@mail.sysu.edu.cn and Ying Li; mnsliy@mail.sysu.edu.cn

Received 7 March 2019; Accepted 14 May 2019; Published 28 May 2019

Guest Editor: Benjamin M. Tabak

Complex networks in stock market and stock price volatility pattern prediction are the important issues in stock price research. Previous studies have used historical information regarding a single stock to predict the future trend of the stock's price, seldom considering comovement among stocks in the same market. In this study, in order to extract the information about relation stocks for prediction, we try to combine the complex network method with machine learning to predict stock price patterns. Firstly, we propose a new pattern network construction method for multivariate stock time series. The price volatility combination patterns of the Standard & Poor's 500 Index (S&P 500), the NASDAQ Composite Index (NASDAQ), and the Dow Jones Industrial Average (DJIA) are transformed into directed weighted networks. It is found that network topology characteristics, such as average degree centrality, average strength, average shortest path length, and closeness centrality, can identify periods of sharp fluctuations in the stock market. Next, the topology characteristic variables for each combination symbolic pattern are used as the input variables for K-nearest neighbors (KNN) and support vector machine (SVM) algorithms to predict the next-day volatility patterns of a single stock. The results show that the optimal models corresponding to the two algorithms can be found through cross-validation and search methods, respectively. The prediction accuracy rates for the three indexes in relation to the testing data set are greater than 70%. In general, the prediction ability of SVM algorithms is better than that of KNN algorithms.

1. Introduction

Stock price volatility patterns classification and prediction is a very important problem in stock market research. The prediction of stock price trends is actually a classified prediction of stock price fluctuation patterns [1]. Literature showed that forecasting stock price patterns is sufficient to generate profitable trades and enable the execution of profitable trading strategies [2]. Therefore, many studies have focused on predicting stock price patterns rather than predicting the absolute prices of stocks [2–4].

To date, most studies have focused on the volatility patterns of a single stock based on its own historical attributes [5, 6] and have paid less attention to the comovement of related stocks and information pertaining to the overall market. A few studies have used historical information regarding related stocks as the input variables for prediction and shown that the price fluctuations in a single stock are not isolated and are often influenced by the trends of multiple related stocks [7, 8]. Thus, how to extract the comovement of multiple stocks and apply this information to the prediction of the fluctuation patterns of a single stock is a problem worth studying.

Complex network analysis provides a new explanation for stock market behavior from a systematic perspective. Using complex network theory to study stock prices not only allows us to analyze the relationship between different stocks, but also allows us to explore the macroaspects of the comovement characteristics of the market in different periods [9–11]. Previous studies have proposed a variety of methods to build complex networks using the time series of stock prices, including visibility graphs [12–14], recurrence networks [15–17], correlation networks [11, 18, 19], pattern networks [10, 20], and K-neighbors networks [21, 22]. Of all the network construction methods, the symbolic pattern network is favored by many scholars because it can more accurately reflect the degree of correlation and direction of the primitive elements in a complex system [10, 20, 23, 24]. In a stock price volatility pattern network, each volatility pattern is regarded as a network node, and the relationship between patterns is regarded as a connection between nodes [10]. By analyzing the topological properties of the network, the characteristics of stock price fluctuations can be better understood. Huang et al. used coarse-grained symbolization methods to construct a network of market prices and transaction volume data in different periods based on the Shanghai Stock Exchange (SSE) composite index, and the results showed that the out-degree distribution of network nodes obeyed the power law and the basic fluctuations exhibited different patterns during different periods [24]. Wang et al. converted the yields of gasoline and crude oil stocks into five patterns and studied the characteristics of crude oil and gasoline node networks in different periods using sliding windows and then accurately predicted the crude oil and gasoline stock price pattern based on the conversion characteristics of the price network [10, 20].

However, most of the existing studies on stock price volatility pattern networks have focused on univariate time series. On this basis, we propose a new network construction method to build the volatility pattern networks of the three most important indexes in the US stock market, namely, the Standard & Poor's 500 Index (S&P 500), the NASDAQ Composite Index (NASDAQ), and the Dow Jones Industrial Average (DJIA). Firstly, the combination symbolic patterns for the three stock indexes are derived using a coarse-grained method. Then, the combination symbolic patterns are used as the nodes of the network, and the frequencies and directions of the conversion of the patterns are used as the weights and directions of the network connections. Finally, we construct directed and weighted networks for the US stock market. By analyzing the network topology properties, we can identify periods of sharp fluctuations in the market.

Meanwhile, many machine learning algorithms have been applied to stock price volatility classification and prediction, such as neural networks [25], random forests [26], decision trees [27], support vector machines (SVM) [3, 7], and K-nearest neighbors (KNN) [1, 28]. Among them, K-nearest neighbors (KNN) and support vector machine (SVM) algorithms have been widely used in pattern recognition and forecasting, machine learning, information retrieval, and data mining. KNN is a simple and effective classification method that is easy to calculate and its performance is comparable to the most advanced classification methods [29, 30]. SVM, which can map nonlinear separable data into high-dimensional space and use hyperplanes for classification, is highly suitable for small sample classification because of its excellent classification ability [26]. Both KNN and SVM algorithms have a mature theoretical basis in relation to classification prediction. Ballings et al. also compared the accuracy of SVM, KNN, and other algorithms in predicting stock price movements one year ahead for 5767 publicly listed European companies, and the results showed that SVM has the better prediction ability than KNN [2]. Teixeira proposed an automatic stock trading method that combined technical analysis with KNN classification. Using 15 stocks from Sao Paulo Stock Exchange (Bovespa), they found that the proposed method generated considerably higher profits than the buy-and-hold method for most of the companies, with few buy actions generated [1]. Huang et al. used SVM algorithms to predict the weekly fluctuations in the Nikkei 225 index and found that SVM outperformed the other classification methods, such as quadratic discriminant analysis and Elman backpropagation neural networks [3].

Literature has demonstrated the ability of SVM and KNN to predict stock patterns. However, they predicted the stock price based on the information of the single stock itself, without considering the information of the network system composed of the relevant stocks. Therefore, another aim of this study is to predict the next-day pattern of a single stock for each combination mode of stocks using the network topology properties as input variables for SVM and KNN algorithms. To the best of our knowledge, this should be the first attempt in existing research. Then, we compare the prediction accuracy using the testing data set after identifying the best models using the training set. The stock price volatility pattern network includes price information for single stocks and related stocks and portrays the macronature of the market, which contains more information than is available using only historical information relating to single stocks. The results show that the pattern network can provide some information to enable us to forecast the price volatility patterns of single stocks. Of the two prediction methods, the optimal parameter search strategy combined with cross-validation and search methods enables us to find the models that perform well on the testing data set. Overall, the performance of SVM algorithms is better than that of KNN algorithms. Combining with complex network and machine learning can provide investors with information on profitability strategies.

The remainder of this paper is organized as follows. In the next section, we introduce the theoretical background for KNN and SVM algorithms. In Section 3, the methodology of constructing the network and of predicting the next-day patterns for each stock index is presented. In Section 4, we show the empirical results and compare the prediction accuracy for KNN and SVM. The last section is devoted to a summary.

2. Theoretical Background for KNN and SVM

2.1. KNN. K-nearest neighbor (KNN) algorithm is a non-parametric classification algorithm that assigns query data to be classified to the category to which most of its K neighbors belong [31]. We use the Euclidean distance metric to find K-nearest neighbors from a sample set of known classifications. Suppose that the known data set has four feature variables $\{f_1, f_2, f_3, f_4\}$ and four categories $\{y_1, y_2, y_3, y_4\}$. The steps to search the category of the new data i through the KNN algorithm are as follows.

Firstly, the Euclidean distance of the feature variables of the data i and the other data j $(j = 1, 2, \ldots, \mathrm{n})$ in the training data set is calculated:

$$D_j = \sqrt{\sum_{g=1}^{4} \left(f_g(i) - f_g(j) \right)^2}, \quad j = 1, 2, \ldots, \mathrm{n}. \tag{1}$$

Secondly, all the data in the training set are sorted in ascending order according to the distance from data i.

Thirdly, K data points with the smallest distance from data i are selected.

Finally, the category with the largest proportion of these K data points will be considered as the category of data i.

An important parameter to be determined in the KNN algorithm is K, which represents the number of the nearest neighbors to be considered when classifying unknown samples [1, 2].

2.2. SVM. SVM was introduced by Vapnik [32] and has been widely used in pattern prediction in recent years. The basic idea of SVM is to nonlinearly transform the input vector into a high-dimensional feature space and then search the optimal linear classification surface in this feature space to maximize the distance between the classification plane and the nearest point. The training samples closest to the classification plane are called support vectors. SVM algorithm can be briefly described as follows.

Consider the binary linear classification problem of training data set $(\boldsymbol{x_i}, y_i)$ (i $=$ 1, 2, . . . , n), $\boldsymbol{x_i} \in R^n$; $y_i \in \{\pm 1\}$, where $\boldsymbol{x_i}$ is a feature vector and y_i is a class label. Suppose these two classes can be separated by a linear hyperplane $(\mathbf{w} \cdot \mathbf{x}) + \mathrm{b} = 0$. In order to make the correct classification and get the largest classification interval, the optimization problem of constructing the optimal plane is described as

$$\begin{aligned} &\min \quad \varphi(\mathbf{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 = \frac{1}{2}\left(\boldsymbol{w}' \cdot \boldsymbol{w}\right), \\ &\text{s.t.} \quad y_i\left[\left(\mathbf{w} \cdot \mathrm{x}_i\right) + \mathrm{b}\right] \geq 1. \end{aligned} \tag{2}$$

The optimal solution of $\mathbf{w}$ and b can be solved by introducing the Lagrange multiplier. Then we can obtain the optimal classification problem like (3).

$$f(x) = \operatorname{sgn}\left\{\boldsymbol{w}^* \cdot \mathbf{x} + \boldsymbol{b}^*\right\}. \tag{3}$$

For a nonlinear classification problem, the feature vector is transformed into high-dimensional space vector firstly. Then the optimal classification hyperplane is constructed. Suppose the transformation function is $\mathbf{\Phi}$, then the optimal problem can be described as

$$\begin{aligned} &\min \quad \varphi(\mathbf{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i = \frac{1}{2}\left(\boldsymbol{w}' \cdot \boldsymbol{w}\right) + C\sum_{i=1}^{N}\xi_i, \\ &\text{s.t.} \quad y_i\left[\left(\mathbf{w}^{\boldsymbol{T}} \cdot \mathbf{\Phi}\left(x_i\right)\right) + \mathrm{b}\right] + \xi_i \geq 1, \\ &\qquad \xi_i \geq 0, \quad \mathrm{i} = 1, 2, \ldots, \mathrm{n}, \end{aligned} \tag{4}$$

where C is the penalty parameter, which specifies the trade-off between classification distance and misclassification [2]. Finally, the optimal classification hyperplane can be described in (5).

$$f(x) = \operatorname{sgn}\left\{\boldsymbol{w}^* \cdot \mathbf{\Phi}(\mathbf{x}) + \boldsymbol{b}^*\right\}. \tag{5}$$

$$k\left(x_i, x_j\right) = \mathbf{\Phi}^{\boldsymbol{T}}\left(x_i\right) \cdot \mathbf{\Phi}\left(x_j\right). \tag{6}$$

The function (6) is called a kernel function. Because the performance of the Gaussian radial basis function (RBF) is excellent when the additional information of the data is limited, it is widely used in the financial time series analysis [3]. The Gaussian radial basis function (RBF) is used as the kernel function to implement the SVM algorithm in this study. The RBF kernel function can be expressed as

$$k\left(x_i, x_j\right) = \exp\left(-\sigma\left\|x_i - x_j\right\|^2\right), \quad \sigma > 0, \tag{7}$$

where σ is the constant of the radial basis function. Before implementing the SVM algorithm, the parameter σ and parameter C need to be determined.

For multiclassification problem, it can be converted into multiple two-classification problems [33]. In this study, a four-classification problem is transferred into six two-classification problems by the “one-versus-one” approach of SVM.

3. Methodology

In this section, we introduce the methodology for predicting stock price patterns using network topology characteristic variables. Figure 1 shows a general framework of the proposed pattern prediction system. It consists of two parts: complex network analysis and pattern prediction using machine learning. We present a more detail procedure in the subsections.

3.1. Constructing a Pattern Network for the Stock Market. Using the daily closing price of each stock index, a sliding window is used to calculate the one-day return r, five-day return R, and five-day volatility V corresponding to day t:

$$r = \ln\frac{Close\,(t)}{Close\,(t-1)}, \tag{8}$$

$$R = \ln\frac{Close\,(t)}{Close\,(t-5)}, \tag{9}$$

$$\mathrm{V} = std\left(r_1, \ldots, r_5\right) * \sqrt{5}, \tag{10}$$

where $Close(t)$ is the closing price on day t, $Close(t-1)$ is the previous day’s closing price, and $std(r_1, \ldots, r_5)$ is the standard deviation of the yield from the first to the fifth day.

Then, we can calculate the average five-day volatility $\overline{V}$ for each stock index:

$$\overline{V} = \frac{1}{N}\sum V, \tag{11}$$

where N is the number of trading days in a time series. Suppose we study M stocks in the stock market. Then, we can obtain the average volatility for the overall market as follows:

$$\mathrm{V}' = \frac{1}{M}\sum_{i=1}^{M}\overline{V}_i. \tag{12}$$

Based on the sign of the five-day return R and the magnitude of the five-day volatility V each day, each stock can be classified into one of four patterns:

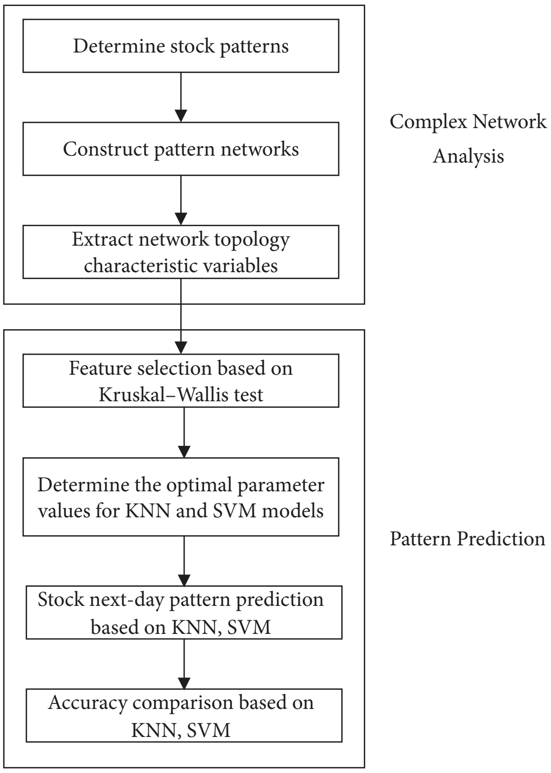

Determine stock patterns
Construct pattern networks
Extract network topology characteristic variables
Complex Network Analysis
Feature selection based on Kruskal–Wallis test
Determine the optimal parameter values for KNN and SVM models
Stock next-day pattern prediction based on KNN, SVM
Accuracy comparison based on KNN, SVM
Pattern Prediction

FIGURE 1: General framework of the stock index pattern prediction.

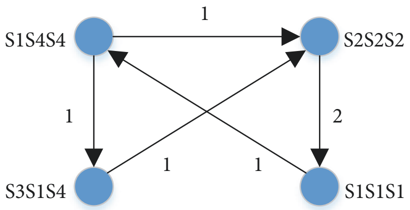


FIGURE 2: Sample directed weighted network.

$$f = \begin{cases} S1, & \mathit{if}\ R \geq 0\ \mathit{and}\ V \geq \mathrm{V}'\ \ (\text{sharp rise}) \\ S2, & \mathit{if}\ R \geq 0\ \mathit{and}\ V < \mathrm{V}'\ \ (\text{stable rise}) \\ S3, & \mathit{if}\ R < 0\ \mathit{and}\ V < \mathrm{V}'\ \ (\text{stable decline}) \\ S4, & \mathit{if}\ R < 0\ \mathit{and}\ V \geq \mathrm{V}'\ \ (\text{sharp decline}). \end{cases} \tag{13}$$

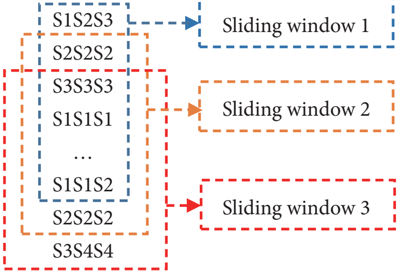
By combining the patterns of each stock index, we can obtain the corresponding combination symbolic pattern for each day. Assuming that we study three stock indexes, we can obtain a maximum of $4^3 = 64$ combination modes. Taking the daily combination patterns as the nodes of the network, the edges and weights of the network can be determined in time order. If the pattern on day t is *S1S4S4* and that on day t+1 is *S2S2S2*, there is a directed edge from *S1S4S4* to *S2S2S2* with a weight of 1. If the conversion frequency from *S1S4S4* to *S2S2S2* is w, the weight of the directed edges from *S1S4S4* to *S2S2S2* is w. For example, if the current patterns of the S&P 500, NASDAQ, and DJIA are $S1$, $S3$, and $S4$, respectively, the current price combination pattern is *S1S3S4*. Suppose that the pattern transformation over a certain period of time is {*S1S4S4, S2S2S2, S1S1S1, S1S4S4, S3S1S4, S2S2S2, S1S1S1*}. Then, we can obtain the directed weighted network shown in Figure 2.

The key to the sliding window selection problem is how to effectively keep the quality and quantity of original time series information while reducing the computational complexity to the most extent [34]. In this study, we apply a sliding window with a length of 30 days (about one month in daily life and half a quarter in the stock market) and a step of one day to the stock indexes time series. So we can obtain a pattern network every 30 days. Table 1 shows the process of using the sliding window.

3.2. Computing Network Topology Characteristic Variables. Next, we calculate the network topology characteristic variables for every 30-day network.

Table 1: The process of using the sliding window.

Date	Stock 1	Stock 2	Stock 3	Combination Pattern
1	S1	S2	S3	S1S2S3
2	S2	S2	S2	S2S2S2
3	S3	S3	S3	S3S3S3
4	S1	S1	S1	S1S1S1
…	…	…	…	…
30	S1	S1	S2	S1S1S2
31	S2	S2	S2	S2S2S2
32	S3	S4	S4	S3S4S4
…	…	…	…	…



3.2.1. Network Average Degree Centrality. In undirected networks, the average degree centrality of the network reflects the level of connection between one node and other nodes in the network, that is, whether one node is connected with the other nodes or not [17]. The formula is as follows:

$$\rho = \frac{1}{N(N-1)} \sum_{i,j=1}^{N} a_{ij}, \tag{14}$$

where N is the number of nodes in the network and $a_{i,j}$ is the value of the adjacency matrix of an undirected network. $a_{i,j} = 1$ if node i and node j are connected, otherwise $a_{i,j} = 0$. The adjacency matrix of an undirected network is a symmetric matrix. However, $a_{i,j} = 1$ does not mean that $a_{j,i} = 1$ in a directed network. In the directed network, we must consider the out-degree and in-degree. We connect the nodes in time order so that the in-degree and the out-degree are the same except for the first node and the last node. Therefore, we only select the in-degree for analysis, and calculate the average in-degree centrality as follows:

$$\rho = \frac{1}{N^2} \sum_{j=1}^{N} \sum_{i=1}^{N} a_{ij}. \tag{15}$$

In terms of narration, in the next sections, we describe average in-degree centrality as average degree centrality. Average degree centrality measures the ratio of the actual number of connections to the maximum number of connections, that is, the edge density of the network. The greater the average degree centrality, the more connections between nodes in the price pattern network, the higher the accessibility between nodes, and the greater the density of the overall network [35].

3.2.2. Average Network Strength. In a network, the strength of the connection from node i to node j is the weight w_{ij} of the directed edge from node i to node j. Similar to the in-degree and out-degree of the directed network, the strength of the directed weighted network can also be divided into in-strength and out-strength [10, 35]. In this study, we describe average out-strength as average network strength:

$$S = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij}. \tag{16}$$

The greater the average strength of the network, the fewer the number of network nodes, the simpler the composition of the price volatility patterns, the smaller the complexity of the network, and the higher the frequency of the same node. The simpler price patterns reflect the fact that the consistency of price changes of different stocks is stronger and lasts for longer.

3.2.3. Network Average Shortest Path Length. The average shortest path length of the network describes the degree of separation between nodes in the network, that is, the size of the network. The average shortest path length can be used to characterize a “small-world” network in a complex network [17]. The distance from node i to node j is defined as the minimum number of edges needed to pass from node i to node j. The average shortest path length of the network is the average length of all the shortest paths in the network:

$$L = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij}, \quad (i \neq j). \tag{17}$$

The shorter the average shortest path, the less intermediate patterns are required for conversion between stock price modes. Modes can be connected by fewer edges, and price modes can interact with each other through fewer other modes. As a result, the conversion efficiency and speed of the overall network are both greater.

3.2.4. Network Closeness Centrality. The closeness centrality of node i is the reciprocal of the average shortest path length from other nodes to node i [15, 36]:

$$C_i = \frac{N-1}{\sum_{j=1}^{N} d_{ji}}. \tag{18}$$

The closer a point is to other points, the easier it is to transmit information. Now, we consider the weighted shortest path l_{ij}, which is defined as the shortest weighted distance from node i to node j. Then, we can obtain the closeness centrality of the network [37]:

$$C = \frac{1}{N} \sum_{i=1}^{N} \frac{n-1}{N-1} \frac{n-1}{\sum_{j=1}^{n-1} l_{ij}}, \quad (i \neq j). \tag{19}$$

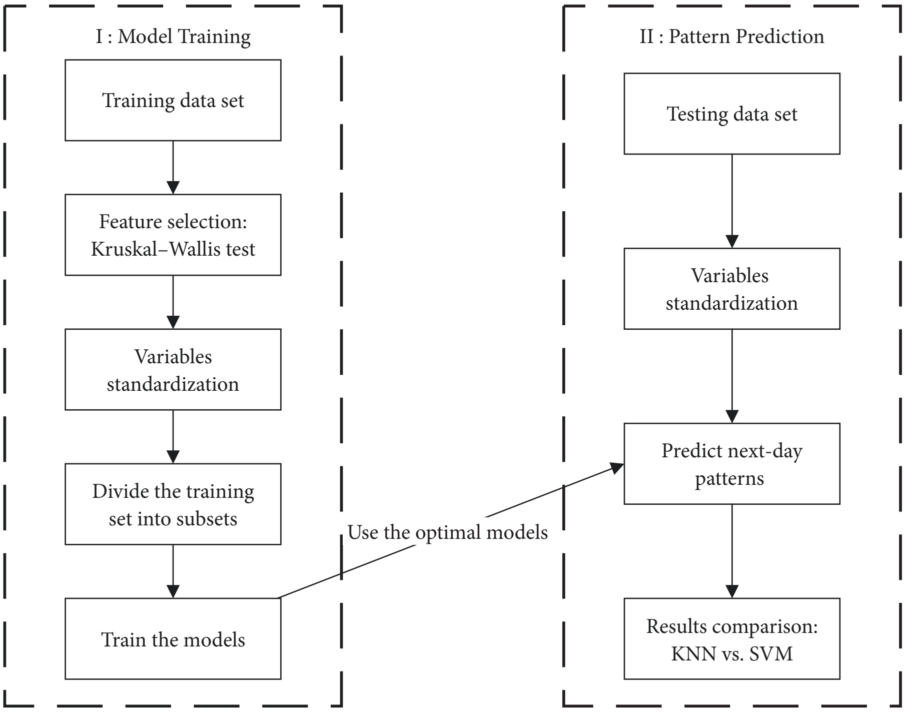

I : Model Training
Training data set
Feature selection: Kruskal–Wallis test
Variables standardization
Divide the training set into subsets
Train the models
Use the optimal models
II : Pattern Prediction
Testing data set
Variables standardization
Predict next-day patterns
Results comparison: KNN vs. SVM

Figure 3: The road map of the pattern prediction experiment.

where n is the number of nodes reachable from node i. The greater the closeness centrality of the network, the smaller the shortest weighted distance between the network nodes, the fewer the conversion times between different price modes, and the smaller the conversion cycle. Pattern nodes tend to transform on their own, and thus the transformation area of the overall network is more concentrated and centrality is more prominent [35].

3.3. Next-Day Stock Pattern Prediction Based on KNN and SVM. We train the optimal prediction models based on KNN and SVM algorithms by the obtained network topology characteristic variables, and then predict next-day patterns of three single stock indexes using the testing data set.

3.3.1. Detail Prediction Procedure. Figure 3 shows the detail process of the pattern prediction experiment for each single stock index. It includes two major steps: the first step is model training, from which we can get the best models; and the second step is pattern prediction.

First, the correlation between the second day's stock patterns and the network characteristic variables of a single stock index in the training set is tested by the Kruskal–Wallis test. The network topological characteristic variables which are significantly correlated with the price pattern of each stock index will be the input variables for the stock index prediction.

The values of topological characteristic variables are normalized so that a smaller valued indicator does not be ignored because of an indicator with larger value [26]. Formula (20) is used to standardize the variables [38]:

$$\hat{y}_i = \frac{y_i - y_{min}}{y_{max} - y_{min}}. \tag{20}$$

Next, in order to get a subset of each combination mode, we divide the training set into several training subsets according to the number of types of combination patterns (or combination pattern nodes) in the training set. Obviously, in each subset, the day's combination pattern is the same, but the next-day patterns of each stock index can be different. For each training subset, the next-day patterns of a single stock index are the classification variables, and the network topological characteristic variables are the input feature variables for the KNN and SVM algorithms. To prevent overfitting, the cross-validation and search method is used to determine the optimal parameters in this study.

Finally, on the testing data set, the next-day patterns of each stock index are predicted by the obtained models.

The optimal training models are found according to the recognized combination pattern, and the next-day stock patterns are predicted on the basis of the topological characteristic variables of the current corresponding 30-day network. The average market volatility and the standardization parameters used for the testing set were obtained through the training set.

3.3.2. Model Selection Criteria. Cross-validation is widely used for model selection because of its simplicity and universality, so we use cross-validation method and search

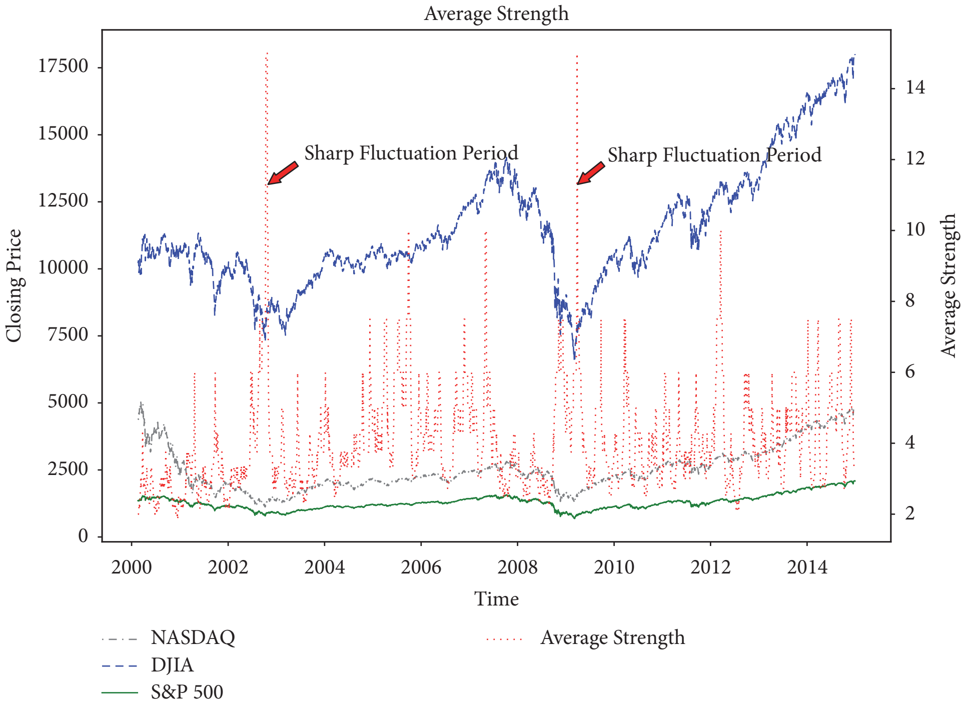


Figure 4: The evolution of the three stock indexes and average strength.

method to determine the optimal parameters in this study [39]. We cross-validated the K-parameters for KNN by trying all values of $K = \{1, 2, 3, \ldots, 30\}$. To determine the optimal parameter values for SVM, we perform a grid search on $C = \{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$, $\sigma = \{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$ to identify the best combination.

Using the k-fold cross-validation method, for each combination of parameters, the training data set is divided into a subset with k equal parts, and k–1 parts of the data are used as the training data, while the other part is used as the verification data. In this way, the accuracy rates of k verification sets can be obtained after k iterations. Taking the average accuracy rate of the k verification sets, that is the verification score, as the criterion for parameter selection, the optimal combination of parameters can be found. In addition, leave-one-out (LOO) is another simple, efficient, and common cross-validation method. When using LOO, one sample is taken from the data set each time as a validation set, and the other samples are used as the training set. Thus, for a data set with n samples, we can get a total of n different test sets and their corresponding training sets. LOO is very suitable for model selection of small samples because only one sample is extracted from the training set at a time as a verification set so that fewer samples are wasted [40].

In this study, we use 3-fold cross-validation if the training subset contains more than 100 samples and use LOO cross-validation otherwise.

4. Empirical Results and Analysis

4.1. Data Processing. We used the closing prices of the S&P 500, NASDAQ, and DJIA from 1 January 2000 to 31 December 2014 as the sample data set. This resulted in 3769 daily records. The data were obtained from the Wind database, one of the most authoritative financial database in China (the Wind database can be downloaded from https://www.wind.com.cn/.). First, the five-day return rate and five-day volatility of each stock index are calculated. The method outlined in the previous section is used to symbolize the stock index, and then we obtain the combination patterns for the three stock indexes each day.

A sliding window with a length of 30 days and a step of one day is used to divide the stock pattern time series into 3740 time periods. A directed weighted network is constructed for each period of stock price patterns, resulting in 3740 networks. There are 47 pattern nodes in all of the networks.

The method we construct the stock networks is original, so we used Python for coding. We used some functions in the Python 3.7 standard library including networkx, sklearn, pandas, and matplotlib for our analysis.

4.2. Analysis of Network Topological Characteristics. The average degree centrality, average strength, average shortest path length, and closeness centrality of each network are calculated using formulas (15), (16), (17), and (19). The evolution of these four network topological characteristics is shown in Figures 4–7.

As can be seen from the figures, the points where average degree centrality, average intensity, and closeness centrality reach their peak value and the average shortest path length reaches its minimum value all correspond to periods when the overall market is fluctuating wildly. When

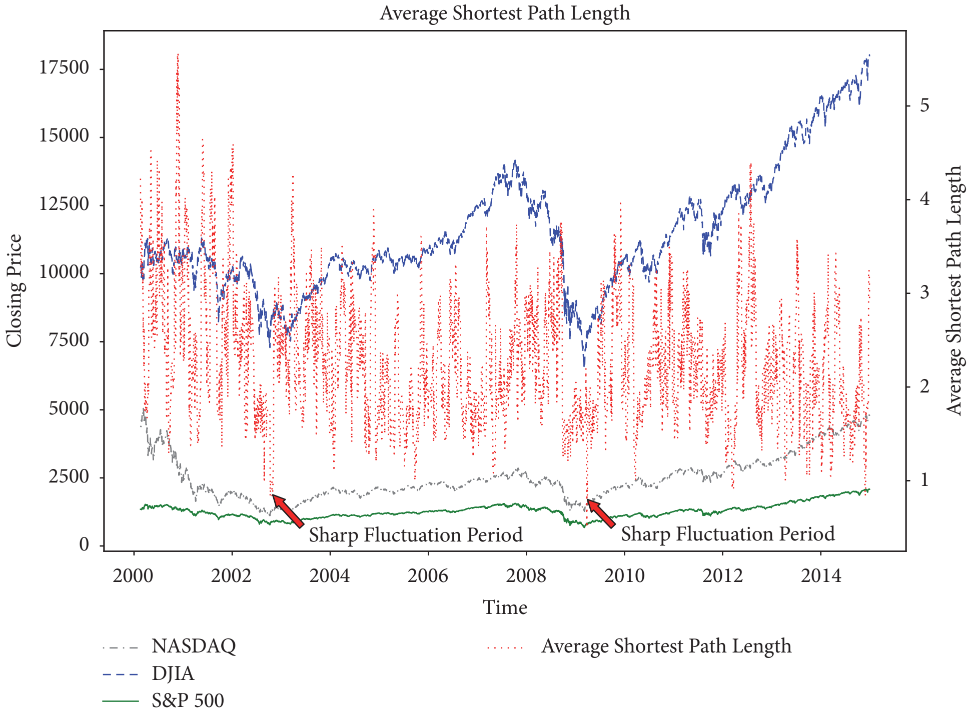


FIGURE 5: The evolution of the three stock indexes and average shortest path length.

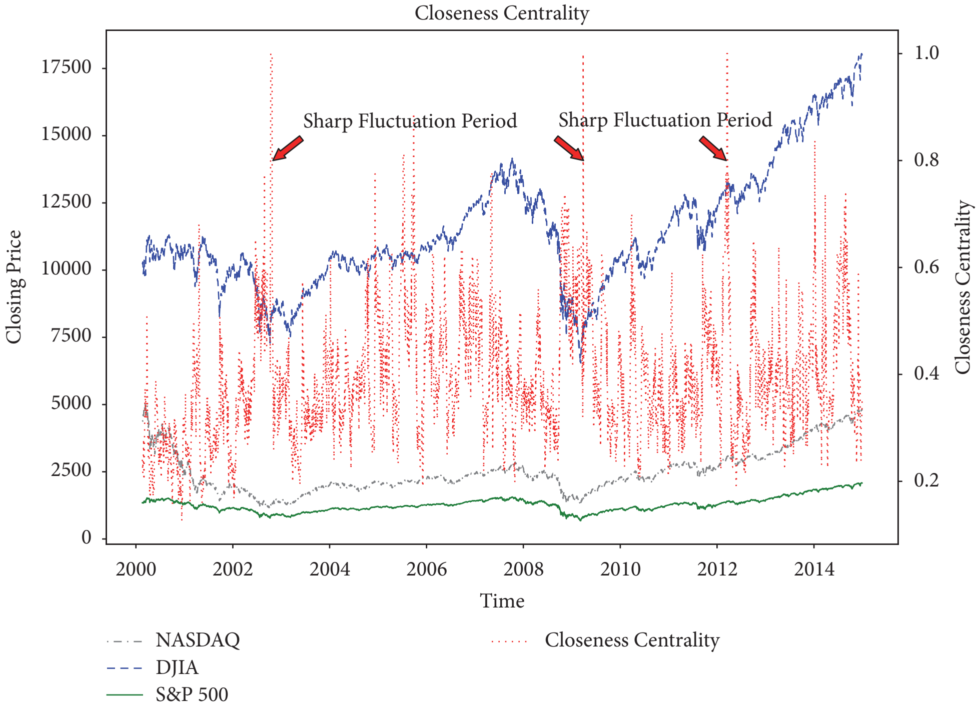


FIGURE 6: The evolution of the three stock indexes and closeness centrality.

the three stock indexes fell to their lowest levels in October 2002 and March 2009, the average degree centrality, average strength, and closeness centrality of the network reached their highest points, while the shortest path length reached its lowest point. These two periods correspond to the last phase of the dot-com bubble crisis and the subprime mortgage crisis. In addition, the closeness centrality and the average degree centrality reached their maximum points again in March 2012, which corresponded with another long period of sharp fluctuations in the US stock market. The results

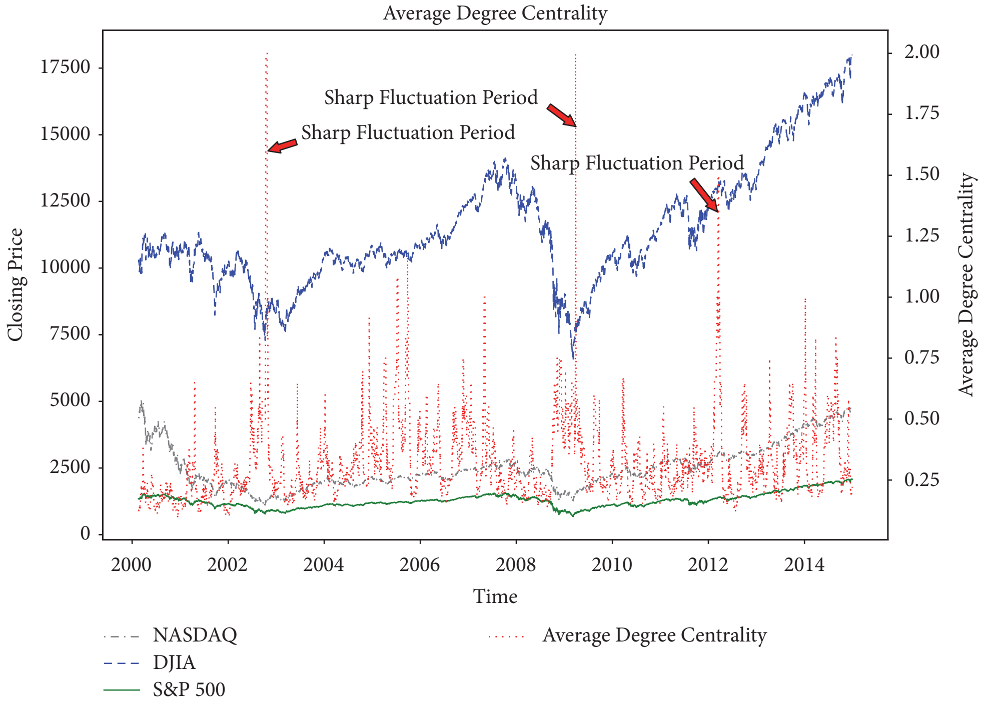


Figure 7: The evolution of the three stock indexes and average degree centrality.

show that these four network topological characteristics have a remarkable relationship to the anomaly of the indexes in the US stock market. During the sharp fluctuation periods, the comovement of three stock indexes is stronger, and the 30-day networks are simpler.

The maximum value of the average strength of the network reflects the fact that there are relatively fewer nodes in the stock pattern network, and the fluctuation modes of stocks are monotonous. It shows that in the month before the extreme values, the price fluctuations of the three stock indexes were synchronous, resulting in the relative simplicity of the volatility combination pattern. Before the three indexes reached their lowest levels in 2002 and 2009, they were basically in a state of substantial decline. The correlation and consistency of the indexes reached their maximum during this period, and so the average strength of the network reached its maximum, which is consistent with the findings of previous studies on complex networks using price time series [9, 41].

In addition, when the stock market is in a period of dramatic fluctuations, the node types in the network are monotonous, the stock network is constantly switching between several price models, and the edge density is larger, so the average degree centrality reaches its maximum. During this period, the nodes are more compact, the transformation between nodes is faster, fewer edges need to be passed, and the average shortest path of the network reaches its minimum. Although other modes may emerge during this period, the large fluctuation mode occupies the most important position, and the price patterns tend to shift between the main patterns so that reaching maximum closeness centrality. This conclusion is the same as that of Wang et al. [10].

4.3. Next-Day Stock Pattern Prediction Using KNN and SVM Algorithms. By analyzing the 30-day network topological characteristics corresponding to each trading day, we find that the extreme values of the network topological characteristics can reflect the periods of dramatic fluctuations in the system composed of the three stock indexes. KNN and SVM algorithms are used to predict the next-day patterns of each stock index when the combination patterns of the three stock indexes and the corresponding 30-day network topological characteristics for the current day are known.

Based on the theory of cross-validation [42], and in order to keep the year intact and ensure the continuity of the years, we used the closing prices of the S&P 500, NASDAQ, and DJIA from 1 January 2000 to 31 December 2014 as the training sample data set. The testing sample data set used the closing prices of the three indices from 1 January 2015 to 31 December 2017. The training set and the testing set contained 3769 and 755 records, respectively. Since the training sample set has 47 pattern nodes, we divided the training data into 47 training subsets.

4.3.1. Kruskal–Wallis Tests to Filter Variables. Following the methods used to select variables in existing studies, we used the Kruskal–Wallis test to filter the four network topological characteristic variables and next-day patterns of each stock index using the training samples [43]. The results are shown in Table 2.

It can be seen from Table 2 that the p-values of the variables and the next-day patterns of the various stock indexes are all less than or equal to 0.1 except for the next-day patterns of the DJIA, where closeness centrality is not significant. Therefore, when predicting the next-day patterns

Table 2: Kruskal–Wallis tests of the four network topological characteristic variables and next-day patterns of each stock index.

	S&P500 stock index	NASDAQ stock index	DJIA stock index
Average strength	43.0578***	340.2039***	29.8488***
	(0.001)	(0.001)	(0.001)
Average shortest path length	9.3241**	28.1597***	7.6535*
	(0.0253)	(0.001)	(0.0537)
Average degree centrality	25.4071***	250.9374***	13.0833***
	(0.001)	(0.001)	(0.0045)
Closeness centrality	8.4532**	190.0352***	0.6204
	(0.0375)	(0.001)	(0.8917)

Note: figures in parentheses are p-values.

Table 3: Prediction accuracy of the optimal KNN and SVM models in relation to next-day patterns for the three stock indexes using the testing set.

Closeness	Algorithms	DJIA stock index	S&P500 stock index	NASDAQ stock index
Yes	KNN	74.83%	72.58%	72.45%
	SVM	74.97%	73.11%	74.57%
No	KNN	72.98%	70.59%	70.86%
	SVM	76.03%	72.45%	73.64%

of the DJIA, the closeness centrality is removed, leaving the three other variables as input variables for the KNN and SVM algorithms. When predicting the next-day patterns of the S&P 500 and the NASDAQ, all four network topological characteristic variables are retained as input variables.

4.3.2. Predicting Stock Patterns Using KNN and SVM Algorithms. The accuracy of prediction is defined as:

$$\text{Accuracy rate} = \frac{\text{The number of correct prediction}}{\text{Total number of sample set}} \times 100\%. \qquad (21)$$

We compare the predicted next-day patterns with the actual next-day patterns of the stock index. If they are the same on a given day, we can say that our prediction is correct. The proportion of the number of correctly predicted samples to the total number of samples is the accuracy rate. The accuracy rate is close to 1 means that the models yield more accurate predictions, whereas the accuracy rate is close to 0 means that the models are less accurate.

After obtaining the optimal models using KNN and SVM algorithms in relation to the training set using the cross-validation and search methods, the models are used to predict patterns using the testing set, and their performance is evaluated based on their prediction accuracy rates. Table 3 shows the prediction accuracy of the optimal models obtained using KNN and SVM algorithms for the three stock indexes.

From Table 3, it can be seen that KNN and SVM algorithms can identify appropriate models based on the training set using the cross-validation and search methods, with prediction accuracies in relation to the testing set of greater than 70%. However, generally, the prediction accuracy of SVM algorithms is higher than that of KNN algorithms. It is similar to the findings of previous studies on the two algorithms; that is, the generalization ability of the SVM classification model is greater than that of the KNN model [1, 2]. To further illustrate the predictive effect of closeness on the three stock indexes, we compare the prediction accuracy rate in the cases of closeness and no closeness. We find an interesting result wherein the prediction model without closeness using SVM has the highest prediction accuracy rate when predicting the next-day patterns of the DJIA stock index. This result indicates that SVM is more accurate and sensitive than KNN. Closeness does not affect predictions regarding the DJIA, as the results of the Kruskal–Wallis test show.

Diether et al. examined short-selling in US stocks using SEC-mandated data for 2005 and found that short-selling activity was strongly positively correlated with previous five-day returns and volatility [44]. The five-day movement of stocks is also very important for short-term investment in stocks or funds in the real world. Thus, if the investor can forecast future five-day volatility patterns, more information can be obtained to support short-selling strategies. For instance, if the next five-day pattern is predicted to be S1 (sharp rise), the investor can execute a short-selling strategy the next day.

5. Conclusion

Based on the complex network method, this study analyzes the stock price fluctuation patterns of the three most important stock indexes for the US stock market. Unlike previous studies, this study uses the three stock indexes to build pattern networks for the system, rather than using a single stock index. From the analyses of the average strength, average shortest path length, average degree centrality, and closeness centrality of the price pattern network every 30 days, it is found that when the overall stock market is in a period of dramatic fluctuations, the average strength, average degree centrality, and closeness centrality reach their maximum values, while the average shortest path length reaches its minimum value. This shows that price volatility pattern networks can reflect special periods on the stock market. In periods of dramatic fluctuations on the stock market, the comovement of various indexes is stronger, the edge density of the corresponding pattern network is greater, the conversion between price modes is faster, and the conversion area of nodes is more concentrated. It shows the validity of using price pattern network characteristics to identify special periods on the stock market. To a certain extent, they can reflect abnormal periods on the stock market from a macropoint of view. When the four indicators approach extreme values, investors should exercise caution.

The stock price network characteristic variables not only contain price change information for individual stocks, but also reflect the overall change characteristics of the market at the macrolevel. Therefore, another focus of this study is the use of the network characteristic variables as input variables for KNN and SVM algorithms to predict the next-day fluctuation patterns of individual stocks. Firstly, the Kruskal–Wallis test is used to test the next-day patterns of three stock indexes and four network characteristic variables, and we find that closeness does not affect predicting the next-day stock patterns of DJIA index. In the case of the combination price patterns for the current day, the network characteristic variables are used as the input variables for KNN and SVM algorithms to predict the next-day stock price patterns, and the accuracy of the two algorithms in relation to the testing set is compared. The results show that both the KNN and SVM algorithms display a high level of accuracy in predicting the next-day stock price patterns, with prediction accuracy of greater than 70% for all three stock indexes. However, the generalization ability of the SVM algorithm is greater than that of the KNN algorithm. Thus, it is possible to predict stock trends by using a proper classification algorithm and combining the structural characteristics of the multistock price network. This approach can generate more information for financial trading strategies in the real world and provides a new focus for future research into stock price prediction.

The application of the complex network method to the stock market is still in the developmental stage. Revealing the characteristics of stock price fluctuations by using complex networks is helpful in understanding the essence of stock price fluctuations and providing profit-making strategies. A more detailed examination of the correlation between financial crises and network topological properties is a worthy topic for further research. In addition, the combined use of machine learning and complex network methods to study the stock market deserves more in-depth discussion and diversified development.

Data Availability

The data used to support the findings of this study have been deposited in https://www.kesci.com/home/dataset/5c82706ed635ff002ca24a19.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported, in part, by the National Natural Science Foundation of China (Grants nos. 71371200, 71071167, and 71071168).

References

[1] L. A. Teixeira and A. L. I. De Oliveira, “A method for automatic stock trading combining technical analysis and nearest neighbor classification,” *Expert Systems with Applications*, vol. 37, no. 10, pp. 6885–6890, 2010.

[2] M. Ballings, D. Van Den Poel, N. Hespeels, and R. Gryp, “Evaluating multiple classifiers for stock price direction prediction,” *Expert Systems with Applications*, vol. 42, no. 20, pp. 7046–7056, 2015.

[3] W. Huang, Y. Nakamori, and S.-Y. Wang, “Forecasting stock market movement direction with support vector machine,” *Computers & Operations Research*, vol. 32, no. 10, pp. 2513–2522, 2005.

[4] Y.-W. Cheung, M. D. Chinn, and A. G. Pascual, “Empirical exchange rate models of the nineties: Are any fit to survive?” *Journal of International Money and Finance*, vol. 24, no. 7, pp. 1150–1175, 2005.

[5] G. Armano, A. Murru, and F. Roli, “Stock market prediction by a mixture of genetic-neural experts,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 5, pp. 501–526, 2002.

[6] A. H. Moghaddam, M. H. Moghaddam, and M. Esfandyari, “Stock market index prediction using artificial neural network,” *Journal of Economics, Finance and Administrative Science*, vol. 21, no. 41, pp. 89–93, 2016.

[7] R. Choudhry and K. Garg, *A Hybrid Machine Learning System for Stock Market Forecasting*, vol. 39, 2008.

[8] Y. K. Kwon, S. S. Choi, and B. R. Moon, “Stock prediction based on financial correlation,” in *Proceedings of the Conference on Genetic Evolutionary Computation*, 2005.

[9] L. Lacasa, V. Nicosia, and V. Latora, “Network structure of multivariate time series,” *Scientific Reports*, vol. 5, 2015.

[10] M. Wang, Y. Chen, L. Tian, S. Jiang, Z. Tian, and R. Du, “Fluctuation behavior analysis of international crude oil and gasoline price based on complex network perspective,” *Applied Energy*, vol. 175, pp. 109–127, 2016.

[11] B. M. Tabak, T. R. Serra, and D. O. Cajueiro, “Topological properties of stock market networks: the case of Brazil,” *Physica A: Statistical Mechanics and Its Applications*, vol. 389, no. 16, pp. 3240–3249, 2010.

[12] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nuno, “From time series to complex networks: the visibility graph,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 13, pp. 4972–4975, 2008.

[13] Z. Gao, Q. Cai, Y. Yang, W. Dang, and S. Zhang, “Multiscale limited penetrable horizontal visibility graph for analyzing nonlinear timeseries,” *Scientific Reports*, vol. 6, no. 1, Article ID 35622, 2016.

[14] E. Zhuang, M. Small, and G. Feng, “Time series analysis of the developed financial markets’ integration using visibility graphs,”

Physica A: Statistical Mechanics and its Applications, vol. 410, pp. 483–495, 2014.

[15] R. V. Donner, Y. Zou, J. F. Donges, N. Marwan, and J. Kurths, "Recurrence networks-a novel paradigm for nonlinear time series analysis," *New Journal of Physics* , vol. 12, no. 3, Article ID 033025, 2010.

[16] Y. Li, H. Caö, and Y. Tan, "Novel method of identifying time series based on network graphs," *Complexity*, vol. 17, no. 1, pp. 13–34, 2011.

[17] N. Marwan, J. F. Donges, Y. Zou, R. V. Donner, and J. Kurths, "Complex network approach for recurrence analysis of time series," *Physics Letters A*, vol. 373, no. 46, pp. 4246–4254, 2009.

[18] Y. Li, H. Cao, and Y. Tan, "A comparison of two methods for modeling large-scale data from time series as complex networks," *AIP Advances*, vol. 1, no. 1, Article ID 012103, p. 509, 2011.

[19] Y. Yang and H. Yang, "Complex network-based time series analysis," *Physica A: Statistical Mechanics and Its Applications*, vol. 387, no. 5-6, pp. 1381–1386, 2008.

[20] M. Wang, A. L. M. Vilela, L. Tian, H. Xu, and R. Du, "A new time series prediction method based on complex network theory," in *Proceedings of the 5th IEEE International Conference on Big Data, (Big Data) 2017*, pp. 4170–4175, December 2017.

[21] Y. Shimada, T. Kimura, and T. Ikeguchi, *Analysis of Chaotic Dynamics Using Measures of the Complex Network Theory*, Springer, Berlin, Germany, 2008.

[22] X. Xu, J. Zhang, and M. Small, "Superfamily phenomena and motifs of networks induced from time series," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 105, no. 50, pp. 19601–19605, 2008.

[23] Z. Ming, W. Er-Hong, Z. Ming-Yuan, and M. Qing-Hao, "Directed weighted complex networks based on time series symbolic pattern representation," *Acta Physica Sinica*, vol. 66, no. 21, Article ID 210502, 2017.

[24] W.-Q. Huang, S. Yao, and X.-T. Zhuang, "A network dynamic model based on SSE composite index and trading volume fluctuation," *Journal of Northeastern University*, vol. 31, no. 10, pp. 1516–1520, 2010.

[25] S. H. Kim and S. H. Chun, "Graded forecasting using an array of bipolar predictions: Application of probabilistic neural networks to a stock market index," *International Journal of Forecasting*, vol. 14, no. 3, pp. 323–337, 1998.

[26] 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.

[27] M.-C. Wu, S.-Y. Lin, and C.-H. Lin, "An effective application of decision tree to stock trading," *Expert Systems with Applications*, vol. 31, no. 2, pp. 270–274, 2006.

[28] M. V. Subha and S. T. Nambi, "Classification of stock index movement using k-nearest neighbours (k-NN) algorithm," *WSEAS Transactions on Information Science and Applications*, vol. 9, no. 9, pp. 261–270, 2012.

[29] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," *Artificial Intelligence Review*, vol. 11, no. 1–5, pp. 11–73, 1997.

[30] C.-J. Huang, D.-X. Yang, and Y.-T. Chuang, "Application of wrapper approach and composite classifier to the stock trend prediction," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2870–2878, 2008.

[31] S. Mitra and T. Acharya, *Data Mining: Multimedia, Soft Computing, and Bioinformatics*, John Wiley & Sons, 2005.

[32] V. Cherkassky, "The nature of statistical learning theory," *Technometrics*, vol. 38, no. 4, pp. 409-409, 1996.

[33] W.-M. Lin, C.-H. Wu, C.-H. Lin, and F.-S. Cheng, "Classification of multiple power quality disturbances using support vector machine and one-versus-one approach," in *Proceedings of the 2006 International Conference on Power System Technology, POWERCON2006*, China, October 2006.

[34] F. Li and J. Xiao, "How to get effective slide-window size in time series similarity search," *Journal of Frontiers of Computer Science & Technology*, vol. 3, no. 1, pp. 105–112, 2009.

[35] X. Sun, M. Small, Y. Zhao, and X. Xue, "Characterizing system dynamics with a weighted and directed network constructed from time series data," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 24, no. 2, Article ID 024402, 9 pages, 2014.

[36] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978-1979.

[37] A. W. Wolfe, "Social network analysis: methods and applications," *American Ethnologist*, vol. 24, no. 4, pp. 136-137, 1995.

[38] K. Shin, T. S. Lee, and H. Kim, "An application of support vector machines in bankruptcy prediction model," *Expert Systems with Applications*, vol. 28, no. 1, pp. 127–135, 2005.

[39] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, pp. 40–79, 2010.

[40] G. C. Cawley and N. L. C. Talbot, "Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers," *Pattern Recognition*, vol. 36, no. 11, pp. 2585–2592, 2003.

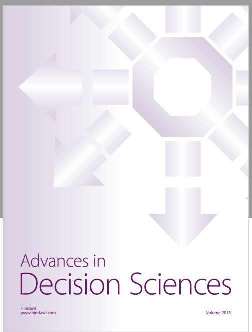
[41] L. Xia, D. You, X. Jiang, and Q. Guobc, "Comparison between global financial crisis and local stock disaster on top of Chinese stock network," *Physica A Statistical Mechanics Its Applications*, vol. 490, Article ID S0378437117307227, pp. 222–230, 2017.

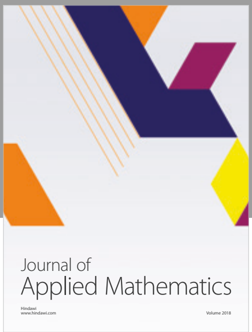
[42] Z. Zhou and Y. Yu, *Machine Learning and Its Application 2011*, Tsinghua University Press, Beijing, China, 2009.

[43] X. Wang, H. Xue, and W. Jia, *Prediction Model of Stock's Rosing and Felling Based on BP Neural Network*, Value Engineering, 2010.

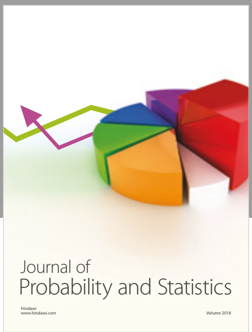
[44] K. B. Diether, K.-H. Lee, and I. M. Werner, "Short-sale strategies and return predictability," *Review of Financial Studies* , vol. 22, no. 2, pp. 575–607, 2009.

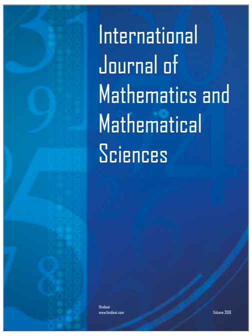
Advances in
Operations Research


Advances in
Decision Sciences

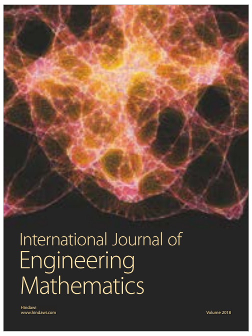

Journal of
Applied Mathematics

The Scientific
World Journal

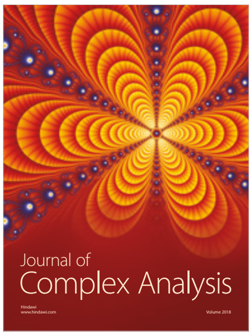

Journal of
Probability and Statistics

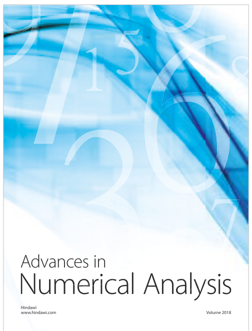

International
Journal of
Mathematics and
Mathematical
Sciences

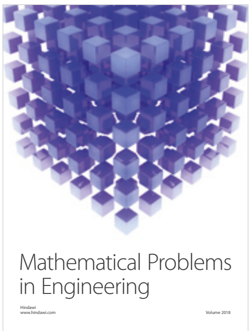
Journal of
Optimization

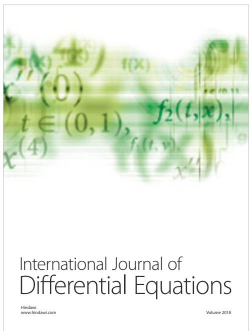

International Journal of
Engineering
Mathematics

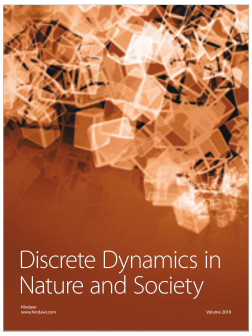
International Journal of
Analysis

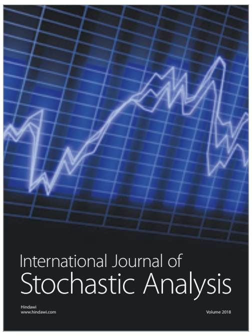

Journal of
Complex Analysis


Advances in
Numerical Analysis


Mathematical Problems
in Engineering

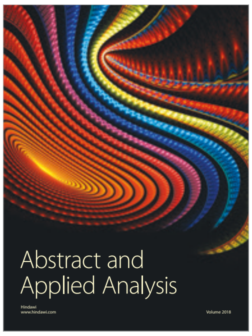

International Journal of
Differential Equations

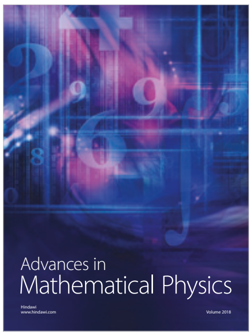

Discrete Dynamics in
Nature and Society


International Journal of
Stochastic Analysis

Journal of
Mathematics

Journal of
Function Spaces


Abstract and
Applied Analysis


Advances in
Mathematical Physics