

Research Article

Prediction of Short-Term Stock Price Trend Based on Multiview RBF Neural Network

Bailin Lv^{1,2} and Yizhang Jiang ^{1,2}

¹School of Artificial Intelligence and Computer Science, Jiangnan University, 1800 Lihu Avenue, Wuxi 214122, Jiangsu, China

²Jiangsu Key Laboratory of Media Design and Software Technology, 1800 Lihu Avenue, Wuxi 214122, Jiangsu, China

Correspondence should be addressed to Yizhang Jiang; yzjiang@jiangnan.edu.cn

Received 8 October 2021; Revised 11 November 2021; Accepted 18 November 2021; Published 28 November 2021

Academic Editor: Qiangyi Li

Copyright © 2021 Bailin Lv and Yizhang Jiang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Stock price prediction is important in both financial and commercial domains, and using neural networks to forecast stock prices has been a topic of ongoing research and development. Traditional prediction models are often based on a single type of data and do not account for the interplay of many variables. This study covers a radial basis neural network modeling technique with multiview collaborative learning capabilities for incorporating the impacts of numerous elements into the prediction model. This research offers a multiview RBF neural network prediction model based on the classic RBF network by integrating a collaborative learning item with multiview learning capabilities (MV-RBF). MV-RBF can make full use of both the internal information provided by the correlation between each view and the distinct characteristics of each view to form independent sample information. By using two separate stock qualities as input feature information for trials, this study proves the viability of the multiview RBF neural network prediction model on a real data set.

1. Introduction

Predicting the fluctuation trend of stock prices plays an extremely important role in asset pricing, investment decision-making, risk management, and market supervision. Through the study of the autocorrelation function, power spectral density, and fluctuation range of stock prices, it is found that there are hidden long-term price linear trends and low-frequency periodic fluctuations in stock prices. This theoretically proves that stock prices that exhibit randomness and unpredictability at the micro level have overall certainty and predictability at the macro level [1]. Therefore, the prediction of stock price trends through machine learning and other methods has become an important topic nowadays.

The BP neural network is the most extensively used model for stock price prediction. White et al. were the first to use the BP neural network to forecast stock prices [2], and numerous improved approaches based on the BP neural network followed [3]. The BP neural network, on the other hand, is readily stuck in local minimums, causing the

model's prediction impact to deteriorate. By improving the SVM algorithm, Huang and Chen overcome to a certain extent the problem that the SVM algorithm can only obtain specific prediction values but cannot predict the trend of stock changes [4]. In addition, the choice of the kernel function of the SVM algorithm will also have a direct impact on the prediction accuracy. The LSTM algorithm can handle time series data problems well, but it may fall into a local optimal solution, and it has problems such as lag in prediction [5]. In addition to resolving the LSTM model's weaknesses, LSTM is frequently merged with other models to produce a hybrid model in order to increase LSTM's prediction performance [6].

All of the aforementioned model experiments are based on a single data viewpoint (view) [7]. Changes in stock prices, on the other hand, are influenced by a range of circumstances, which will eventually be represented in the stock's different characteristic data, such as the starting price, highest price, closing price, ups and downs, and trading volume. When compared to using single attribute stock data as experimental data, combining data from

multiple attributes to uncover the relationship between distinct attribute data without compromising the independence of each attribute data would certainly improve the model's generalization capabilities. This work presents a multiview collaborative learning technique [8] to attain this aim. A function with collaborative learning capacity is introduced based on the construction of a matching model for each view, allowing it to fully utilize the related information between the view sample data sets [9]. Multiview learning seeks to train one function to represent each view and then optimize all of the functions at the same time to increase generalization performance [10].

The RBF NN is used as the activation function in the RBF neural network's hidden layer to execute a fixed nonlinear transformation. The output layer of the RBF neural network is linearly integrated in the new space, mapping the low-dimensional input space to the new high-dimensional output space [11–13]. The linear property of such an output unit makes RBF neural network parameter change straightforward, and there is no local minimum difficulty. RBF neural network is also a useful basis for constructing multiview collaborative learning tasks, because it has strong multidimensional nonlinear mapping skills, generalization capabilities, and parallel information processing capabilities [14, 15].

The development of the field of artificial intelligence stems from the desire to give computers a way of thinking similar to human cognition. The core idea of the multiview learning method also hopes to include more effective information and explore the relevance and independence between views [16, 17]. The data of various characteristics represent the effect of numerous elements on the stock in the process of forecasting the trend of stock price variations, and the influence of different attributes on the final outcome is certain to be different. As a result, we must examine the degree of effect of diverse viewpoints on the final output in the multiview learning process [18]. Such a decision-making process is similar to our human thinking and decision-making process, which is also the advantage of multiview collaborative learning. In the future scenario of intelligent life and the interconnection of all things, the use of a single data source for experimental analysis cannot meet actual needs. At that point, we will evaluate the substance of the present scene based on the data information produced by different data sources. The application requirements are similar to those discussed in this paper's multiview learning approach. As a result, we have cause to assume that multiview learning technology, which incorporates many data sources, will be an area of neural network research that cannot be overlooked in the future.

In this paper's experimental effort, we start with the question, "How can we enhance the accuracy of stock price prediction using existing approaches even more?" By reading the relevant literature in recent years, we have focused on the multiview direction, because the multiview thinking is a more feasible method in both cognition and theory. Next, our focus is to find multiple perspectives on stock data. We discovered that stock data is made up of several different types of information, including the highest price, lowest price, opening price, and closing price. More complicated data gathering demonstrates

that multiview approaches may be used to analyze stock data. Another problem we encountered is what kind of algorithm can be used to capture the connection between different perspective data at the same time without destroying the independence of a single perspective data. After continuous attempts and improvements, we construct an objective function with collaborative learning ability. Simultaneously, the Lagrangian multiplier is applied to simplify the functional formula, which is then turned into a quadratic programming issue according to Lagrangian optimization theory. Finally, we tested the method's viability using numerous data sets in experiments.

The rest of this paper is laid out as follows. The radial basis neural network and the stock data processing approach are introduced in Section 2. Section 3 offers our multiview RBF classification model and builds a multiview learning framework. The experimental results are presented in Section 4. The conclusion is reached in Section 5.

2. Related Work

In this part, we will go through stock data processing procedures including shifting windows, data normalization, and other stages. The radial basis function neural network (RBF NN) is then introduced, along with related improvement strategies, to build the theoretical and technological groundwork for the creation of multiview models.

2.1. Stock Data Processing. The original stock data used in this paper comes from Lvjing real estate (ID 000502), which includes information on ten different attributes such as transaction date, opening price, closing price, highest price, lowest price, yesterday's closing price, change amount, change range, trading volume, and turnover. Except for the transaction date in the original data, other stock data can be regarded as dynamic time series. For such data, a time series analysis method is usually used. The method emphasizes that a region is continuously observed and calculated for a short period time to extract relevant features and analyze its change process.

The moving window approach is the most often used time series analysis method. Suppose there is a time series: $x = \{x_i | x_i \in R, i = 1, 2, \dots, L\}$, and the last m values in the series should be predicted by the first n data in the series. The first n data predictions are used as a moving window to map the value of the future time. Specifically, suppose we use the first 20 data to predict the 21st data for the first time, then, next, we add the 21st data from the previous prediction to the moving window, and delete the first data in the moving window. As a result, there are 20 new pieces of data in the window to anticipate the next data, and so on. To utilize the moving window approach, remove and add continually. When the window size is 20 [19–21], Figure 1 depicts a schematic representation of processing portion of stock data using the moving window approach.

Furthermore, data normalization is particularly crucial in data processing and subsequent studies, because different stock data is used as input for various perspectives. Take closing price and transaction volume as an example. When these two types of data are input as feature values, the value range is quite different.

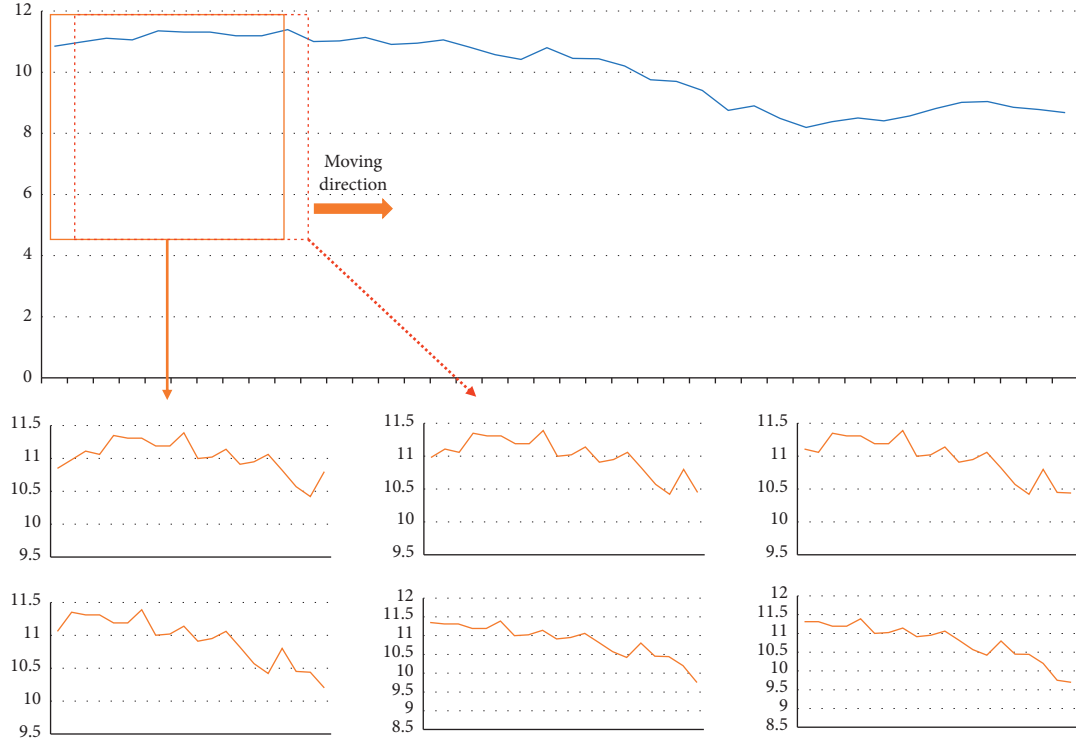


FIGURE 1: Moving window method for stock data processing.

It is necessary to avoid excessively large transaction volume from having too much influence on the forecast result, so the data needs to be normalized. Therefore, data normalization can control the range of the data within a reasonable range [22].

2.2. RBF NN Model. Figure 2 depicts the standard RBF NN model. The RBF NN is a single-hidden-layer forward network. The input layer is the first one. The concealed layer is the second layer. The number of nodes in the hidden layer is determined by the problem's requirements. The radial basis function (RBF), also known as the transformation function, of the buried layer neurons is a nonnegative linear function that is radially symmetric and attenuated to the center point. The low-dimensional pattern input data is translated into the high-dimensional space, and the input vector is changed into the high-dimensional space, so that the linearly inseparable problem in the low-dimensional space becomes linearly separable in the high-dimensional space. The third layer is the output layer, which reacts to input patterns [23].

In the RBF NN model, the $f: R^d \rightarrow R^1$ is the nonlinear mapping, and it can be expressed mathematically as

$$y^0 = f(x) = \sum_{i=1}^M \omega_i \Phi(\|x - c_i\|), \quad i = 1, \dots, M, \quad (1)$$

where $x \in R^d$ is the network's input vector, and $c_i \in R^d$ is the center vector of the network's hidden layer nodes' RBF NN. The connection weight between the network's hidden layer and the output node is ω_i . The norm is represented by $\|\cdot\|$, and

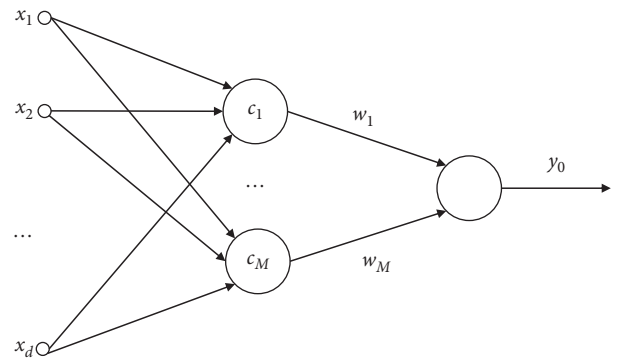


FIGURE 2: Classical RBF NN model structure.

here is the Euclidean norm. The RBF NN $\Phi(\cdot)$ completes the nonlinear transformation of $R^d \rightarrow R^1$ and has the following form:

$$\Phi(x - c_i) = \exp\left(-\frac{\|x - c_i\|^2}{\delta_i}\right), \quad (2)$$

where δ_i is the width value.

Three parameters are required in the RBF NN model presented above. The hidden layer center node $c_i = [c_i^1, \dots, c_i^k, \dots, c_i^d]^T$, the hidden layer width value δ_i , the hidden layer connection weight, and the output layer ω_i .

In the RBF NN model described above, three parameters are needed; namely, the hidden layer center node $c_i = [c_i^1, \dots, c_i^k, \dots, c_i^d]^T$, the width value of the hidden layer

δ_i , and the connection weight of the hidden layer and the output layer ω_i .

2.3. RBF NN with Linear Model. The central node parameter $c_i = [c_i^1, \dots, c_i^k, \dots, c_i^d]^T$ is estimated using the clustering approach, and the RBF NN's hidden layer width parameter is δ_i . The fuzzy C-means (FCM) clustering algorithm is used in this work [24], and $c_i = [c_i^1, \dots, c_i^k, \dots, c_i^d]^T$ and δ_i can be estimated by the following formula:

$$c_{ik} = \frac{\sum_{j=1}^N u_{ji} x_{jk}}{\sum_{j=1}^N u_{ji}}, \quad (3)$$

$$\delta_i^k = h \frac{\sum_{j=1}^N u_{ji} \|x - c_i\|^2}{\sum_{j=1}^N u_{ji}}, \quad (4)$$

where u_{ji} represents the fuzzy membership of the sample $x_j = [x_{j1}, \dots, x_{jd}]^T$ calculated by the FCM clustering method to the i -th category [25–27], and the parameter h is an adjustable scaling parameter.

If the hidden layer parameters of RBF NN are computed using the aforementioned clustering approach, then there are [8]

$$\tilde{x}^i = \Phi(x - c_i), \quad i = 1, \dots, M, \quad (5)$$

$$x_g = [\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^M]^T, \quad (6)$$

$$P_g = [\omega_1, \dots, \omega_M]^T. \quad (7)$$

The radial basis network's mapping function, presented in formula (1), may be written as

$$y^0 = P_g^T x_g. \quad (8)$$

When the hidden layer nodes of the radial basis network are computed using the clustering approach, the network's final output may be described as the output of a linear model using formula (8). In this way, the parameter learning process of the network is transformed into a parameter learning problem of a linear model. Based on this linear model, the ε -insensitive loss function is introduced to construct a multiview learning item, so that the radial basis network can meet the needs of multiview learning.

2.4. RBF NN Using ε -Insensitive Loss Function. The RBF NN's objective function, based on the ε -insensitive loss function, is defined as

$$\min_{P_g} \left. \begin{aligned} L &= \sum_{i=1}^N |y_i^0 - y_i| = \sum_{i=1}^N |P_g^T x_{gi} - y_i|_{\varepsilon} \\ \text{s.t.} &\begin{cases} y_i - P_g^T x_{gi} < \varepsilon + \xi_i^+ \\ P_g^T x_{gi} - y_i < \varepsilon + \xi_i^- \end{cases}, \forall i \end{aligned} \right\}. \quad (9)$$

By introducing slack variables ξ_i^+ and ξ_i^- , using the theory and method of quadratic programming, formula (9) may be rewritten as a standard quadratic programming problem, with the following particular form:

$$\min_{P_g, \xi^+, \xi^-, \varepsilon} \left. \begin{aligned} L(P_g, \xi^+, \xi^-, \varepsilon) &= \frac{1}{\tau} \cdot \frac{1}{N} \sum_{i=1}^N ((\xi_i^+)^2 + (\xi_i^-)^2) + \frac{1}{2} (P_g^T P_g) + \frac{2}{\tau} \cdot \varepsilon \\ \text{s.t.} &\begin{cases} y_i - P_g^T x_{gi} < \varepsilon + \xi_i^+ \\ P_g^T x_{gi} - y_i < \varepsilon + \xi_i^- \end{cases}, \forall i \end{aligned} \right\}. \quad (10)$$

Formula (10) introduces the Lagrange multiplier term, transforming formula (9) into the quadratic programming problem of formula (11),

$$\left. \begin{aligned} \arg \max_{\tilde{\alpha}} & -\tilde{\alpha}^T \tilde{H} \tilde{\alpha} + \tilde{\alpha}^T \beta \\ \text{s.t.} & \tilde{\alpha}^T = 1, \alpha_i \geq 0, \forall i, \end{aligned} \right\}, \quad (11)$$

where $\tilde{H} = [\tilde{h}_{ij}]_{2N \times 2N}$, $\tilde{h}_{ij} = z_i^T z_j + ((N\tau)/2)\delta_{ij}$, and $\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$.

The optimal solution of the parameters $(\lambda^+)^*$ and $(\lambda^-)^*$ can be achieved by solving formula (10). To obtain the best solution of the generalization term P_g of the RBF NN based on the ε -insensitive loss function, use formulas (12a) and (12d).

$$P_g = \frac{2}{\tau} \sum_{i=1}^N (\lambda_i^+ - \lambda_i^-) x_{gi}, \quad (12a)$$

$$\xi_i^+ = N\lambda_i^+, \quad (12b)$$

$$\xi_i^- = N\lambda_i^-, \quad (12c)$$

$$\begin{aligned} \varepsilon &= \sum_{i=1}^N (\lambda_i^+ - \lambda_i^-) x_{gi} - \frac{N}{2} ((\lambda_i^+)^2 + (\lambda_i^-)^2) \\ &\quad - \frac{1}{\tau} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i^+ - \lambda_i^-) (\lambda_j^+ - \lambda_j^-) \cdot x_{gi}^T x_{gj}. \end{aligned} \quad (12d)$$

The resultant parameters of the RBF NN based on the ε -insensitive loss function can be generated using the

antecedent parameters, such as the hidden layer center node and width parameters, calculated using formulas (3) and (4), in combination with the subsequent parameters calculated using formulas (12a) and (12d).

3. Multiview RBF NN Model Framework

A single-view data collection is typically used to build traditional RBF neural networks. For example, previous closing prices are utilized as training data to anticipate the rise and fall of the next trading day when predicting stock price trends. However, while each trading day generates many stock attribute data, only the closing price is utilized as training data, and the impact of other attribute data on prediction outcomes cannot be accounted for in the prediction model. When the traditional RBF network is dealing with a data set composed of multiple data (multiview), a more feasible strategy is to use samples from different views to construct a corresponding network model. This strategy provides a feasible solution for applications in multiview scenarios, but the correlation between the views cannot be reflected in this strategy. This will also lead to uneven generalization capabilities of prediction models obtained from different views.

This study proposes a modeling notion as illustrated in Figure 3 to express the correlation between data from distinct viewpoints in the model's prediction process, so that the new model may learn from numerous views. In this modeling strategy, data from multiple views are learned collaboratively, instead of training in isolation of sample data from each view. Such a strategy can improve the generalization ability of the new model for data from various views and improve the applicability of the model.

Figure 4 depicts the overall data input method into the multiview RBF model. The data from views A and B are entered into the RBF model corresponding to view A and view B, respectively, after preprocessing. The connection weight between the hidden layer and the output layer is defined by $(P_{g,A})^*$ and $(P_{g,B})^*$ calculated by the following training process in the RBF model's three-layer structure.

3.1. The Construction of Multiview Collaborative Learning Objective Function. The RBF NN is one of the most successful modeling approaches currently available. The least square error criteria are used to implement the standard RBF neural network. This sort of training strategy is best for small sample data sets or noisy data sets, as it is easy to overfit, reducing the RBF neural network's generalization capabilities. In order to address this issue, this work develops a new objective function by including the ε -insensitive loss function and structural risk term and then converts the suggested new objective function's solution into a standard quadratic programming problem. This successfully overcomes the overfitting problem as well as noise sensitivity flaws, while also demonstrating strong resilience.

In this section, dual views are taken as the specific research scene. Based on the traditional RBF classification model, the construction of the objective function of the multiview RBF classification model is discussed. To make full use of the independence of samples between different views and the correlation between data from different views while building a multiview RBF classification model, this study provides the following objective function formula with multiview learning capabilities.

$$\begin{aligned} & \text{Min}_{P_{g,A}, \xi_A^+, \xi_A^-, \varepsilon_A} l_A(P_{g,A}, \xi_A^+, \xi_A^-, \varepsilon_A) + l_B(P_{g,B}, \xi_B^+, \xi_B^-, \varepsilon_B) + l_S(\eta^+, \eta^-, \varepsilon_S) \\ & \text{s.t.} \left\{ \begin{array}{l} y_{i,A} - P_{g,A}^T x_{gi,A} < \varepsilon_A + \xi_{i,A}^+ \\ P_{g,A}^T x_{gi,A} - y_{i,A} < \varepsilon_A + \xi_{i,A}^- \\ y_{i,B} - P_{g,B}^T x_{gi,B} < \varepsilon_B + \xi_{i,B}^+ \\ P_{g,B}^T x_{gi,B} - y_{i,B} < \varepsilon_B + \xi_{i,B}^- \\ P_{g,B}^T x_{gi,B} - P_{g,A}^T x_{gi,A} < \varepsilon_S + \eta_i^+ \\ P_{g,A}^T x_{gi,A} - P_{g,B}^T x_{gi,B} < \varepsilon_S + \eta_i^- \end{array} \right. , \forall i, \end{aligned} \quad (13)$$

where

$$l_A(P_{g,A}, \xi_A^+, \xi_A^-, \varepsilon_A) = \frac{1}{N\tau_A} \sum_{i=1}^N \left[(\xi_{i,A}^+)^2 + (\xi_{i,A}^-)^2 \right] + \frac{1}{2} (P_{g,A}^T P_{g,A}) + \frac{2}{\tau_A} \cdot \varepsilon_A, \quad (13a)$$

$$l_B(P_{g,B}, \xi_B^+, \xi_B^-, \varepsilon_B) = \frac{1}{N\tau_B} \sum_{i=1}^N \left[(\xi_{i,B}^+)^2 + (\xi_{i,B}^-)^2 \right] + \frac{1}{2} (P_{g,B}^T P_{g,B}) + \frac{2}{\tau_B} \cdot \varepsilon_B, \quad (13b)$$

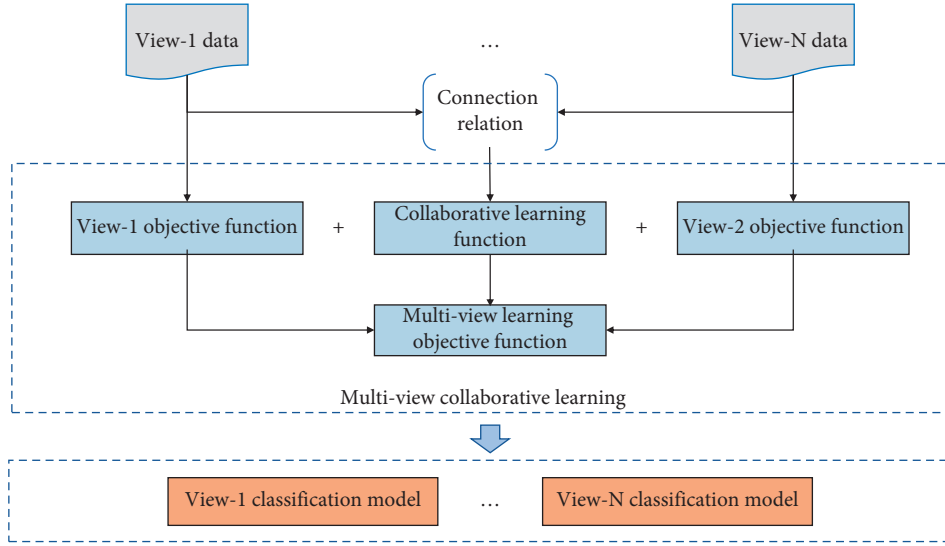


FIGURE 3: Multiview RBF classification model learning framework.

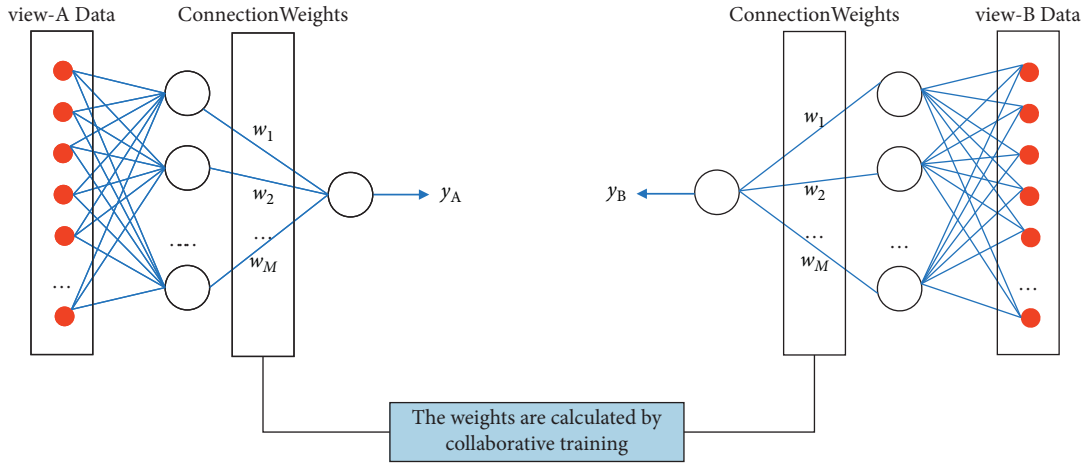


FIGURE 4: Data Processing in the model.

$$l_S(\eta^+, \eta^-, \varepsilon_S) = \frac{1}{N\tau_S} \sum_{i=1}^N [(\eta_i^+)^2 + (\eta_i^-)^2] + \frac{2}{\tau_S} \cdot \varepsilon_S. \quad (13c)$$

In formula (13), $l_A(P_{g,A}, \xi_A^+, \xi_A^-, \varepsilon_A)$ and $l_B(P_{g,B}, \xi_B^+, \xi_B^-, \varepsilon_B)$ are the objective function terms of the RBF classification model established based on the data samples of two independent views A and B. These two ensure that new approaches and strategies may take full use of variances in the characteristics of data samples from different perspectives, resulting in a classification model that is specific to the view's features.

The new modeling technique takes into account not only the changes in sample features between views (independent information), but also the capacity of multiview learning to discover the link between views, i.e., the consistency of each view's decision output. This study creates formula (13c) with collaborative learning capacity in response to such demands. Formula

(13c) can make view A and view B undertake collaborative learning. Finally, the output of the classification model corresponding to each view tends to be consistent, and the difference in the generalization performance of the classification model of each view caused by the change of perspective feature is avoided.

It should be noted that the regularization parameter $\tau_A > 0$, $\tau_B > 0$, $\tau_S > 0$ controls the complexity and degree of error of the classification model, and its value is manually set to select an appropriate value [28–30].

3.2. MV-RBF Parameter Learning Rules. Continuing to derive the objective function formula, the Lagrange equation is as follows:

$$\begin{aligned}
J = & \frac{1}{N\tau_A} \sum_{i=1}^N \left[(\xi_{i,A}^+)^2 + (\xi_{i,A}^-)^2 \right] + \frac{1}{2} (P_{g,A}^T P_{g,A}) + \frac{2}{\tau_A} \cdot \varepsilon_A + \\
& \frac{1}{N\tau_B} \sum_{i=1}^N \left[(\xi_{i,B}^+)^2 + (\xi_{i,B}^-)^2 \right] + \frac{1}{2} (P_{g,B}^T P_{g,B}) + \frac{2}{\tau_B} \cdot \varepsilon_B + \frac{1}{N\tau_S} \sum_{i=1}^N \left[(\eta_i^+)^2 + (\eta_i^-)^2 \right] \\
& + \frac{2}{\tau_S} \cdot \varepsilon_S + \sum_{i=1}^N \lambda_{i,A}^+ (y_{i,A} - P_{g,A}^T x_{gi,A} - \varepsilon_A - \xi_{i,A}^+) + \sum_{i=1}^N \lambda_{i,A}^- (P_{g,A}^T x_{gi,A} - y_{i,A} - \varepsilon_A - \xi_{i,A}^-) \\
& + \sum_{i=1}^N \lambda_{i,B}^+ (y_{i,B} - P_{g,B}^T x_{gi,B} - \varepsilon_B - \xi_{i,B}^+) + \sum_{i=1}^N \lambda_{i,B}^- (P_{g,B}^T x_{gi,B} - y_{i,B} - \varepsilon_B - \xi_{i,B}^-) + \\
& \sum_{i=1}^N \beta_i^+ (P_{g,B}^T x_{gi,B} - P_{g,A}^T x_{gi,A} - \varepsilon_S - \eta_i^+) + \sum_{i=1}^N \beta_i^- (P_{g,A}^T x_{gi,A} - P_{g,B}^T x_{gi,B} - \varepsilon_S - \eta_i^-).
\end{aligned} \tag{14}$$

According to the partial derivative of the variable in equation (14) being 0, the following relationship can be derived:

$$P_{g,A} = \sum_{i=1}^N (\lambda_{i,A}^+ - \lambda_{i,A}^-) x_{gi,A} + \sum_{i=1}^N (\beta_{i,S}^+ - \beta_{i,S}^-) x_{gi,A}, \tag{15}$$

$$P_{g,B} = \sum_{i=1}^N (\lambda_{i,B}^+ - \lambda_{i,B}^-) x_{gi,B} + \sum_{i=1}^N (\beta_{i,S}^+ - \beta_{i,S}^-) x_{gi,B}, \tag{16}$$

$$\xi_{i,A}^+ = \frac{N\tau_A}{2} \lambda_{i,A}^+, \tag{17}$$

$$\xi_{i,A}^- = \frac{N\tau_A}{2} \lambda_{i,A}^-, \tag{18}$$

$$\xi_{i,B}^+ = \frac{N\tau_B}{2} \lambda_{i,B}^+, \tag{19}$$

$$\xi_{i,B}^- = \frac{N\tau_B}{2} \lambda_{i,B}^-, \tag{20}$$

$$\eta_i^+ = \frac{N\tau_S}{2} \beta_i^+, \tag{21}$$

$$\eta_i^- = \frac{N\tau_S}{2} \beta_i^-, \tag{22}$$

$$\frac{2}{\tau_A} = \sum_{i=1}^N (\lambda_{i,A}^+ - \lambda_{i,A}^-), \tag{23}$$

$$\frac{2}{\tau_B} = \sum_{i=1}^N (\lambda_{i,B}^+ - \lambda_{i,B}^-), \tag{24}$$

$$\frac{2}{\tau_S} = \sum_{i=1}^N (\beta_i^+ - \beta_i^-). \tag{25}$$

Substituting formula (15) to (25) into (14), and deriving formula (26) after simplification,

$$\begin{aligned}
& \arg \max_{\alpha} \quad \frac{1}{2} \alpha^T H \alpha + \alpha^T \beta \\
& \text{s.t.} \quad \left\{ \begin{array}{l} \alpha^T L_A = \frac{2}{\tau_A}, \\ L_A = \left[1, \dots, 1, 0, \dots, 0, 0, \dots, 0 \right] \\ \alpha^T L_B = \frac{2}{\tau_B}, \\ L_B = \left[0, \dots, 0, 1, \dots, 1, 0, \dots, 0 \right], \\ \alpha^T L_S = \frac{2}{\tau_S}, \\ L_B = \left[0, \dots, 0, 0, \dots, 0, 1, \dots, 1 \right], \end{array} \right. \quad \alpha_i \geq 0, \forall i
\end{aligned} \tag{26}$$

where

$$\tilde{H}_A = [\tilde{h}_{ij,A}]_{2N \times 2N},$$

$$\tilde{h}_{ij,A} = z_{i,A}^T z_{j,A} + \left(\frac{N\tau_A}{2} \right) \delta_{ij,A},$$

$$\delta_{ij,A} = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases}$$

(26a)

$$\begin{aligned}\tilde{H}_B &= [\tilde{h}_{ij,B}]_{2N \times 2N}, \\ \tilde{h}_{ij,B} &= z_{i,B}^T z_{i,B} + \left(\frac{N\tau_B}{2}\right) \delta_{ij,B},\end{aligned}\quad (26b)$$

$$\delta_{ij,B} = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases}$$

$$\begin{aligned}\tilde{H}_S &= [\tilde{h}_{ij,S}]_{2N \times 2N}, \\ \tilde{h}_{ij,S} &= -3z_{i,B}^T z_{i,B} + z_{i,A}^T z_{i,A} + \left(\frac{N\tau_S}{2}\right) \delta_{ij,S},\end{aligned}\quad (26c)$$

$$\delta_{ij,S} = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases}$$

$$\begin{aligned}\tilde{H}_{AS} &= [\tilde{h}_{ij,AS}]_{2N \times 2N}, \\ \tilde{h}_{ij,AS} &= z_{i,A}^T z_{i,A},\end{aligned}\quad (26d)$$

$$\begin{aligned}\tilde{H}_{BS} &= [\tilde{h}_{ij,BS}]_{2N \times 2N}, \\ \tilde{h}_{ij,BS} &= z_{i,B}^T z_{i,B},\end{aligned}\quad (26e)$$

$$H = \begin{pmatrix} \tilde{H}_A & 0 & \tilde{H}_{AS} \\ 0 & \tilde{H}_B & -\tilde{H}_{BS} \\ \tilde{H}_{AS} & -\tilde{H}_{BS} & \tilde{H}_S \end{pmatrix}.\quad (26f)$$

At this point, formula (14) has been transformed into a classic quadratic programming problem as shown in formulas (26), and (26) is solved using the quadratic programming solution method in literature [8]. After completing the quadratic programming calculation, the optimized parameters $(\lambda_A^+)^*$, $(\lambda_A^-)^*$, $(\lambda_B^+)^*$, $(\lambda_B^-)^*$, $(\beta_S^+)^*$, $(\beta_S^-)^*$ corresponding to formula (26) are obtained. Using these optimized parameters, we can derive the subsequent parameters of the optimized RBF classification model under each view as follows:

$$(P_{g,A})^* = \sum_{i=1}^N ((\lambda_{i,A}^+)^* - (\lambda_{i,A}^-)^*) x_{gi,A} + \sum_{i=1}^N ((\beta_{i,S}^+)^* - (\beta_{i,S}^-)^*) x_{gi,A},\quad (27a)$$

$$(P_{g,B})^* = \sum_{i=1}^N ((\lambda_{i,B}^+)^* - (\lambda_{i,B}^-)^*) x_{gi,B} + \sum_{i=1}^N ((\beta_{i,S}^+)^* - (\beta_{i,S}^-)^*) x_{gi,B}.\quad (27b)$$

Formula (26a) is the subsequent parameter of the RBF classification model corresponding to view A, and formula (26b) is the subsequent parameter of the RBF classification model corresponding to view B. As mentioned above, the cluster center and width parameters calculated by formulas (3) and (4) are used as the antecedent parameters of the multiview RBF classification model. This gives us access to all of the parameters that make up the multiview RBF classification model. Algorithm 1 depicts the multiview RBF classification model (MV-RBF) learning algorithm; please see the detailed Algorithm 1 as follows:

4. Experiment

In this part, we create two sets of experiments to test the performance of the multiview RBF classification model in stock trend prediction and compare and contrast the multiview RBF classification model to other algorithm models. To prove MV practicality, RBF's first set of trials compares it to the classic RBF classification model. The second set of trials compares MV-RBF against a well-used classification technique.

4.1. Setup. To verify the feasibility of the multiview RBF classification model (MV-RBF) proposed in this paper, this section selects Lvjing real estate (ID 000502) from January 2, 2018, to December 31, 2019, for a total of 473 trading days. The various indicators constitute the original data set. We

picked the two characteristics of the closing price and volume in the original data set as the two perspectives of the experiment and processed them into two data sets D_1 and D_2 using the moving window approach and data normalization described in the previous study. The data sets D_1 and D_2 have a window size of 20, which means that the input data has a dimension of 20. Part of the data in the data set for data specification is shown in Table 1.

Table 1 shows part of the data in a single-view data set. Each row is used as an input, and the last column of each row represents the fluctuation of the predicted trading day and also serves as the two categories of the model classification. In addition, we changed the size of the moving window, that is, the dimension of the input data, to find a suitable window size to improve the accuracy of classification. We also chose two other features from the original data set as two perspectives, such as opening and closing prices, opening and highest prices, to see how the choice of view affected the categorization findings.

Some symbolic representations involved in the experiment are defined as follows: D_1 and D_2 represent the training data set of view 1 and view 2 in the multiview scene, respectively, and D_{1_test} , D_{2_test} are the test data sets. $[D_1, D_2]$ represents the single-view training data set obtained by combining the training data sets of view 1 and view 2 by the feature expansion method, and $[D_1, D_2]_{-test}$ is the test data sets. RBF (View- D_1), RBF (View- D_2) and RBF (View- $[D_1, D_2]$) are classification models constructed by traditional RBF neural networks based on the D_1 , D_2 and

Step 1: Set the regularization parameters τ_A, τ_B, τ_S , as well as the RBF network's hidden layer nodes M_A, M_B ;

Step 2: Use the FCM algorithm or other algorithm with the ability to divide the sample space to generate the data sets D_A, D_B corresponding to the classification model of each view sample, and the corresponding cluster centers $c_{ik,A}, c_{ik,B}$ and width parameters $\delta_{i,A}^k, \delta_{i,B}^k$;

Step 3: Solve with formula (14) to get the Lagrange multiplier $(\lambda_A^+)^*, (\lambda_B^+)^*, (\lambda_B^-)^*, (\beta_S^+)^*, (\beta_S^-)^*$;

Step 4: According to the solution of dual problem formula (14) obtained in Step 3, use formulas (25a) and (25b) to find $(P_{g,A})^*, (P_{g,B})^*$;

Step 5: According to the parameters of each view obtained in Step 4, an RBF classification model that meets the characteristics of each view can be generated;

ALGORITHM 1: MV-RBF algorithm.

TABLE 1: A sample of the information in the data set. D_1

	1	2	...	19	20	21
1	0.8841	0.9252	...	0.8766	0.8000	0
2	0.9252	0.9159	...	0.8000	0.7664	0
3	0.9159	0.9645	...	0.7664	0.7701	1
4	0.9645	0.9738	...	0.7701	0.6673	1
5	0.9738	0.9589	...	0.6673	0.6654	0

$[D_1, D_2]$ data sets. The multiview RBF neural network modeling techniques MV-RBF (View- D_1) and MV-RBF (View- D_2) suggested in this study are based on the classification model generated by D_1 and D_2 .

Since this paper is to predict the rise and fall of stock prices, we choose the accuracy that is often used in classification tasks. The accuracy rate can reflect the proportion of the correct prediction in the test sample.

The manual setting parameters involved in the experiment process include regularization parameters τ_A, τ_B, τ_S and the number of hidden layer nodes of the RBF neural network M , all of which use the parameter ranges given by cross-validation in Table 2.

B. Comparison of MV-RBF and traditional RBF classification model.

We used the moving window method and data normalization to standardize the form of the data set in this section of the experiment. We selected the closing price and volume data as the experimental data sets D_1 and D_2 and used the moving window method and data normalization to standardize the form of the data set. We used a basic cross-validation procedure, scrambled the data sets D_1 and D_2 , and ran many trials to further demonstrate the practicality of MV-RBF. Table 3 shows the outcomes of the experiment.

Six sets of experimental data are shown in Table 3. The classification impact of the multiview RBF model is higher than that of the classic RBF model, according to the experimental data. This demonstrates that when compared to the traditional single-view method, the method of constructing multiview in this paper has the ability of multiview learning, as it can more effectively utilize the independent feature space information in the data set of each view and the correlation between views. The effectiveness of the classification approach used in this study may be validated.

The 6 groups of experimental results of the traditional RBF classification model on 3 data sets in Table 3 can show that the single-view sample $[D_1, D_2]$ artificially constructed

by the feature expansion method can balance the difference between views to a certain extent. This shows that although there is a certain correlation between the various views, the differences between the views are also obvious. In most circumstances, the classification effect of the single-view sample $[D_1, D_2]$ based on the standard RBF model is lower than the classification effect of the classification model based on the technique of this study in any view due to this disparity.

During the experiment, we also encountered some other situations. In the fourth set of experiments in Table 3, the classification accuracy of the view D_1 dataset under the two classification models is the same. No matter how the parameters of MV-RBF are adjusted, the accuracy will not be significantly improved. In addition, in the experimental group not shown in Table 3, there are similar situations where the classification effects of the two models for the same data set are similar. With such a result, we infer that the two data sets failed to provide useful spatial feature information for each other in the process of collaborative learning, or the data sets of the two views selected in the experiment are data of two similar attributes of stocks. Even if the collaborative training process is carried out, due to the similar attributes, there is not much correlation between the two data sets, resulting in the fact that the model classification effect is not much improved.

4.2. Comparison of Multiview RBF Classification Model with Other Commonly Used Classification Models. In the preceding part, we compared the multiview method's classification impact to that of the classic RBF classification model, demonstrating the multiview RBF classification model's efficacy. In this part, we compare the multiview approach presented in this research against other regularly used categorization methods in order to confirm its viability. For comparison, we used three frequently used classification

TABLE 2: Parameter settings of all classification models.

Models	Parameters	Search rangers
RBF	Number of hidden nodes M	$\{10, 11, \dots, 19, 20\}$
MV-RBF	Regularization parameter τ_A	$\{10^{-3}, 10^{-2}, \dots, 10^1\}$ or $\{2^0, 2^1, \dots, 2^4\}$
	Regularization parameter τ_B	$\{10^{-3}, 10^{-2}, \dots, 10^1\}$ or $\{2^0, 2^1, \dots, 2^4\}$
	Regularization parameter τ_S	$\{2^0, 2^1, \dots, 2^6\}$
	Number of hidden nodes M	$\{10, 11, \dots, 19, 20\}$

TABLE 3: A comparison of RBF and MV-RBF using accuracy index.

D_1/D_2	RBF			MV-RBF	
	D_1	D_2	$[D_1, D_2]$	D_1	D_2
1	0.5824	0.5495	0.5663	0.6154	0.5714
2	0.5055	0.5934	0.5862	0.5604	0.6044
3	0.5604	0.4615	0.5062	0.6044	0.5165
4	0.6923	0.4286	0.4941	0.6923	0.4945
5	0.5495	0.4615	0.5529	0.6484	0.5385
6	0.4954	0.4396	0.4494	0.5934	0.4725

methods: decision tree, support vector machine (SVM), and closest neighbor classifier (KNN), and the experimental results are displayed in Table 4.

The experimental findings in Table 4 show that the multiview RBF classification model presented in this research can be used for stock prediction analysis in the vast majority of situations. Except for individual studies, the MV-RBF model has a somewhat greater classification impact than other regularly used classification models. This also demonstrates that, during model training, the RBF classification approach with multiview learning capability can generate the classification model for each view by fully using the independent component and relevance component of the multiview data set. Whether we look at these trained models from an independent single point of view or the correlation between several points of view, the results suggest that the strategy in this research is feasible and successful. As a result, the classification model created using the approach described in this study may forecast stock price trends and can improve the classification impact of the single-view classification model.

4.3. Experimental Result of MV-RBF Classification Model on Other Data Sets. To further validate the efficiency of the strategy in this research, we picked stock data from different industries and sectors and processed them into acceptable data sets for experimentation. Table 5 shows the experimental findings of the SAIC Motor stock data set with the stock code 600104, while Table 5 shows the experimental results of the GREE Electric stock data set with the stock code ID 000651.

The aforementioned experimental findings on various data sets demonstrate that the MV-RBF classification model performs better than the classic RBF classification model, demonstrating the efficacy of the strategy presented in this study.

4.4. Parameter Sensitivity Analysis and Cross-Validation to Determine the Approximate Range of Parameters. We employed the approach of controlled variables to examine the sensitivity of several parameters in the MV-RBF model in this section of the experiment. The number of nodes M in the RBF NN's hidden layer, and the regularization parameters τ_A , τ_B , τ_S in the multiview learning ability formula, are among these parameters. Compare the impact of adjusting these parameter values on the categorization model. Table 6 illustrates the results of experiments with various numbers of hidden layer nodes.

According to the experimental results in Table 6 and Figure 5, changing the number of hidden nodes in the RBF neural network had no significant effect on the experimental results, but it can be seen that the experimental data is roughly within a certain range after multiple sets of repeated experiments. This is because the RBF neural network's hidden layer employs the FCM clustering algorithm, which redivides the cluster centers with each trial; that is, it recalculates the relevant membership degree u_{ji} and the width parameter h (refer to Section 2). In addition, the regularization parameters τ_A , τ_B , τ_S in the multiperspective learning ability formula are also carried out in related experiments, and the experimental results are similar to Table 6. We find that the multiview RBF neural network classification model is less susceptible to parameters but has a specific active interval, thanks to the RBF neural network's hidden layer clustering mechanism.

There is currently no correct theoretical analysis or methodology for determining the ideal value of the insensitive parameter ε ; however, this type of parameter is analogous to the insensitive parameter ε in the support vector regression (SVR) approach. The optimal value of this parameter and the variance of the noise in the data have an approximately proportional relationship, meaning that the greater the noise, the larger the value of this parameter usually required to obtain a good training effect, according to a theoretical study of this parameter in SVR.

TABLE 4: Comparison of accuracy between MV-RBF and other classification models.

	MV-RBF		Decision tree		Naive Bayes		SVM		KNN		Subspace KNN	
	The view of D_1	The view of D_2	The view of D_1	The view of D_2	The view of D_1	The view of D_2	The view of D_1	The view of D_2	The view of D_1	The view of D_2	The view of D_1	The view of D_2
1	0.6154	0.5714	0.6120	0.5240	0.5180	0.5460	0.5990	0.4890	0.4820	0.5040	0.4780	0.4890
2	0.5604	0.6044	0.5640	0.5480	0.5130	0.5310	0.6060	0.5200	0.5240	0.5110	0.4600	0.5130
3	0.6044	0.5165	0.5750	0.5510	0.5290	0.5480	0.5750	0.5590	0.5260	0.5150	0.4520	0.5180
4	0.6923	0.4945	0.6080	0.5550	0.4910	0.5310	0.5970	0.5330	0.5040	0.5180	0.4450	0.5240
5	0.6484	0.5385	0.5810	0.4760	0.4890	0.5480	0.5930	0.5200	0.4980	0.4980	0.4780	0.5020
6	0.5923	0.4725	0.5620	0.5260	0.5020	0.5350	0.6080	0.5510	0.4800	0.4850	0.4600	0.4850
7	0.5275	0.5385	0.6430	0.5240	0.5220	0.5310	0.5900	0.5460	0.4800	0.4820	0.4540	0.5240
8	0.6044	0.5055	0.5900	0.5200	0.5290	0.5510	0.5700	0.5460	0.4800	0.5020	0.4800	0.4910
9	0.6154	0.5165	0.6100	0.5400	0.5550	0.5400	0.6150	0.5150	0.5400	0.4890	0.4780	0.5000
10	0.5385	0.4835	0.5620	0.5370	0.5070	0.5570	0.5680	0.5680	0.5240	0.5290	0.4890	0.5370

TABLE 5: Experimental results on other data sets.

D_1/D_2	RBF		MV-RBF	
	The view of D_1	The view of D_2	The view of D_1	The view of D_2
1	0.5824	0.4176	0.6154	0.4835
2	0.5385	0.5385	0.5824	0.5495
3	0.6154	0.5165	0.6484	0.5495
4	0.4835	0.4725	0.5604	0.5495
5	0.4945	0.5934	0.5165	0.6154
6	0.5165	0.5604	0.5385	0.5714
7	0.4725	0.4615	0.5165	0.4395
8	0.4835	0.4286	0.5275	0.4615
9	0.5495	0.4945	0.5385	0.5604
10	0.4835	0.5495	0.5385	0.5714

TABLE 6: Parameter sensitivity analysis.

	$M = 5$		$M = 10$		$M = 15$		$M = 18$		$M = 20$	
	D_1	D_2	D_1	D_2	D_1	D_2	D_1	D_2	D_1	D_2
1	0.4505	0.5385	0.4615	0.5714	0.5385	0.5495	0.4945	0.5495	0.4505	0.5165
2	0.4505	0.5055	0.4505	0.4835	0.4176	0.5495	0.4066	0.5714	0.4396	0.5835
3	0.5165	0.5604	0.5495	0.5824	0.6154	0.4835	0.5714	0.5604	0.5385	0.5385
4	0.5165	0.5055	0.4505	0.5714	0.4396	0.5385	0.4725	0.5604	0.3956	0.4835
5	0.4945	0.5714	0.5604	0.4604	0.5275	0.4835	0.4505	0.4725	0.4505	0.4945

The artificially set parameters τ_A , τ_B , τ_S involved in the experiment process control the complexity and degree of error of the classification model. The experimental results of finding the approximate range of the regularization parameters τ_A , τ_B , τ_S are shown in Table 7.

4.5. Complexity Analysis. The temporal complexity of the MV-RBF classification method is mostly made up of two parts, as illustrated in Algorithm 1: the learning of the antecedent parameters (corresponding to Step 2) and the learning of the subsequent parameters (corresponding to Steps 3 and 4). In multiview learning, the antecedent parameters of the classification model corresponding to each view of the MV-RBF algorithm are obtained by the FCM clustering algorithm, and its time complexity is $O(MTN)$, where M is the number of

views, T is the number of iterations in the algorithm, and N is the number of samples. The difficulty of determining the extreme value of quadratic programming is the key to learning the subsequent parameters of each MV-RBF view. The time complexity is $O(N^3)$. By using decomposition methods such as SMO to deal with secondary planning problems, it is possible to lower the time complexity to $O(N^2)$. Therefore, in multiview learning, the time complexity of MV-RBF subsequent parameter learning is between $O(MN^2) \sim O(MN^3)$. The traditional RBF classification model used in this paper has a time complexity of only $O(TN) + O(N^2)$, so the multiview MV-RBF algorithm is inferior to the traditional single-view RBF classification model in terms of time performance. The main reason is that the MV-RBF modeling strategy introduces a multiview collaborative learning method, while the traditional single-view RBF classification model does not have multiview

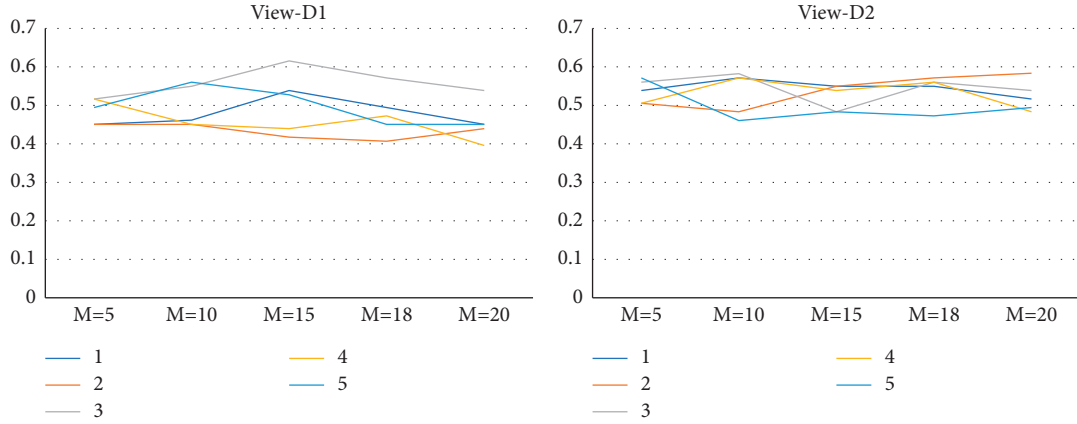


FIGURE 5: Parameter sensitivity line chart.

TABLE 7: Cross-validation regularized parameter range.

D_1/D_2	τ_A	τ_B	τ_S	D_1	D_2
1	10^0	1	2^0	0.5495	0.5055
	10^{-1}	1	2^0	0.5385	0.5275
	10^{-2}	1	2^0	0.5385	0.5495
	10^{-3}	1	2^0	0.5275	0.5385
	10^{-4}	1	2^0	0.5275	0.5385
2	1	10^0	2^0	0.5495	0.5165
	1	10^{-1}	2^0	0.5604	0.5165
	1	10^{-2}	2^0	0.5604	0.4945
	1	10^{-3}	2^0	0.5495	0.4835
	1	10^{-4}	2^0	0.5495	0.4835
3	1	1	2^0	0.5495	0.5055
	1	1	2^1	0.5275	0.5385
	1	1	2^2	0.4945	0.5385
	1	1	2^3	0.4945	0.5385
	1	1	2^4	0.4945	0.5385

learning capabilities, so the MV-RBF classification model is more time-consuming. However, the generalization performance of MV-RBF has a certain improvement compared with the traditional classification model.

5. Conclusion

This study builds on the standard single-view RBF classification model by including the concept of multiview collaborative learning and establishing a new multiview RBF classification model based on the independence and relevance information between the views. This approach successfully uses the independence component in the multiview data set as well as the correlation component between the views to improve the classification model's accuracy under each view.

The efficiency and practicality of the strategy in this research are demonstrated by a simulated experiment on a stock data set. However, owing to the complexity of the multiview methodology, this paper's study only covers the particular modeling approach of MV-RBF from a dual-view perspective. In the future, we will focus on developing a more compact and appropriate multiview categorization

model. Simultaneously, the use of other traditional classification techniques in multiview scenarios will be examined.

Data Availability

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

Conflicts of Interest

All the authors do not have any conflicts of interest.

Acknowledgments

This research was funded in part by the National Natural Science Foundation of China (Grant 62171203), the Natural Science Foundation of Jiangsu Province (Grant BK20210449), the 2018 Six Talent Peaks Project of Jiangsu Province (Grant XYDXX-127), the Science and Technology Demonstration Project of Social Development of Wuxi (Grant WX18IVJN002), and in part by the Science and Technology Demonstration Project of Social Development of Jiangsu Province under (Grant BE2019631).

References

- [1] J. J. Murphy, *Technical Analysis of the Financial Markets*, Prentice Hall Press, Hoboken, NJ, USA, 1999.
- [2] H. White, "Economic prediction using neural networks: the case of IBM daily stock returns," in *Proceedings of the IEEE 1988 International Conference On Neural Networks*, New York, NY, USA, July 1988.
- [3] H. Zhang, "The forecasting model of stock price based on PCA and BP neural network," *Journal of Financial Risk Management*, vol. 7, no. 4, pp. 369–385, 2018.
- [4] T. Y. Huang and F. F. Chen, "Application of kernel function of stock price forecasting based on SVM," *Journal of Chongqing University of Technology (Natural Science)*, vol. 2, pp. 89–94, 2016.
- [5] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data," *PLoS One*, vol. 14, no. 2, p. e0212320, 2019.
- [6] A. Moghar and M. Hamiche, "Stock Market Prediction Using LSTM Recurrent Neural Network," *Procedia Computer Science*, vol. 170, pp. 1168–1173, 2020.
- [7] Y. Z. Yizhang Jiang, F. L. Fu-Lai Chung, and S. T. Shitong Wang, "Collaborative fuzzy clustering from multiple weighted views," *IEEE Transactions on Cybernetics*, vol. 45, no. 4, pp. 688–701, 2015.
- [8] Z. H. Deng, Y. Z. Jiang, and K. Choi, "Knowledge-leverage-based TSK fuzzy system modeling," *IEEE Transactions on Neural Networks and Learning System*, vol. 24, no. 8, pp. 1200–1212, 2013.
- [9] Y. Z. Jiang, Z. H. Deng, and J. Wang, "Collaborative partition multi-view fuzzy clustering algorithm using entropy weighting," *Journal of Software*, vol. 25, no. 10, pp. 2293–2311, 2014.
- [10] S. L. Zhang, X. Yu, and Y. Sui, "Object tracking with multi-view support vector machines[J]," *IEEE Transactions on Multimedia*, vol. 17, no. 3, pp. 265–278, 2015.
- [11] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.
- [12] Y. Lan, Y. C. Soh, and G. B. Huang, "Constructive hidden nodes selection of extreme learning machine for regression," *Neurocomputing*, vol. 73, no. 16–18, pp. 3191–3199, 2010.
- [13] G. B. Huang, X. J. Ding, and H. M. Zhou, "Optimization method based extreme learning machine for classification," *Neurocomputing*, vol. 74, no. 1–3, pp. 155–163, 2011.
- [14] P. Zhang, X. Zhou, and P. Pelliccione, "RBF-MLMR: a multi-label metamorphic relation prediction approach using RBF neural network," *IEEE Access*, vol. 5, 2017.
- [15] X. Meng, P. Rozycki, and J. F. Qiao, "Nonlinear system modeling using RBF networks for industrial application," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, 2018.
- [16] G. F. Tzortzis and A. C. Likas, "Multiple view clustering using a weighted combination of exemplar-based mixture models," *IEEE Transactions on Neural Networks*, vol. 21, no. 120, pp. 1925–1938, 2010.
- [17] S. Merugu, S. Rosset, and C. Perlich, "A new multi-view regression approach with an application to customer wallet estimation," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 656–661, Philadelphia, PA, USA, August 2006.
- [18] G. X. Li, K. Y. Chang, and S. C. H. Hoi, "Multi-view semi-supervised learning with consensus," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 11, pp. 2040–2051, 2012.
- [19] L. Zheng, W. Hu, and Y. Min, "Raw wind data preprocessing: a data-mining approach," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 1, pp. 11–19, 2015.
- [20] Z. B. Zhang and W. Liu, "Analysis of data preprocessing technology in data mining," *Digital Technology and application*, vol. 10, pp. 216–217, 2017.
- [21] W. Shi, Y. Zhu, T. Huang, G. Sheng, Y. Lian, and G. Wang, "An integrated data preprocessing framework based on Apache spark for fault diagnosis of power grid equipment," *Journal of Signal Processing System*, vol. 86, no. 2-3, pp. 221–236, 2017.
- [22] Z. Q. Zhan, J. Y. Zhao, Y. Zhang, J. T. Gong, and Q. Y. Wang, "Grabbing the Long Tail: a data normalization method for diverse and information dialogue generation," *Neurocomputing*, vol. 460, pp. 374–384, 2021.
- [23] J. L. Tao, Z. Yu, R. D. Zhang, and F. R. Gao, "RBF neural network modeling approach using PCA based LM-GA optimization for coke furnace system," *Applied Soft Computing*, vol. 111, 2021.
- [24] Y. Z. Jiang, F. L. Chung, and S. T. Wang, "Enhanced fuzzy partitions vs data randomness in FCM," *Journal of Intelligent and Fuzzy Systems*, vol. 27, no. 4, pp. 1639–1648, 2014.
- [25] J. S. R. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft-Computing*, Prentice-Hall, Upper Saddle River, NJ, USA, 1997.
- [26] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, NY, USA, 1998.
- [27] S. T. Wang, *Neural Fuzzy System and its Application*, Beijing University of Aeronautics and Astronautics, Beijing, China, 1998.
- [28] K. Ito and R. Nakano, "Optimizing support vector regression hyperparameters based on cross-validation," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 2077–2082, Portland, OR, USA, July 2003.
- [29] J. T. Kwok and I. W. Tsang, "Linear dependency between epsilon and the input noise in epsilon-support vector regression," *IEEE Transactions on Neural Networks*, vol. 14, no. 3, pp. 544–553, 2003.
- [30] G. E. P. Box and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*, Holden Day, San Francisco, CA, USA, 2nd edition, 1976.