information

MDPI

Article

A Hybrid Algorithm for Forecasting Financial Time Series Data Based on DBSCAN and SVR

Mengxing Huang [1,2], Qili Bao [1,2], Yu Zhang [1,2,*] and Wenlong Feng [1,2]

1 State Key Laboratory of Marine Resource Utilization in South China Sea, Hainan University, Haikou 570228, China; huangmx09@163.com (M.H.); baokeeley@163.com (Q.B.); fwlfwl@163.com (W.F.)
2 College of Information Science and Technology, Hainan University, Haikou 570228, China
* Correspondence: yuzhang_nwpu@163.com

Received: 17 December 2018; Accepted: 1 March 2019; Published: 7 March 2019

Abstract: Financial prediction is an important research field in financial data time series mining. There has always been a problem of clustering massive financial time series data. Conventional clustering algorithms are not practical for time series data because they are essentially designed for static data. This impracticality results in poor clustering accuracy in several financial forecasting models. In this paper, a new hybrid algorithm is proposed based on Optimization of Initial Points and Variable-Parameter Density-Based Spatial Clustering of Applications with Noise (OVDBCSAN) and support vector regression (SVR). At the initial point of optimization, ε and MinPts, which are global parameters in DBSCAN, mainly deal with datasets of different densities. According to different densities, appropriate parameters are selected for clustering through optimization. This algorithm can find a large number of similar classes and then establish regression prediction models. It was tested extensively using real-world time series datasets from Ping An Bank, the Shanghai Stock Exchange, and the Shenzhen Stock Exchange to evaluate accuracy. The evaluation showed that our approach has major potential in clustering massive financial time series data, therefore improving the accuracy of the prediction of stock prices and financial indexes.

Keywords: financial time series; parameter optimization; DBSCAN; SVR

1. Introduction

The analysis and forecast of financial time series are of primary importance in the economic world [1]. Compared to general data, financial data have their own particularity. There is a temporal correlation between data and data [2]. The financial time series are a dataset obtained from the selling price, limit up, and fluctuation of financial products in the financial field over time [3]. How to dig out valuable information in these massive data and find out rules to better guide scientific research has become a hot research topic. By studying the time series, the future trend of the index can be predicted [4]. In the capital market, where information is more and more open, it is an inevitable trend to disclose financial forecast information to a wider extent. Compared to historical financial information, financial forecasting has a strong correlation with decision-making services for investors, but as a kind of prior information, it is highly uncertain. If the reliability is low, it may mislead investors. In such a macro environment, many financial scholars are striving to explore financial forecasting methods [5].

Traditional data mining methods [6] have some limitations in data clustering and prediction. On one hand, the dimensions of time series are higher and higher, and the randomness is stronger and stronger [7]. On the other hand, traditional technology cannot achieve satisfactory results when dealing with noisy, random, and nonlinear financial time series. However, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm can solve this problem better. The DBSCAN algorithm

clusters different nodes from the perspective of a similarity measurement between nodes so that the different categories of nodes can be accurately evaluated [8]. In addition, support vector regression (SVR) is also combined to predict financial sequence research. The biggest feature of SVR is that it is proposed for the principle of structural risk minimization and has good generalization ability [9]. Therefore, this paper uses DBSCAN and SVR in the classification and regression of financial fields.

This paper combines existing clustering algorithms to mine, form a model of financial time series data, and predict relevant financial data. First, this article finds the limitations of the DBSCAN clustering algorithm in non-uniform density datasets and analyzes the feasibility of parameter adaptations of the DBSCAN algorithm. Second, the DBSCAN algorithm is improved by optimizing initial point and parameter adaption [10]. The article analyzes the influence of parameter dynamic change on the clustering effect and implements an improved algorithm combined with support vector regression [11]. Finally, the algorithm is applied to the prediction of financial time series data. The innovation points of this paper are as follows:

1. The parameters of the DBSCAN algorithm are sensitive and global, so it cannot effectively cluster datasets of different densities. This paper proposes an Optimization Initial Points and Variable-Parameter DBSCAN (OVDBSCAN) algorithm based on parameter adaption;
2. This paper combines the OVDBSCAN algorithm with SVR and proposes a new "hybridize OVDBSCAN with SVR" (HOS) algorithm. By establishing the regression prediction model, the regression prediction of the unsteady noise data is realized, and the prediction accuracy of stock price and the financial index is improved.

Section 1 mainly introduces the research background and structure of financial time series. Section 2 mainly introduces the current research status of the forecasting methods and models of financial time series at home and abroad. Section 3 proposes the parameter adaptive clustering algorithm based on financial time series data. In Section 4, the algorithm is used to conduct experiments and verify the improvement effect of the OVDBSCAN algorithm and predict the financial data. Section 5 is a summary of this paper.

2. Related Work

As early as the 1920s, the British mathematician Yule gave a regression model for predicting the law of market changes. Later, R. Agrawal [12] made a systematic elaboration on time series similarity. When the mathematician Robert F. Engel analyzed inflation in the financial market, he proposed the conditional variance model [13]. An American metrologist, G.E.P. Box, put forward the modeling theory, the analysis method of time series, and discussed the principle of the autoregressive integrated moving average model [14]. These methods are all classic methods of time series analysis. However, in the early days, most of these methods were used in a single variable and the same variance model. As financial markets have boomed and data have exploded, the barriers to early models have become more prominent and limited in dealing with large, complex, and noisy data. At the end of the last century, G. Das [15] proposed to intercept the data flow by sliding window, transforming the data obtained according to certain rules and then conducting clustering. Sheng-Hsun Hsu [16] proposed a self-organizing neural network and SVR to predict the stock price. Cheng-lung Huang [17] proposed a hybrid self-organizing feature map (SOFM)-SVR model for monetary expansion in financial markets. In the new century, artificial intelligence and mining technology have been improved and integrated with each other [18], which made data mining technology perfect. G. Peter compared the neural network model with the moving average (MA) model when predicting time series data and found that the neural network [19] had higher prediction accuracy when facing more complex nonlinear time series data.

The domestic research on the problem of financial time series not only expanded the application scope of relevant technologies and methods, but also improved and complemented relevant theories and put forward many solutions while introducing and learning the foreign models. For example,

in 1980, Professor Jiahao Tang and other professors proposed a classical model known as the nonlinear domain, namely the threshold autoregressive model [20]. Qihui Yao proposed a method to approximate the nonlinear regression function of high dimension, which was a new metric and estimation algorithm. This new method provided the possibility of finding more reasonable measurements and making an accurate calculation. Zhengfeng Xiong [21] expounded the characteristics of financial time series. He proposed several important features and proposed a new estimation method through the wavelet transform method. Chao Huang et al. [22] proposed a time series dimensional reduction model for the trend volatility of financial time series data. The time complexity of this model is linear and is not sensitive to noise interference. Bin Li proposed a frequent structural model for the time correlation of time series data and realized the discovery of multiple financial time series [23]. These results provided a more accurate test method for the stationarity of regression models.

With the progress of internet technology and the accumulation of data in multiple fields, data analysis methods and mathematical models have been constantly improved. Due to the difference in financial background and characteristics at home and abroad, researchers have proposed a variety of predicted methods and models for financial time series data. As can be seen from the above, there are many methods for forecasting. Time series analysis [24] is the analysis of a set of discrete data. In the case of the same time interval, each point in time corresponds to one datum. This is a trend prediction analysis method, which mainly analyzes past data to predict the trend of future data. However, time series analysis is mainly applied to statistical methods for dynamic data processing of electric power and power systems. If there is a big change in the outside world, it predicts based on data that have occurred in the past, and there is often a large deviation. As a general function approximator, a neural network [25] can approximate the modeling of arbitrary nonlinear objects with arbitrary precision. Although neural network technology has made great progress, there are still some difficult problems to solve, such as the difficulty in determining the number of hidden layer nodes in a neural network, the existence of learning phenomena, local minimum problems in the training process, etc.

In order to solve these problems, this paper proposes a hybrid algorithm for the forecasting of financial time series data based on DBSCAN and SVR to better predict financial data. Based on the OVDBSCAN algorithm for data clustering, SVR changes the principle of empirical risk minimization in traditional neural networks and proposes the principle of structural risk minimization, so it has good generalization ability.

3. Hybridizing OVDBSCAN with a Support Vector Regression Algorithm

This paper chooses a density-based parameter adaptive clustering algorithm and proposes an algorithm for hybridizing OVDBCSAN with SVR, referred to as an HOS algorithm. The algorithm is mainly designed for large data volumes, high signal-to-noise ratios, and inconspicuous features. First of all, OVDBSCAN is an improvement over the DBSCAN algorithm. It is an unsupervised algorithm, and the data can be classified without manual laborious marking. Second, the cluster after OVDBSCAN clustering can make the cluster into a highly cohesive state through a parameter setting, which is not only conducive to the proposed consistency of the cluster but also improves the antinoise. Finally, the nonlinear regression prediction is made with SVR. Therefore, the HOS algorithm is proposed to deal with the characteristics of a high signal-to-noise ratio, instability, and nonlinearity in financial time series. The overall structure of the algorithm is shown in Figure 1.

1. The width of the sliding window is fixed. The data of $n-1$ day are selected as the input data, and the data of the nth day as the output data, so that the data with a span of m years can be mined and studied. The format of the time series data intercepted by the sliding window is as

follows, where I is the input sequence, O is the output sequence, x is the $n-1$ input data on the nth day, and y is the output data on the nth day:

$$D = \begin{bmatrix} I_1O_1 \\ I_2O_2 \\ \vdots\vdots \\ I_nO_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1(n-1)}y_1 \\ x_{21} & x_{22} & \cdots & x_{2(n-1)}y_2 \\ \vdots & \vdots & \ddots\vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{n(n-1)}y_n \end{bmatrix}. \quad (1)$$

2. There are two basic domain parameters in the OVDBSCAN algorithm: ε represents the distance threshold of a certain domain, and MinPts represents the number of sample points in the domain with radius ε. First of all, select point p, find the distance of m points closest to point p, and calculate its average value. Then calculate the average distance of m nearest points from all points and store it in the distance_all of the structure. The average distance dataset of all points is clustered through DBSCAN to obtain cps clustering results, and the maximum value of the average distance point is obtained for each class i in cps. Then the distance between p and m closest to it is going to be ε_i of that kind of a point. The obtained ε_i is clustered from small to large, and the smallest X is selected. MinPts remains unchanged, and DBSCAN clustering is performed on the dataset. Choose the second smallest ε_i until all ε_i are used. After clustering, n independent clusters A and their center points will be obtained.
3. Model training of ε-SVR is carried out for n independent clusters A. Since the dataset is nonlinear, SVR introduces the kernel function to solve the nonlinear problem. The expression is

$$f(x) = w.\varphi(x) + b = \sum_{i=1}^{n} (a_i - a_i^*)K(x, x_i) + b. \quad (2)$$

The radial basis function is used to select appropriate parameters, namely the penalty coefficient C, insensitive loss coefficient ε, and width coefficient γ, to train the cluster A model. The expression is as follows:

$$k(x, y) = \exp(-\gamma||x - y||_2). \quad (3)$$

4. Cluster A_i, trained by model ε-SVR, is optimized for particle swarm optimization (PSO) parameters: ε can be set by past experience. Mean square error is obtained by k-fold cross validation:

$$MSE = \frac{\sum_{i=1}^{n}(x_i - \hat{x_i})}{n}, \quad (4)$$

where $\hat{x_i}$ is the predicted value of x_i.
5. After optimizing the parameters, the cluster A_i that was optimized by the parameters is trained, and the model M_i is obtained.
6. The test data are matched with n cluster centers after clustering using the DBSCAN algorithm to find the most similar W cluster center and the model M corresponding to the W cluster.
7. The predicted value of test data is calculated with the SVR of the corresponding model M. Complete dataset mining and regression analysis.

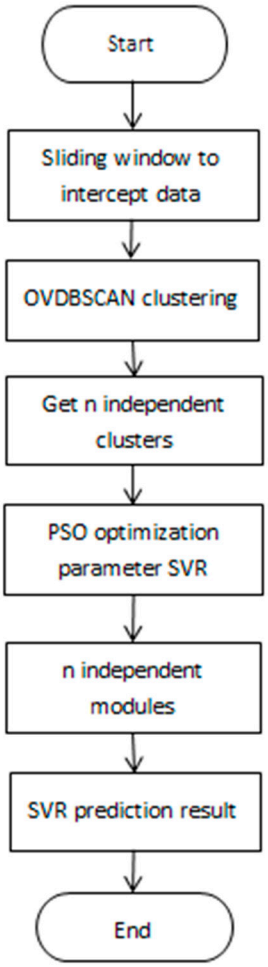

Figure 1. "Hybridize Optimization Initial Points and Variable-Parameter Density-Based Spatial Clustering of Applications with Noise (OVDBSCAN) with support vector regression (SVR)" (HOS) structural process.

According to the subsequent experimental simulation and evaluation verification, compared to the traditional single methods (such as SVR's and SOFM's prediction of financial time series), HOS, the hybrid algorithm of OVDBSCAN and SVR proposed in this paper, can find a large number of similar classes and then establish a regression prediction model, which improves the accuracy of the prediction of stock prices and financial indexes.

3.1. DBSCAN Parameter Optimization

In the clustering algorithm, since *k*-means is sensitive to the initial point, it is often applied to the optimization of its initial point. Similarly, with DBSCAN [26,27], a clustering algorithm, people do not pay as much attention to the optimization of its initial point as *k*-means. However, through the research in this paper, it was found that the optimization of the initial point of the DBSCAN algorithm can improve the clustering effect to a certain extent. Inspired by some researchers' initial point optimization of the *k*-means algorithm, this paper proposes an OVDBSCAN algorithm of initial point optimization and variable parameters to adapt to a density dataset of any shape and size changes.

The steps for initial point optimization are as follows:

1. Find the distance of m points closest to P and average them. Find the average distance of m closest points from all points;
2. The average distance dataset of m nearest points of all points is clustered through DBSCAN, and cps clustering results are obtained;

3. Find the maximum value of the average distance point for each class i in cps;
4. The distance between point P and the closest point m to it is the distance of this point.

The above method of finding ε at the initial point of optimization mainly deals with datasets of different densities. According to different densities, appropriate parameters are selected for clustering through optimization.

After the initial point optimization, DBSCAN clustering of variable parameters can be further carried out. The steps are as follows:

1. The ε_i obtained in the above paper is sorted from small to large to start clustering;
2. Select the smallest ε_i. MinPts remains unchanged. DBSCAN clustering is performed on the dataset;
3. Select the second smallest ε_i. MinPts remains unchanged. Cluster the data marked as noise;
4. Cycle through the above operations until all ε_i are used up and the clustering ends.

In the process of cyclic clustering, the value of ε_i is carried out from small to large. When ε_i is small, because the dataset is far away from ε_i and not clustered, the smaller value of ε_i can only cluster to the high-density point, but has no influence on the low-density data.

In the case of uneven data distribution, the DBSCAN algorithm may cluster high-density clusters into low-density clusters due to parameters, or may process elements in low-density clusters as noise. The parameter-adaptive OVDBSCAN algorithm can find clusters of any shape, and can effectively realize clustering of datasets with large density differences. The specific results are shown in the experiment in the fourth part of this paper.

3.2. PSO Optimization Parameters

A particle swarm optimization algorithm is an evolutionary computing technology whose idea is to find the optimal solution based on the collaboration between individuals in the group and the sharing of information [28].

3.2.1. Particle Swarm Optimization Principle

For each problem, the best solution is a bird in the search space, namely a "particle", and the optimal solution is the "corn field" that the birds are looking for [29]. Each particle has a position vector and a velocity vector, and the adaptive value of the current position can be calculated according to the objective function, which can be understood as the distance from the "corn field". The PSO is initialized as a group of random particles, and then the optimal solution is found by iteration. During the iteration, it completes self-renewal by tracking the "extreme value" (Pbest, Gbest). After tracing to the optimal value, the particle updates its speed and location based on the following two formulas:

$$v_i = v_i + c_1 \times \text{rand}() \times (\text{pbest}_i - x_i) + c_2 \times \text{rand}() \times (\text{gbest}_i - x_i), \quad (5)$$

$$x_i = x_i + v_i, \quad (6)$$

where x_i is the position of the current particle, V_i is the velocity of the particle, and c_1 and c_2 are learning factors, usually with a value of 2. Rand () is a random number with a value between (0, 1).

3.2.2. PSO to Optimize $\varepsilon-$SVR

It can be seen from the above that $\varepsilon-$SVR has three parameters: ε can be set by previous experience. Therefore, the optimization of $\varepsilon-$SVR by PSO only needs to optimize c and γ. The fitness value is obtained by k-fold cross-validation to obtain the mean square error.

The optimization steps of the parameters are as follows: A set of (c, γ) is randomly generated as the initial value.

1. Divide the training data into k parts: s_1, s_2, ..., s_k;
2. Use the current (c,γ) for $\varepsilon-$SVR training, and obtain mean square error (MSE) by cross-validation;
3. Initialize i, and let it equal 1;
4. Use s_i as the test dataset and the other as the training set;
5. Calculate the MSE of the ith subset. Perform $i = i + 1$. Return to the fifth step until $i = k + 1$;
6. Calculate the average value of k times of MSE;
7. Select the average MSE value after k-fold cross-validation, and record the Pbest and Gbest of individuals and groups. Continue to find better (c, γ); Repeat steps 2 through 8 until the set number of times is satisfied;
8. End.

In summary, the main parameter of $\varepsilon-$SVR is (ε, γ, c), where ε can be directly set to a certain constant, and γ and c can be obtained by PSO optimization parameters.

3.3. The Thought of the HOS Algorithm

The HOS algorithm not only has the clustering ability of OVDBSCAN, which can extract features, but also has the regression prediction ability of SVR. This allows data with similarities to be clustered, while noisy and nonstationary data are suppressed well. The idea of the HOS algorithm is that after the training data are clustered, they are divided into n clusters. Then each cluster is given to the SVR for training, and each cluster is trained to corresponding feature models to obtain n training. The model is then handed over to HOS for testing. The test data are calculated by OVDBSCAN to calculate the distance from the data point to the core point and then get the cluster with the shortest distance. The test data points are based on the model corresponding to the cluster for SVR regression prediction. The thought of HOS algorithm is shown in Figure 2.

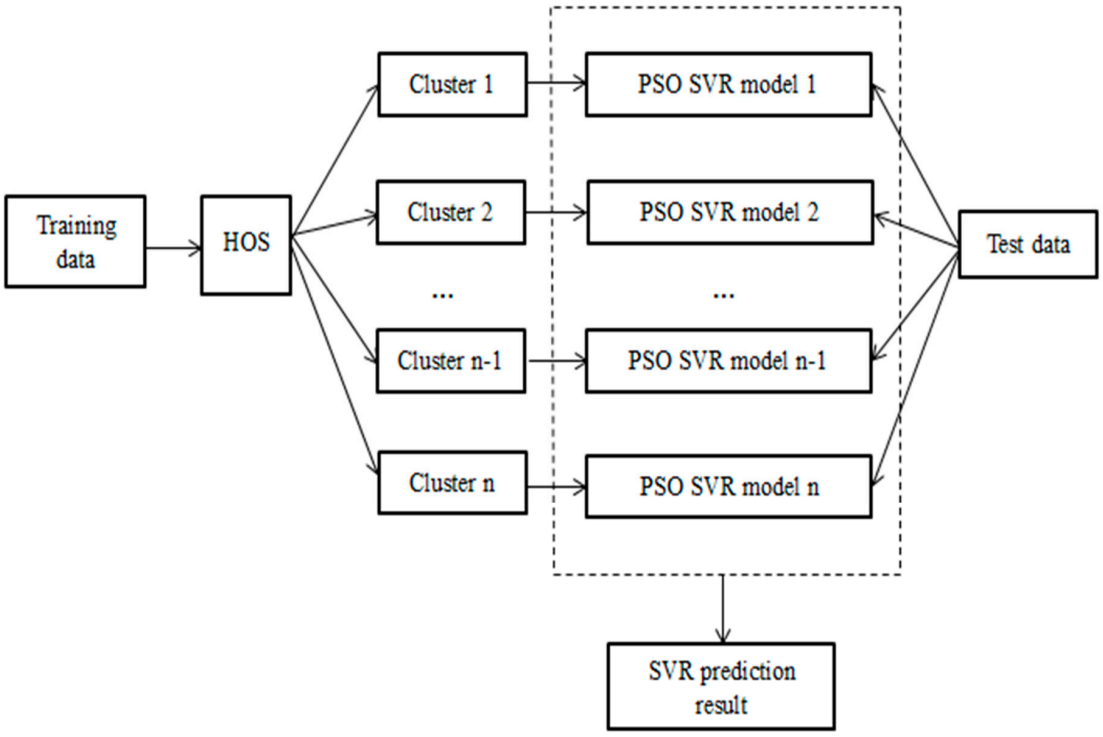

Figure 2. The thought of HOS.

The hybrid algorithm proposed in this chapter clusters the data into multiple clusters, extracts the characteristics of each cluster, and makes the matching of these different features and test data

to obtain the predicted result. According to the experimental needs, the system mainly has three functional modules:

1. System interface module;
2. Experimental introduction module. Explain the function of each module briefly;
3. Experimental module for SVR model design. Implement various functions of SVR (dataset loading, parameter settings, regression demonstration).

The flow chart of the HOS algorithm is shown in Figure 3.

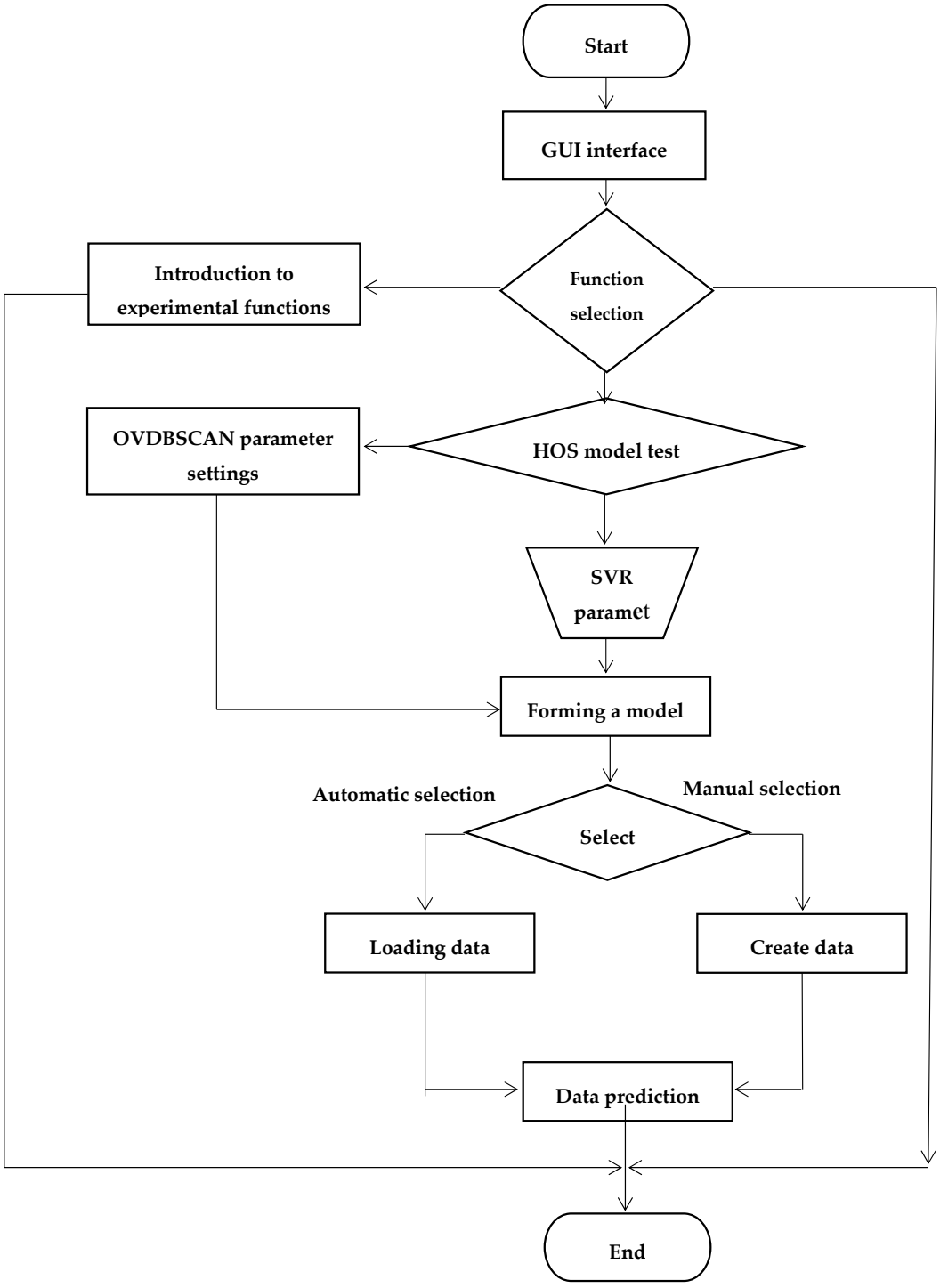

Figure 3. HOS algorithm flow chart.

4. Research on Financial Time Series Predictions Based on HOS Algorithm

This experiment was conducted in Windows (CPU 2.10 GZ, memory 6 G). The operating system was Windows10. The test tool was MATLAB 2012. There were three experiments in this part. The first experiment was based on the algorithm optimization of the DBSCAN algorithm. It proved that the parameter adaptive algorithm OVDBSCAN had a better clustering effect on the data. The second experiment was based on the HOS algorithm for financial index prediction. It was found that the HOS algorithm was closer to the real value and had a stronger prediction ability. The third experiment was based on the HOS algorithm to predict the daily limit of a stock, which proved that the prediction result of the HOS algorithm was more accurate and can better guide people's business behavior.

4.1. Optimization of Clustering Algorithm Based on DBSCAN Algorithm

This part proposed an OVDBSCAN algorithm for initial point optimization and variable parameters. It was verified through experiments that the OVDBSCAN algorithm could adapt to a density dataset of arbitrary shape and size.

4.1.1. Experimental Design

Three two-dimensional datasets were used for experiments. Dataset 1 was two different density clusters with noise data. Dataset 2 was three different clusters that all contained noise data. The first cluster had few data and had a higher density. The second cluster had a large amount of data and had a higher density. The third cluster had a small amount of data and had a low density. Dataset 3 had four clusters that contained noise but different densities.

Three dataset samples were clustered according to the OVDBSCAN algorithm. Inspired by [30], we found the average distance dis_means (7) of $m = 7$ points closest to point P. We found the average distance dis_allmeans () of all points. We performed DBSCAN clustering on dis_allmeans, where $\varepsilon = 8$, MinPts = 4. After the average distance through clustering, the maximum of 6 distances of the average distance point in each class was selected as the highest value of crane. The ε values of the three datasets were

1. $\varepsilon_1 = 23.012816$, $\varepsilon_2 = 78.175813$;
2. $\varepsilon_1 = 27.56658$, $\varepsilon_2 = 80.039573$, $\varepsilon_3 = 80.039673$;
3. $\varepsilon_1 = 40.1995502$, $\varepsilon_2 = 83.743656$;

In the above clustering process, MinPts was set to 4, and the effect of clustering on the three datasets is shown in Figures 4–6.

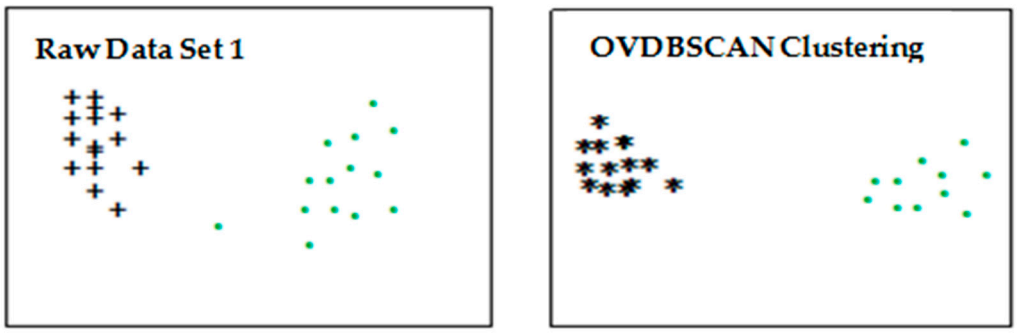

Figure 4. Dataset 1 cluster process.

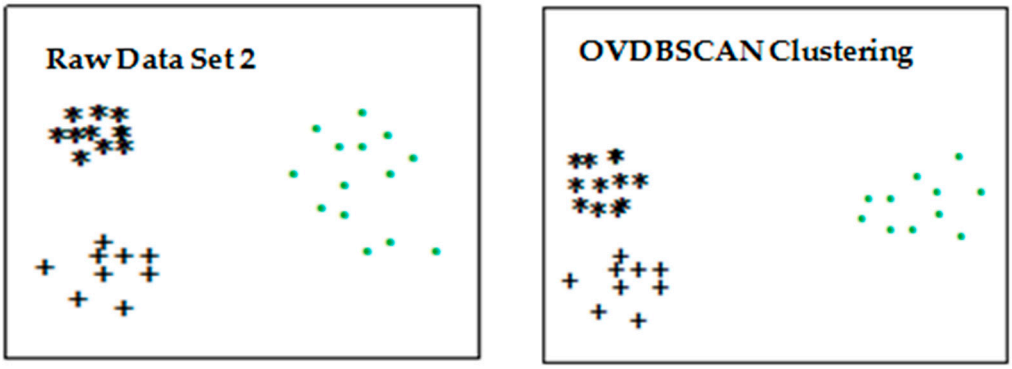

Figure 5. Dataset 2 cluster process.

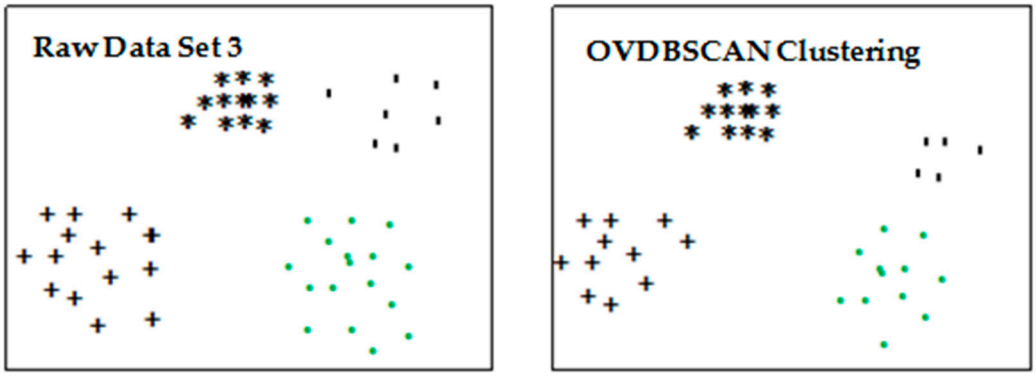

Figure 6. Dataset 3 cluster process.

The clustering results in the figure above use different colors and shapes to distinguish different clusters. The results show that the improved DBSCAN algorithm implemented effective clustering for clusters with different densities.

In order to further illustrate the effect of the OVDBSCAN clustering algorithm, we used the data of dataset 1 to observe the influence on the clustering results by setting different ε values. See Figure 7.

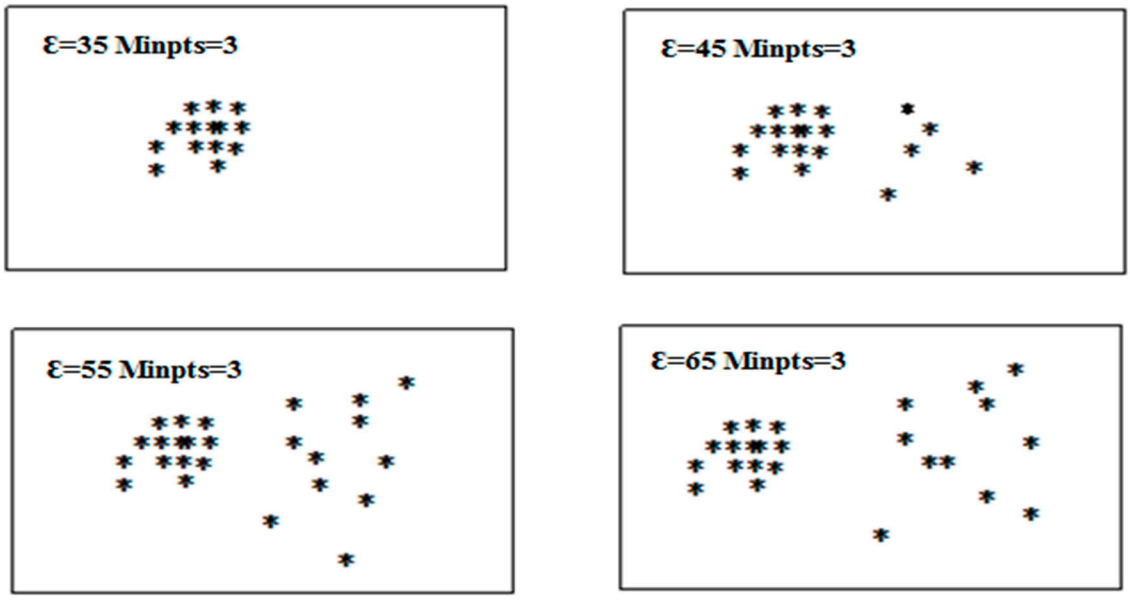

Figure 7. The effect of change ε.

4.1.2. Experimental Results

It was known from the experimental results that the parameters had a great influence on the DBSCAN algorithm. When the parameters are not set properly, the clustering results have a certain deviation. In the case of uneven data distribution, the DBSCAN algorithm may cluster high-density clusters into low-density clusters due to parameters, or may process elements in low-density clusters as noise. The OVDBSCAN algorithm proposed in this paper can effectively achieve clustering of datasets with large density differences and can discover clusters of arbitrary shape. The idea is that different cluster densities are clustered using different ε based on the form of variable parameters. The experimental results showed that the OVDBSCAN algorithm was effective.

4.2. Financial Index Prediction Based on HOS Algorithm

4.2.1. Experimental Design

In order to verify the prediction effect of the HOS algorithm on a financial index, it was compared to the SVR algorithm and the SOFM-SVR algorithm prediction. The data obtained in this paper were data from Ping An Bank from 4 January 2013 to 31 December 2014, as experimental data. The experimental data are shown in Figure 8.

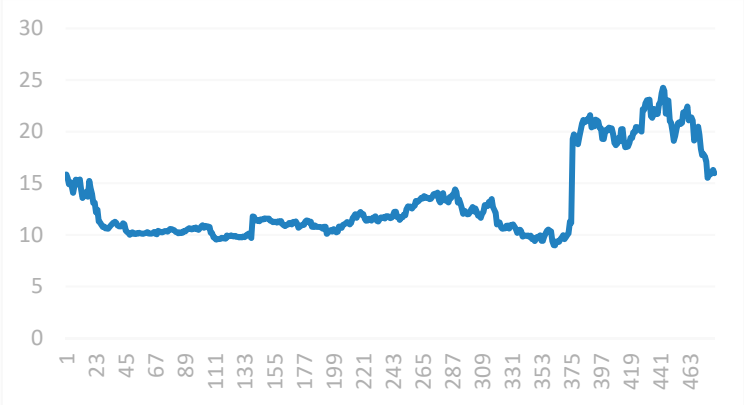

Figure 8. Weighted index trend chart.

The data of these two years were divided into four groups to make the experimental results more concise and more convincing. The first 80% of each group of data were used as experimental data, and the last 20% were used as test data. The specific grouping situation is shown in Table 1.

Table 1. Data grouping.

Packet Number	Training Data	Test Data
1	2013.1.4–2014.4.4	2014.4.5–2014.9.4
2	2013.2.4–2014.5.4	2014.5.5–2014.10.4
3	2013.3.4–2014.6.4	2014.6.5–2014.11.4
4	2013.4.4–2014.7.4	2013.7.5–2014.12.31

The financial data format downloaded from the above website is shown in Figure 9 below.

	A	B	C	D	E	F
1	code	date	open	high	low	close
2	sz000001	2014/12/31	15.61	15.88	15.3	15.84
3	sz000001	2014/12/30	14.93	15.59	14.87	15.5
4	sz000001	2014/12/29	15.61	16.05	14.75	14.92
5	sz000001	2014/12/26	14.61	15.18	14.52	15.1
6	sz000001	2014/12/25	14.35	14.75	14.04	14.69
7	sz000001	2014/12/24	14.74	14.85	13.68	14.07
8	sz000001	2014/12/23	14.99	15.28	14.5	14.75

Figure 9. Financial data format.

The (code, date, opening price, highest price, lowest price, closing price) will be described as $S_i = (code_i, date_i, open_i, low_i, close_i)$ In this paper, the data stream was processed by a sliding window with a window width of 4. The converted format was

$$\begin{aligned} \mathrm{S_i} = (\mathrm{data_i}(\frac{\mathrm{close_{i-1}}}{\mathrm{close_{i-2}}} - 1) \times 100, (\frac{\mathrm{open_i}}{\mathrm{close_{i-2}}} - 1) \times 100, (\frac{\mathrm{high_i}}{\mathrm{close_{i-2}}} - 1) \times 100, \\ (\frac{\mathrm{low_i}}{\mathrm{close_{i-2}}} - 1) \times 100, (\frac{\mathrm{close_i}}{\mathrm{close_{i-2}}} - 1) \times 100, (\frac{\mathrm{close_{i+1}}}{\mathrm{close_{i-2}}} - 1) \times 100) \end{aligned} \tag{7}$$

As the predicted value, $\left(\frac{\mathrm{close_{i+1}}}{\mathrm{close_{i+2}}} - 1\right) \times 100$ was the closing price of the next day. We represented the processed data as the volatility of the data. Since the range of fluctuations was the ratio of the previous day's difference, the range of values was not large. Normalization of the original data was also unnecessary. This allowed for showing more internal connections between the data. Although the next day's data trend was not directly linearly determined by the data of the day, it could be found through an analysis of the stock market in the past that there were still many higher repetition rates of the trend of the situation. Second, the trading situation of the day (such as the opening price and the highest price) is an important reference for investors, and directly affects investors' investment behaviors (such as buying and selling). Therefore, it was very valuable to dig out the data of the day and analyze the market situation. The selection of financial index parameters by HOS is shown in Table 2 below.

Table 2. The selection of parameters by HOS. PSO: Particle swarm optimization.

HOS	Numerical Value	SVR	Numerical Value	PSO	Numerical Value
k	25	Maximum value of c	100	Local search capability	1.5
cps	2000	Maximum value of c	0.01	Global search capability	1.7
		Maximum value of γ	1000	Maximum evolutionary quantity	200
		Maximum value of γ	0.01	Maximum population value	20
		ε	0.01	Cross-validation k	5

In order to test the predictive performance of the HOS algorithm, we used the mean absolute error (MAE), the mean square error (MSE), and the mean absolute percentage error (MAPE) [31,32].

The expression of MAE is

$$\mathrm{MAE} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} |\mathrm{A_i} - \mathrm{F_i}|}{\mathrm{n}}. \tag{8}$$

The expression of MSE is

$$\mathrm{MSE} = \frac{\sum_{i=1}^{n} (A_i - F_i)^2}{n}. \quad (9)$$

The expression of MAPE is

$$\mathrm{MAPE} = \frac{\sum_{i=1}^{n} |A_i - F_i|}{n} \times 100. \quad (10)$$

A_i represents the true value of each data. F_i is the predicted value of the point, and n is the total number of data points. As with MSE and MAPE, if the value of MAE is smaller, the prediction accuracy is greater.

Using the above three indicators, the performance of the HOS algorithm on the four datasets of Ping An Bank through experiments is shown in Table 3 below.

Table 3. Evaluation of algorithm index of HOS. MSE: Mean square error; MAE: Mean absolute error; MAPE: Mean absolute percentage error.

Dataset	MSE	MAE	MAPE
1	4567	53.62	0.7318
2	5926	59.38	0.9637
3	4857	58.16	0.9368
4	3598	48.71	0.6862

The performance of data evaluations using SVR and SOFM-SVR algorithms alone had given clear answers in the literature [33], and their performances are shown in Tables 4 and 5.

Table 4. Evaluation of algorithm index of SVR.

Dataset	MSE	MAE	MAPE
1	27,853	163.12	2.7518
2	365,819	429.68	6.3294
3	13,829	265.42	3.5291
4	21,964	103.55	2.1372

Table 5. Evaluation of algorithm index of SOFM-SVR.

Dataset	MSE	MAE	MAPE
1	26,197	153.28	2.2275
2	48,392	162.53	2.5296
3	8375	68.97	1.3625
4	19,374	77.82	1.6482

4.2.2. Experimental Results

Through a comparison of the above three methods, the performances of the HOS algorithm, SVR algorithm, and SOFM-SVR algorithm on MAPE, MAE, and MSE were found. The HOS algorithm performed best. The HOS algorithm predicted that the algorithm was closer to the real value than the latter two algorithms and had stronger prediction ability.

4.3. Prediction of the Next-Day Trading Limit of a Stock Based on the HOS Algorithm

4.3.1. Experimental Design

In the previous paper, through the experimental verification of the proposed hybrid algorithm, it was found that the HOS algorithm had a good prediction effect on the prediction of the financial

index. In this section, the verified algorithm was used to analyze the trading price of a stock on that day and predict the next day's closing price. This article did not output all the predicted results, only those that exceeded the rise and the code number of stocks that continued to rise.

The data obtained in this paper were 1000 data listed on the Shanghai Stock Exchange and 1000 data on the Shenzhen Stock Exchange. The total number of time series was 2000 × 485, which was 970,000. The time span was from 4 January 2013 to 31 December 2014. In order to make the experimental results have a clear contrast and to make the prediction more convincing, the experimental data were divided into two parts, as shown in Table 6.

Table 6. Data grouping.

Dataset Number	Training Data	Test Data
1	2013.01.04–2014.8.31	2014.9.01–2014.11.31
2	2013.01.04–2014.9.30	2014.10.01–1014.12.31

The original format of the data is shown in Figure 10.

	A	B	C	D	E	F
1	code	date	open	high	low	close
2	sz000068	2014/12/31	8.76	8.93	8.5	8.74
3	sz000068	2014/12/30	8.87	8.98	8.71	8.75
4	sz000068	2014/12/29	9.25	9.26	8.85	8.86
5	sz000068	2014/12/26	9.19	9.38	9.05	9.21
6	sz000068	2014/12/25	9.08	9.46	9.08	9.2
7	sz000068	2014/12/24	8.7	9.03	8.62	9.02
8	sz000068	2014/12/23	8.81	9.03	8.6	8.62

Figure 10. Original data format.

That is, $S_i = (code_i, date_i, open_i, low_i, close_i)$, and the processing format is as follows:

$$\begin{aligned} S_i = (&code_i data_i (\frac{mean(i-9,i-5)}{mean(i-10,i-6)} - 1) \times 100, (\frac{mean(i-8,i-4)}{mean(i-10,i-6)} - 1) \times 100, (\frac{high_i}{close_{i-2}} - 1) \times 100, \\ &(\frac{mean(i-7,i-3)}{mean(i-10,i-6)} - 1) \times 100, (\frac{mean(i-6,i-2)}{mean(i-10,i-6)} - 1) \times 100, \\ &(\frac{mean(i-5,i-1)}{mean(i-10,i-6)} - 1) \times 100, (\frac{close_{i-1}}{close_{i-2}} - 1) \times 100, \\ &(\frac{close_i}{close_{i-1}} - 1) \times 100, (\frac{close_{i+1}}{close_i} - 1) \times 100) \\ mean(n-4, n) = &\frac{close_{n-4} + close_{n-3} + close_{n-2} + close_{n-1} + close_n}{5}. \end{aligned} \quad (11)$$

The mean $(n-4, n)$ represents the average of the closing prices (close) for the five days from the $n-4$ to the nth day. Thus, the original data format is transformed into code=(date, change(i − 5, i − 6), change(i − 4, i − 6), change(i − 3, i − 6), change(i − 2, i − 6), change(i − 1, i − 2) close(i − 1, i − 2), close(i, i − 1), close(i + 1, i), close(i + 1, i)) after processing the original data in the following manner. Change (i − 5, i − 6) is the comparison between the increase of the i − 5 daily average and the increase of the i − 6 daily average, and close(i − 1, i − 2) is the comparison between the increase of the closing price on the i − 1 day and the increase of the closing price on the i − 2 day.

The purpose of doing this with raw data was to better reflect trends in share prices and recent days of trading in huge numbers of stocks. Investors tend to be concerned about a stock's trading in the last few days and make a judgment of the day's investing behavior based on recent trading trends. From the final experimental results, it could be found that such data processing was effective and meaningful.

The parameter settings for stock forecasting are shown in Table 2 above. All next-day gains are shown in Figures 11–13 (regardless of whether the day was up or down): All the first days of the days,

the daily increase of the daily limit (the daily limit of the day when the daily limit was not updated would record the data of the next day's gain), and the next day's increase in the HOS forecast (that is, after the intervention of the HOS algorithm, all the output prompts were recorded as the next day's increase data).

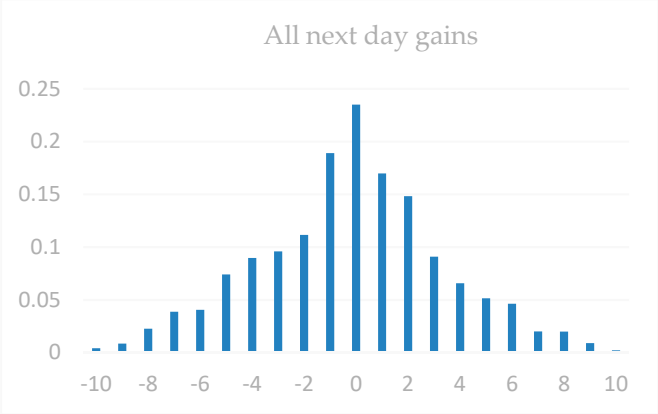

Figure 11. Comparison of the next day's rise in the days of dataset 1.

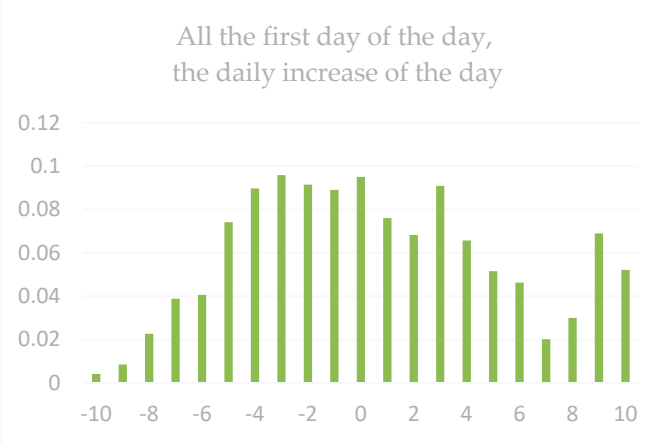

Figure 12. Comparison of the day's first rise in the days of dataset 1.

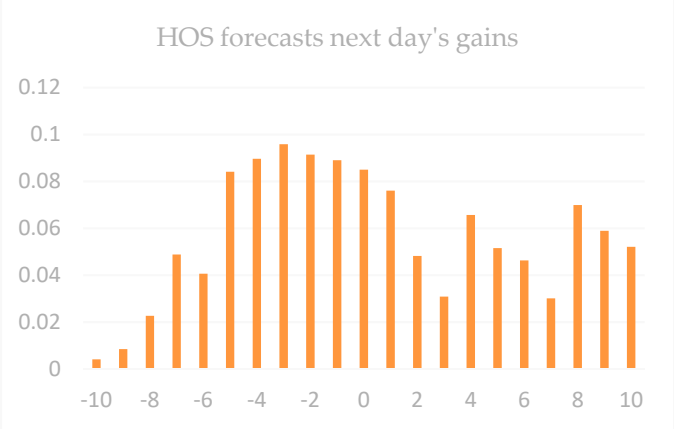

Figure 13. Comparison of the day's rise in the days of dataset 1.

It can be seen from the figure that in all the next day's gains, the gain density around 0 was higher. Therefore, it was really difficult for average investors to observe the massive financial data and draw

corresponding forecasts. Because of the whole process of the increase, the market showed a relatively stable state. Therefore, only by scientifically and effectively researching these data could we discover the laws. It could be found from the increase in the first daily limit of the first day of the day that it was still the highest concentration of the increase in 0 and spread to both ends, especially for the next day. Because the daily limit of the day is an important reference for investors, this directly affects the investor's investment trend the next day.

In order to have a digital comparison of the chart, the next day's increases in the three cases of dataset 1 are shown in Table 7 below.

Table 7. The contrast of the next day.

Dataset 1	Gain < −9	Gain < −5	Gain < −3	Gain > 0	Gain > 3	Gain > 5	Gain > 9
All next-day gains	0.57	4.62	11.86	48.37	9.06	4.35	1.3
All the first daily limits	2.08	7.32	13.57	62.52	35.95	22.73	12.66
HOS	1.03	10.0	17.92	60.39	37.62	23.83	16.18

From the above table, we can see that the increase was higher than 3 and had a higher ratio than the first daily limit. On one hand, dataset 1 was caused by low market volume. On the other hand, this algorithm needs to be improved in data applications.

As shown in Figures 14–16, it can be found from the morphological comparison chart of dataset 2 that the HOS algorithm proposed in this paper had its advantages in predicting the next-day trading limit of stocks. From its distribution graph, it could be found that it was not concentrated near a certain point, and the proportion of increases below −5 was negligible. The overall distribution moved up significantly. For a better data comparison, see Table 8.

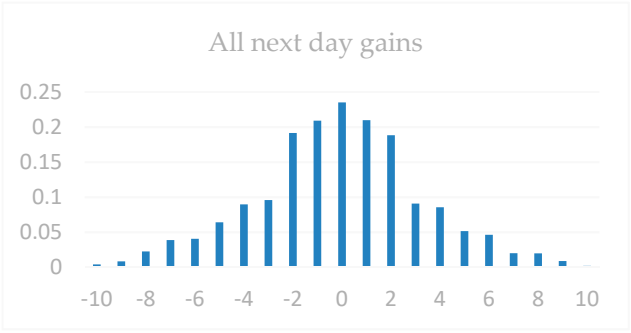

Figure 14. Comparison of the next day's rise in the days of dataset 2.

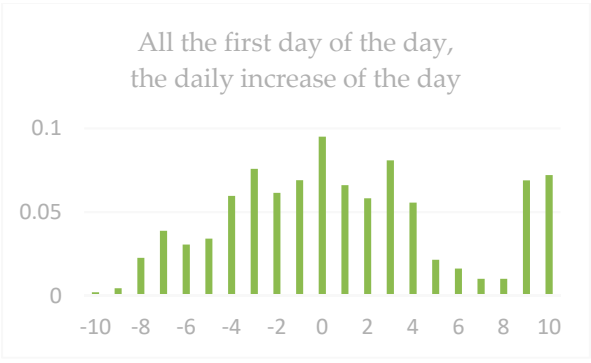

Figure 15. Comparison of the day's "the first rise in the day" of dataset 2.

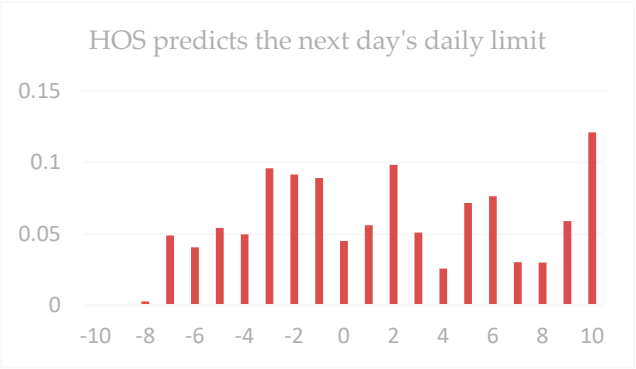

Figure 16. Comparison of the day's rise in the days of dataset 2.

Table 8. The contrast of the next day.

Dataset 2	Gain < −9	Gain < −5	Gain < −3	Gain > 0	Gain > 3	Gain > 5	Gain > 9
All next-day gains	0.38	1.69	6.97	57.25	15.03	5.75	1.58
All the first daily limits	0.26	2.21	11.06	66.82	39.97	25.41	13.05
HOS	0	1.58	7.89	71.02	46.35	38.07	17.01

Tables 7 and 8 show the proportion of the next day's increase in different datasets. It could be found that the ratio of the next day's increase above 0 was around 62%, while the dataset in dataset 2 was greater than the dataset 1. The advantages were even more obvious.

4.3.2. Experimental Results

The analysis of the two datasets through the above experiments showed that the HOS algorithm had obvious advantages over other prediction methods in the prediction results of the next day's increase in a stock's daily limit.

4.4. Stock Price Predicting Based on HOS Algorithm

4.4.1. Experimental Design

For stock price predicting, the data used in this paper came from the listed stock data of Minsheng Bank (stock code sh600016) and China Unicom (stock code sh600050) of the Shanghai Stock Exchange. The interception period was from 4 January 2013 to 31 December 2014. The first 80% of the data were used as the training set, and the rest were used as the test set. The data were divided as shown in Table 9.

Table 9. Data time division.

Data Source	Training Data	Test Data
Minsheng Bank	2013.1.4–2014.8.15	2014.8.16–2014.12.31
China Unicom	2013.1.4–2014.8.15	2014.8.16–2014.12.31

The format of the stock data downloaded from the website was (stock code, trading time, opening price, maximum amount, minimum amount, closing price, price change), and its format is as shown in Figure 17.

	A	B	C	D	E	F	G
1	code	date	open	high	low	close	change
2	sh600016	2014/12/31	10.71	10.95	10.46	10.88	0.009276
3	sh600016	2014/12/30	10.77	10.98	10.52	10.78	0.00466
4	sh600016	2014/12/29	11.1	11.54	10.49	10.73	-0.0156
5	sh600016	2014/12/26	10.84	11.05	10.5	10.9	0.014898
6	sh600016	2014/12/25	10.7	11	10.35	10.74	0.042718
7	sh600016	2014/12/24	10.3	10.77	9.98	10.3	-0.01905
8	sh600016	2014/12/23	10.15	11.1	9.94	10.5	-0.00095
9	sh600016	2014/12/22	9.49	10.58	9.43	10.51	0.091381

Figure 17. Stock data format.

The table can be described as follows. We normalized the data to be transformed between [0, 1]. Here we used the linear normalization method, and the conversion method is as follows:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \tag{12}$$

X is the raw data. X_{max} represents the maximum value in the dataset. X_{min} is the minimum value. The parameter settings are the same as in Table 2. In order to test the predictive performance of the HOS algorithm, we used MSE and MAPE to evaluate the evaluation index. Through the regression analysis and prediction of the stock prices of the two companies, it could be found that the HOS algorithm had a small prediction error in the data. In order to see the effect more intuitively, the actual value and predicted value of the test set data are shown here in Figures 18 and 19.

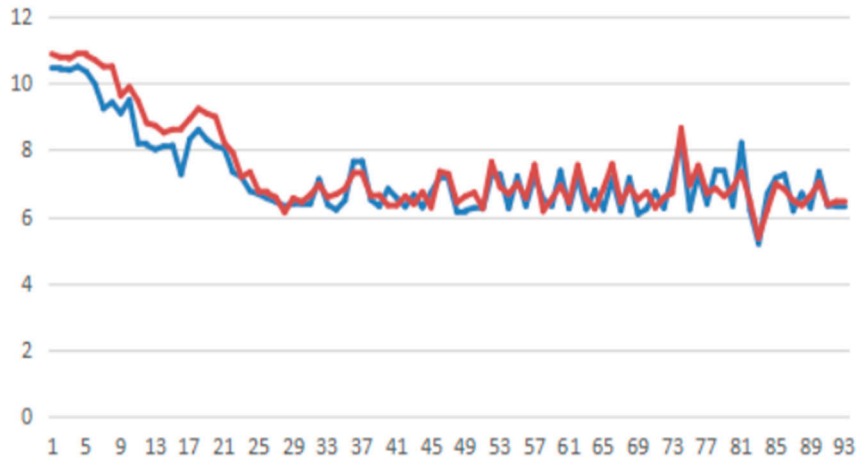

Figure 18. Forecasting value of the test set of Minsheng Bank (actual value predicted value).

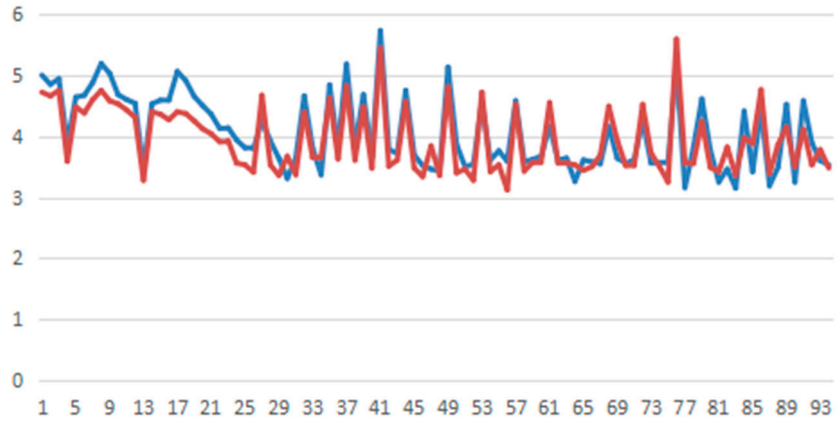

Figure 19. Forecasting value of the test set of Chine Unicom (actual value predicted value).

4.4.2. Experimental Results

It could be found that the HOS algorithm forecasted the stock price more accurately. This has a great reference value for investors, and the HOS algorithm can better guide people's economic behavior.

5. Conclusions

In this paper, we proposed a new hybrid algorithm for the forecasting of financial time series based on DBSCAN and SVR. OVDBSCAN optimizes the global invariability of parameters and realizes parameter adaptation. This provides direction for good clustering of datasets of different densities. HOS is able to establish regression prediction models by finding a large number of similar classes, which improves the accuracy of the financial index and stock next-day trading limit predictions. The experimental results confirmed the effectiveness and robustness of the proposed algorithm. However, the running time of the HOS algorithm compared to traditional SVR and SOFM algorithms is much larger than the latter two algorithms. Therefore, our future direction is to improve the ability of processing the data. We will consider not loading all data into the model, but rather selecting the data in the form of random sampling. Judging from the evaluation indexes, the accuracy of the HOS algorithm prediction still has a lot of room for improvement.

Author Contributions: M.H. and Q.B. conceived and designed the algorithm. W.F. analyzed the dataset. Q.B. and Y.Z. designed, performed, and analyzed the experiments. All authors read and approved the final manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant #: 61462022 and Grant #61662019), the Major Science and Technology Project of Hainan province (Grant #: ZDKJ2016015), the Natural Science Foundation of Hainan province (Grant #:617062), and the Higher Education Reform Key Project of Hainan province (Hnjg2017ZD-1).

Acknowledgments: The authors would like to express their great gratitude to the anonymous reviews that helped to improve the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dablemont, S.; Verleysen, M.; Van Bellegem, S. Modelling and Forecasting financial time series of "tick data". *Forecast. Financ. Mark.* **2007**, *5*, 64–105.
2. Washio, T.; Shinnou, Y.; Yada, K.; Motoda, H.; Okada, T. Analysis on a Relation Between Enterprise Profit and Financial State by Using Data Mining Techniques. In *New Frontiers in Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2006.
3. Yan, L.J. The present situation and future development trend of financial supervision in China. *Cina Mark.* **2010**, *35*, 44.
4. Liao, T.W. Clustering of time series data—A survey. *Pattern Recogn.* **2005**, *38*, 1857–1874. [CrossRef]

5. Rubin, G.D.; Patel, B.N. Financial Forecasting and Stochastic Modeling: Predicting the Impact of Business Decisions. *Radiology* **2017**, *283*, 342. [CrossRef] [PubMed]
6. Qadri, M.; Ahmad, Z.; Ibrahim, J. Potential Use of Data Mining Techniques in Information Technology Consulting Operations. *Int. J. Sci. Res. Publ.* **2015**, *5*, 1–4.
7. Xu, Y.; Ji, G.; Zhang, S. Research and application of chaotic time series prediction based on Empirical Mode Decomposition. In Proceedings of the IEEE Fifth International Conference on Advanced Computational Intelligence, Nanjing, China, 18–20 October 2012.
8. Ertöz, L.; Steinbach, M.; Kumar, V. *Fiding Clusters of Different Sizes, Shapes, and Densities in Noise, High Dimensional Data*; SIAM: Philadelphia, PA, USA, 2003.
9. Li, L.H.; Tian, X.; Yang, H.D. Financial time series prediction based on SVR. *Comput. Eng. Appl.* **2005**, *41*, 221–224.
10. Fu, Z.Q.; Wang, X.F. The DBSCAN algorithm based on variable parameters. *Netw. Secur. Technol. Appl.* **2018**, *8*, 34–36.
11. Li, L.; Xu, S.; An, X.; Zhang, L.D. A Novel Approach to NIR Spectral Quantitative Analysis: Semi-Supervised Least-Squares Support Vector Regression Machine. *Spectrosc. Spectr. Anal.* **2011**, *31*, 2702–2705.
12. Agrawal, R.; Faloutsos, C.; Swami, A. Efficient similarity search in sequence database. In Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms, Chicago, IL, USA, 13–15 October 1993; Springer: London, UK, 1993; pp. 69–848.
13. Park, J.; Sriram, T.N. Robust estimation of conditional variance of time series using density power divergences. *J. Forecast.* **2017**, *36*, 703–717. [CrossRef]
14. Rojas, I.; Pomares, H. *Time Series Analysis and Forecasting*; Springer: Berlin/Heidelberg, Germany, 2016.
15. Das, G.; Lin, k.; Mannila, H.; Renganathan, G.; Smyth, P. Rule discovery from time series. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 27–31 August 1998.
16. Hsu, S.H.; Hsieh, P.A.; Chih, T.C.; Hsu, K.C. A two-stage architecture for stock price forecasting by integrating self-organizing map and support vector regression. *Expert Syst. Appl.* **2009**, *36*, 7947–7951. [CrossRef]
17. Huang, C.L.; Tsai, C.Y. A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. *Expert Syst. Appl.* **2009**, *36*, 1529–1539. [CrossRef]
18. Zhang, G.P. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* **2003**, *50*, 159–175. [CrossRef]
19. Folkes, S.R.; Lahav, O.; Maddox, S.J. An artificial neural network approach to the classification of galaxy spectra. *Mon. Not. R. Astron. Soc.* **2018**, *283*, 651–665. [CrossRef]
20. Tang, J.H. A review on the nonlinear time series model of regularly sampled data. *Math. Progress* **1989**, *18*, 22–43.
21. Xiong, Z.F. Wavelet Method for Fractal Dimension Estimation of Financial Time Series. *Syst. Eng. Theory Pract.* **2002**, *22*, 48–53.
22. Xu, M.; Huang, C. Financial Benefit Analysis and Forecast Based on Symbolic Time Series Method. *CMS* **2011**, *19*, 1–9.
23. Li, B.; Zhang, J.P.; Liu, X.J. Time-series Detection of Uncertain Anomalies Based on Hadoop. *Chin. J. Sens. Actuators* **2015**, *7*, 1066–1072.
24. Box, G.E.P.; Jenkins, G.M. Time Series Analysis: Forecasting and Control. *J. Time* **2010**, *31*, 303.
25. Xi, L.; Muzhou, H.; Lee, M.H.; Li, J.; Wei, D.; Hai, H.; Wu, Y. A new constructive neural network method for noise processing and its application on stock market prediction. *Appl. Soft Comput.* **2014**, *15*, 57–66. [CrossRef]
26. Kumar, K.M.; Reddy, A.R.M. A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method. *Pattern Recognit.* **2016**, *58*, 39–48. [CrossRef]
27. Limwattanapibool, O.; Arch-Int, S. Determination of the appropriate parameters for K-means clustering using selection of region clusters based on density DBSCAN (SRCD-DBSCAN). *Expert Syst.* **2017**, *34*, e12204. [CrossRef]
28. Wang, F.S.; Chen, L.H. *Particle Swarm Optimization (PSO)*; Springer: Berlin/Heidelberg, Germany, 2013.
29. Gou, J.; Lei, Y.X.; Guo, W.P.; Wang, C.; Cai, Y.Q.; Luo, W. A novel improved particle swarm optimization algorithm based on individual difference evolution. *Appl. Soft Comput.* **2017**, *57*, 468–481. [CrossRef]

30. Shah, G.H. An improved DBSCAN, a density based clustering algorithm with parameter selection for high dimensional data sets. In Proceedings of the Nirma University International Conference on Engineering, Ahmedabad, India, 28–30 November 2013.
31. Wei, W.; Jiang, J.; Liang, H.; Gao, L.; Liang, B.; Huang, J.; Zang, N.; Liao, Y.; Yu, J.; Lai, J.; et al. Application of a Combined Model with Autoregressive Integrated Moving Average (ARIMA) and Generalized Regression Neural Network (GRNN) in Forecasting Hepatitis Incidence in Heng County, China. *PLoS ONE* **2016**, *11*, e0156768. [CrossRef] [PubMed]
32. Wu, H.; Cai, Y.; Wu, Y.; Zhong, R.; Li, Q.; Zheng, J.; Lin, D.; Li, Y. Time series analysis of weekly influenza-like illness rate using a one-year period of factors in random forest regression. *BioSci. Trends* **2017**, *11*, 292–296. [CrossRef] [PubMed]
33. Lodwick, W.A.; Jamison, K.D. A computational method for fuzzy optimization. In *Uncertainty Analysis in Engineering and Sciences: Fuzzy Logic, Statistics and Neural Network Approach*; Avvub, B.M., Guota, M.M., Eds.; Kluwer Academic Publisher: Boston, MA, USA, 1998.