Hindawi
Complexity
Volume 2018, Article ID 5170281, 12 pages
https://doi.org/10.1155/2018/5170281

Research Article

Research on Decision-Making of Complex Venture Capital Based on Financial Big Data Platform

Tao Luo

School of Economics, Xihua University, Chengdu, China

Correspondence should be addressed to Tao Luo; luotao@mail.xhu.edu.cn

Received 12 July 2018; Revised 12 September 2018; Accepted 20 September 2018; Published 2 December 2018

Guest Editor: Zhihan Lv

The prediction of stock premium has always been a hot issue. By predicting stock premiums to provide a way for companies to respond to financial risk investments, companies can avoid investment failures. In this paper, under the financial big data platform, bootstrap resampling technology and long short-term memory (LSTM) are used to predict the value of the stock premium within 20 months. First, using the theme crawler, jsoup page parsing, Solr search, and Hadoop architecture to build a platform for financial big data. Secondly, based on the block bootstrap resampling technology, the existing data information is expanded to make full use of the existing data information. Then, based on the LSTM network, the stock premium in 20 months is predicted and compared with the values predicted by support vector machine regression (SVR), and the *SSE* and R-square average indicators are calculated, respectively. The calculation results show that the *SSE* value of LSTM is lower than SVR, and the R-square value of LSTM is higher than SVR, which means that the effect of LSTM prediction is better than SVR. Finally, based on the forecast results and evaluation indicators of the stock premium, we provide countermeasures for the company's financial risk investment.

1. Introduction

Nowadays, investment strategies based on deep learning to analyze the future of corporate finance have become a trend [1–4]. In the historical data, a certain corresponding mode between the independent variable and the dependent variable is obtained through deep learning. The existing independent variable can be input into the pattern to predict the numerical value of the dependent variable; since the historical data has big data, the characteristics of the traditional prediction algorithm is difficult to achieve. Considering the above two reasons, we proposed a real-time forecasting model of LSTM stock premium on the financial big data platform. By predicting the value of the stock premium, it can provide a countermeasure for the company's financial investment risk.

Financial data has the characteristics of big data in time series. To perform real-time predictive analysis on financial big data, it is necessary to use the financial big data platform [5, 6]. Literature [7] based on an in-depth study of the Tmall platform web page structure uses automated testing techniques to simulate the way people browse web pages and e-commerce platform search engines, effectively avoiding the limitations of anticrawling technology. The correct data rate for batch acquisition exceeds 96%, meeting the practical requirements of scientific and industrial data analysis. Literature [8] puts forward and realizes the platform of jsoup as the basic technology, including data crawling, information extraction, and data analysis of agricultural product big data from the Internet, in view of the problem that agricultural product information circulation is difficult to maintain balance between agricultural production and market. Literature [9] uses the open-source software Solr to build a high-speed search engine to achieve efficient search for a large number of heterogeneous and unstructured medical data. And based on this, a web search and viewing system was developed. Conclusion: Solr's search performance far exceeds that of traditional relational databases, and Solr has proven to be highly efficient. Literature [10] proposed a series of methods to optimize concurrent MapReduce. Optimize Hadoop group load modeling based on data-dependent dependencies and allocate appropriate resources for concurrent tasks. This method takes into account the negative interception of the Hadoop group for the first time for Hive/MapReduce optimization and achieves ideal results in practical experiments.

The prediction of financial data has always been a hot issue [11–14]. Business organizations accumulate a large amount of historical transaction information while conducting transactions, and corporate managers expect to analyze some of the available models from these data in order to discover business opportunities. Through trend analysis, you can even find out in advance some emerging opportunities. In the financial services industry, analysts can develop targeted software that analyzes time series data to find profitable trading patterns. After further verification, the operator can use these trading patterns to make actual transactions and thus make a profit.

The literature [15] used the Bayesian network to predict the average price of the NIKKEI stock and the price-earnings ratio of Toyota Motor Corporation's stock price. The results show that the correlation accuracy and the root mean square error are better than the traditional time series prediction algorithm. Literature [16] sought to use an adaptive neurofuzzy inference system to design a model to investigate the trend of stock prices of the "Iran KHODRO" company in Tehran Stock Exchange. Three independent variables were selected, including the dependent variable of trading volume, yield, closing price, and stock price volatility as the optimal model. The results of the study show that the trend of the stock price can be predicted with a lower error level. Literature [17] proposed a hybrid prediction model that uses multiple technical indicators to predict stock price trends. By selecting the popular indicators of the correlation matrix, the cumulative probability distribution method (CPDM), the minimum entropy principle method (MEPA), the rough set theory (RST) tool set, and the genetic algorithm (GA) are used to obtain better prediction accuracy and stock gains. The validity of the proposed model was verified using a six-year TAIEX as an experimental data set. The experimental results show that the proposed model is superior to the above two prediction models (RST and GAs) in accuracy, and the stock return evaluation shows that the proposed model produces higher profits than the above two models.

In this paper, we studied the issue of stock premiums, considering the complex combination of book-to-market ratio (b/m), net equity expansion (ntis), cross-sectional premium (csp), etc. Equity premium is determined by book-to-market ratio (b/m), net equity expansion (ntis), cross-sectional premium (csp), and other indicators. It can be seen that this is a very complicated risk investment decision problem. Under the financial big data platform created, we make full use of the existing information based on the block bootstrap method and use the LSTM method to predict the stock premium in real time, which provides a countermeasure for the company's financial risk investment. LSTM-based systems can also learn to control robots, analyze images, summarize documents, recognize videos and handwriting, run chat bots, predict diseases and click rates and stock markets, compose music, and much more.

2. Financial Big Data Platform

2.1. Characteristics of Financial Big Data. In the bond market, stock prices change over time. Therefore, for other big data, it has unique characteristics. These characteristics have a great impact on the subsequent data preprocessing and LSTM real-time prediction models, so the characteristics of financial big data are briefly described [18].

(1) Financial data is infinite. In the financial market, data is generated all the time

(2) Financial data is time series. The generation of financial data changes with the time series, so financial data is often processed in order

(3) Financial data is real time. Financial data is generated at each time node and the time at which the data arrives is often ordered

(4) Financial data is unpredictable. In financial markets, the price of stocks is the result of multiple factors

According to the characteristics of big data, combining the characteristics of the above financial data, financial data can be called financial big data.

2.2. Key Technologies of Financial Big Data Platform

(1) Theme crawler. The most basic working principle of the crawler: starting from the URL (universal resource locator) of one or several initial web pages, the URL is used as the initial node. In the process of crawling the web page, the new URL is continuously obtained from the current page until a certain stop condition of the system is satisfied. Web crawlers are divided into general crawlers and theme crawlers. The general crawler is suitable for large search engines. When you see a web page, it will be crawled. For example, Baidu's web crawler, while the theme crawler will focus on the crawler of a certain topic. The topic crawler mentioned in this article refers to the focus on the financial field, news, and other data

(2) jsoup web page parsing. Web page parsing refers to the process of parsing the DOM structure of a web page and obtaining web page nodes to obtain web page content. jsoup is an open-source Java-based HTML parser that directly parses a URL address and HTML text content. It can get the nodes of the document in a way similar to JQuery, which can be very convenient to parse the web page. jsoup supports reading a web page from a URL or stream and obtains the node information of the web page according to the DOM structure of the web page but generally does not use jsoup to directly read the web page from the URL because the network processing of jsoup is not powerful enough. In this article, we choose jsoup for parsing HTML pages

(3) Solr search. Solr is an open-source project from Apache and is an enterprise search engine based on

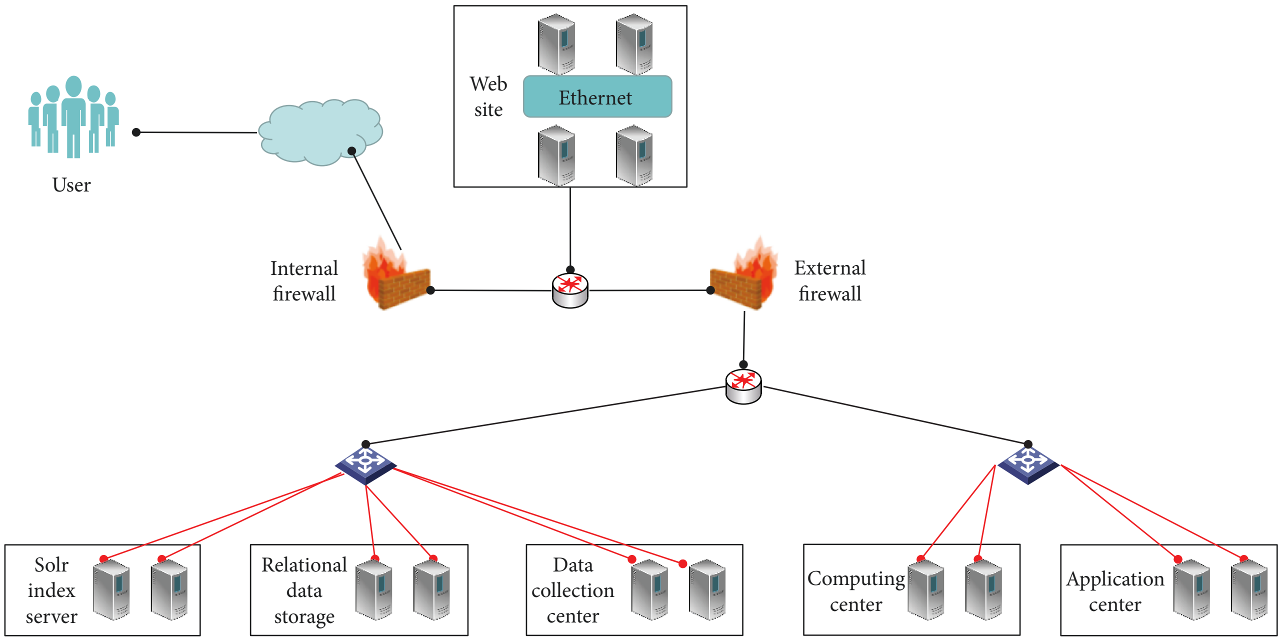

Figure 1: System network topology.

Lucene. Solr provides full-text search capabilities and supports highlighting search results and distributed indexes. Solr's output formats are XML format and JSON format. Solr has been applied on many large websites, mature and stable, and is a search engine solution that is very suitable for big data platforms

(4) Hadoop architecture. Hadoop is a distributed system infrastructure developed by the Apache Foundation. The significance of Hadoop is that users can distribute distributed programs without taking into account the underlying details of the distributed and make full use of the power of the cluster for high-speed computing and storage. The core part of Hadoop's framework is HDFS and MapReduce. HDFS provides storage for massive amounts of data, and MapReduce provides calculations for massive amounts of data. HDFS can be viewed as a hierarchical file system, but its architecture is built on a specific set of distributed nodes. These nodes include a primary node, the NameNode, and several child nodes, the DataNode. Files stored in HDFS are divided into blocks, which are then copied to multiple computers (DataNodes). This is very different from the traditional RAID architecture

Physically, application centers, computing centers, data collection centers, relational data stores, and index servers are all inside a local area network. The computer in the computing center deployed a Hadoop cluster for offline distributed computing. The application center's server provides services to the outside. The data collection center server is used for data collection, and the relational data storage server is used for relational database construction. Solr index server users build a distributed search engine. The overall network topology of the system is shown in Figure 1.

2.3. Construction of Financial Big Data Platform.

The architecture of the whole platform is introduced from a holistic perspective, and then the overall structure of each part is introduced from the perspective of each part [19, 20]. The system summary design section mainly explains the design ideas and concepts from the overall perspective, as well as the architecture of the system. The big data platform is divided into six parts: computing center, data center, collection center, dispatch center, application center, and open center.

The computing center includes four parts: streaming computing, distributed computing, task management, and computing module library. Stream computing refers to real-time streaming computing, which can provide real-time computing functions; distributed computing mainly refers to some distributed MapReduce calculations; task management provides management functions for managing cluster computing tasks; and the computational module library mainly encapsulates some computational algorithms that can be supplied to a variety of computational model libraries.

The data center is responsible for the storage, indexing, and management of data. Data indexing refers to indexing various data collected by the collection center, including unstructured financial data, crawled web page data, and some structured data. Data storage is stored in the form of file storage and relational data storage.

The collection center is responsible for the collection and integration of system data, including system integration acquisition, financial data collection, and some other data collection. System fusion acquisition refers to the integration of all structured and unstructured data of other units; financial data collection refers to the collection and processing of all financial related information; and other data collection refers to the collection of other data information, including information of personnel, etc.

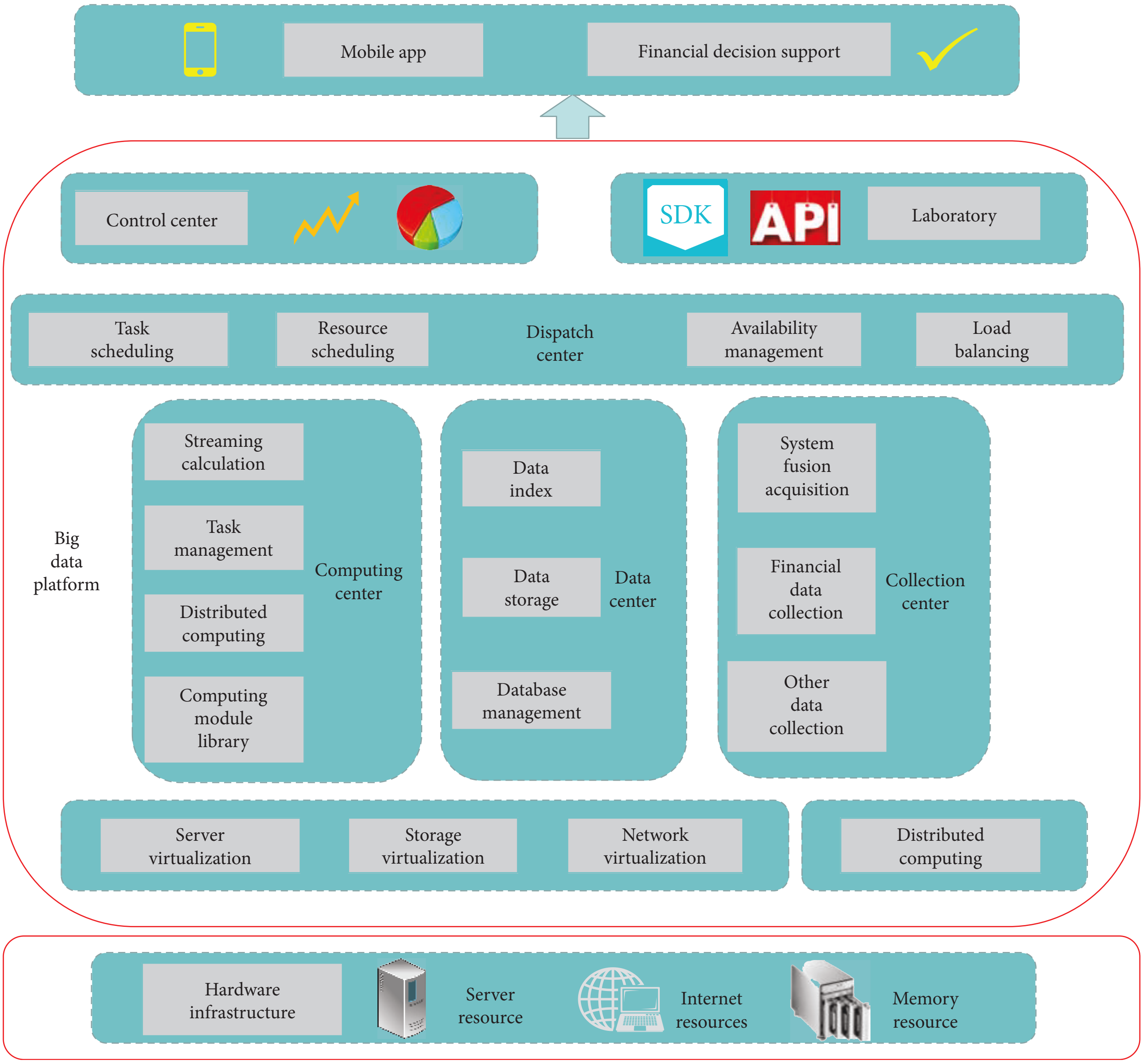

Figure 2: Overall structure view of the big data platform.

The dispatch center includes task management, resource scheduling, availability management, and load balancing. The open center is designed to provide data access interfaces to various organizational platforms outside the platform to support the application data sources of other organizations.

The application center provides some applications based on the data platform. Currently, the applications include data collection applications, financial product topics, and research reports. The overall structure view of the big data platform is shown in Figure 2.

3. Data and Preprocessing

3.1. Data Acquisition and Structure Analysis. We obtained a monthly equity premium for a company between 1947 and 2015, as well as a book-to-market ratio (b/m), a net equity expansion (ntis), cross-sectional premium (csp), and other independent variable data. Figure 3 is a three-dimensional view of the independent variables in the experimental data, which is drawn by MATLAB. Interpolation is used in the process of drawing to fit the discrete data [21]. In Figure 3, x and y are equally divided into 50 equal parts.

It can be seen from Figure 3 that there is a large fluctuation in the magnitude of the independent variable. Therefore, the data needs to be normalized when inputting training data. The standardization of data is to scale the data to fit into a small specific interval. In the paper, we normalize the data using min–max normalization based on the characteristics of the data. Normalization needs to be after block bootstrap operations. The first normalization is for the convenience of subsequent data processing, and the second is to ensure that the program runs faster when it converges. The processed data can reduce the training efficiency of the LSTM and reduce the time spent on training.

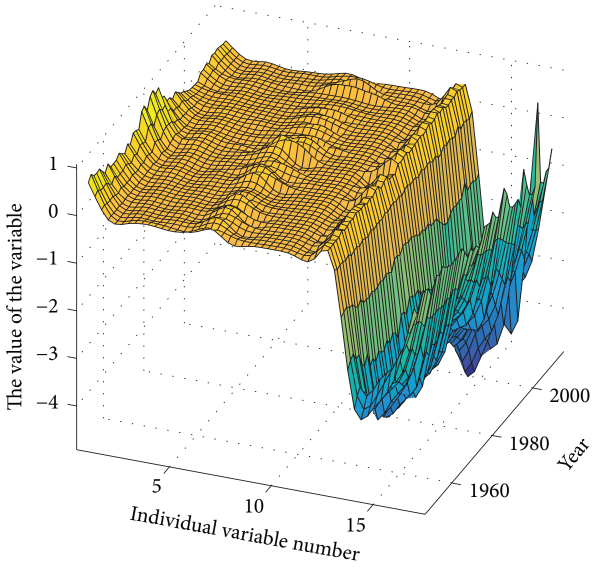

Figure 3: The 3D view of the independent variables in the experimental data.

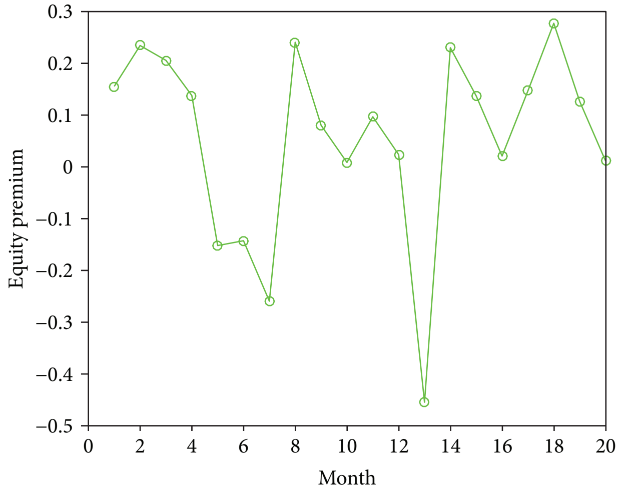

Figure 4: Schematic diagram of the equity premium that needs to be predicted.

Here, we need to predict the stock premium in December. Stock premium means that the issue price of a stock is higher than the face value of the stock, also known as stock premium issue. The issue premium of the stock represents the market's expectation of the stock and is a representative indicator of the company's development. By predicting the value of the stock premium, the company will provide corresponding countermeasures for the company's financial risk investment, thus helping the company to make a profit. Figure 4 is a schematic diagram of the 20-day equity premium that needs to be predicted.

3.2. Bootstrap Repeated Sampling. The bootstrap method is a statistical inference method that relies on only given observation information, without the need for other assumptions and the addition of new observations, by the American technologist Eflon [22, 23]. Bootstrap can use a small number of samples to simulate large samples for statistical estimation. It is based on the original data-based simulation sampling statistical inference method; the sample capacity is expanded by repeated sampling, so as to obtain the empirical distribution of the sample sequence statistics. It can be used to study a certain statistical distribution of a set of data, especially for interval estimates and hypothesis tests that are difficult to obtain parameters by conventional methods.

The basic principle of the bootstrap method is to consider a random sample sequence of length n from a completely undefined distribution P. Among them, X_{i} is the independent random sampling of the distribution, which is the basic requirement of the bootstrap method. The bootstrap process consists of the following steps:

(1) Let t_i denote the value of a particular sample statistic T. Extracting B random replacement samples with sample size n from the sample field, denoted by S_i (subscript i denotes the ith resampling)

(2) Let $S_i^* = \{X_1^*, X_2^*, \dots, X_i^*\}$ denote a simple random sample taken from the guide, which is a bootstrap sample

(3) For each subsample S_i^*, calculate its T value, denoted by $\{t_1, t_2, \dots, t_i\}$, respectively. The distribution of this T value is called the bootstrap empirical distribution

If B is large enough, an approximation is provided for all possible values of the statistic T by repeating the sampling therefrom. In this way, the small sample can be statistically simulated by the bootstrap method, and the statistical estimation of the known distribution and the known parameters can be obtained. The bootstrap implementation process is shown in Figure 5.

3.3. Data Preprocessing Based on Block Bootstrap. In many cases, statistical data cannot satisfy the nature of independent and identical distribution, but there are certain dependent structures, such as time series data. In this case, if you continue to apply independent and identically distributed bootstrap, it is likely to cause failure [24].

Because there is a dependency within the data, bootstrap cannot be performed on a single data point; otherwise, the dependent structure is completely destroyed. Then, in resampling, it must be ensured that the data of a whole "block" is extracted in the same unit, which is the origin of the "block." Here is a brief introduction to the bootstrap method of block structure [25, 26].

(1) First, the n data points are split into N blocks according to their order, and each block has a data length of l, that is, $B_i = \{X_1, \dots, X_i\}, \dots, B_i = \{X_N, \dots, X_n\}$, among them $N = n - l + 1$

(2) After the data is split, the resampled object becomes the N data blocks. For example, $n/1$ data blocks are extracted from them, and the resampled data with

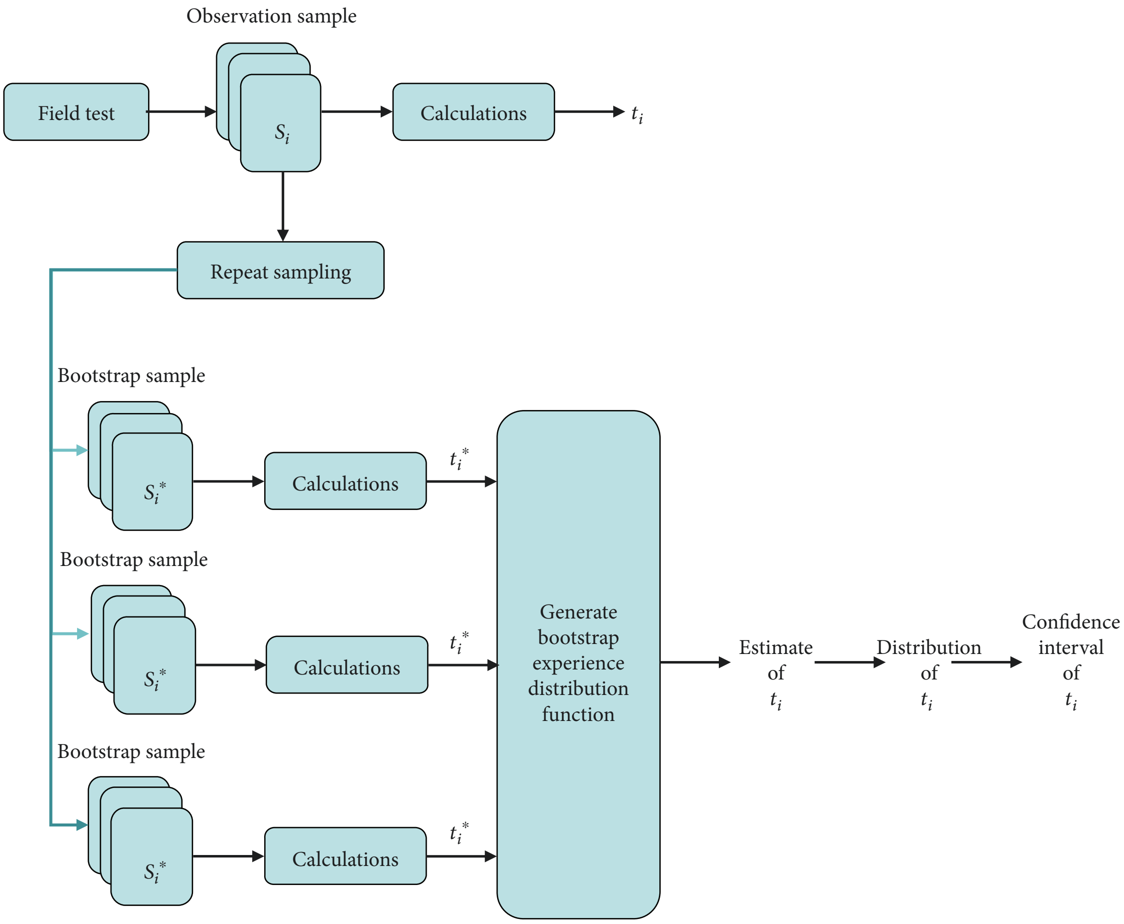

FIGURE 5: Bootstrap algorithm flow.

the sample size still can be formed by splicing together, supposing l can be divided into n blocks. For resampled data, the method described in Section 3.2 can still be used to calculate the corresponding estimated value and approximate statistical distribution

In order to clearly illustrate the operation of the block bootstrap, blocked bootstrap operations were performed on 0.07, 0.18, 0, 33, 049, 067, 071, 0.8, 1.01, 0.93, 09, 101, 099, 088, 082, 078, 063, 044, 026, 018, and 011. Figure 5 is a diagram of the results of the bootstrap operation.

It can be seen from the point in Figure 6 that such a batch of data has a certain internal structure, the data with a small sample number has an upward trend, and the data with a large serial number has a downward trend. If you use independent and identically distributed bootstrap, this dependency structure will be masked. The data blocks extracted in the figure overlap: the 5th, 6th, and 11th blocks are drawn, and the 5th block is drawn twice (not directly visible on the figure).

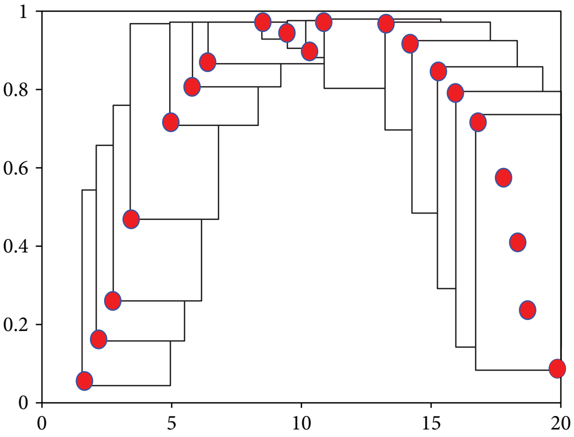

FIGURE 6: Bootstrap operation process diagram.

4. Research on Real-Time Forecasting Model of Equity Premium Based on LSTM

4.1. LSTM Internal Structure. Recurrent neural networks (RNN) are a very powerful computational model that can simulate almost any dynamic structure. However, gradient-based methods have important limitations. The magnitude of the error signal propagating in time depends on the amount of weight. Therefore, in the case where the hysteresis between input and output is greater than 5–10 discrete time steps, the standard RNN cannot be learned. This raises questions about whether standard RNNs have a significant advantage over time-window-based feedforward networks

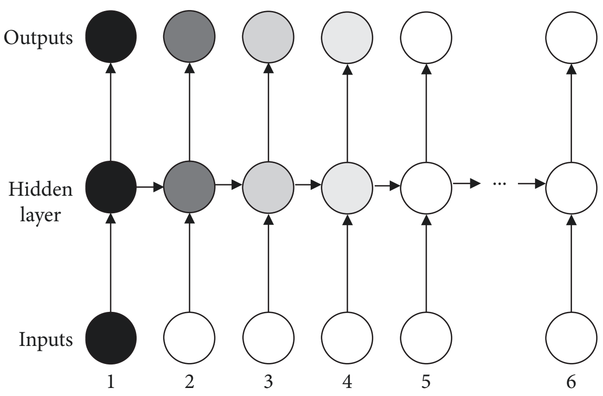

Figure 7: The vanishing gradient problem for RNNs.

[27, 28]. Figure 7 is a schematic diagram of the gradient disappearance problem of RNN.

In the opposite direction of time, 1 explains why the gradient of the RNN will disappear.

$$\begin{aligned}\delta_k^T &= \frac{\partial E}{\partial \mathrm{net}_k} = \frac{\partial E}{\partial \mathrm{net}_t}\frac{\partial \mathrm{net}_t}{\partial \mathrm{net}_k}\\ &= \frac{\partial E}{\partial \mathrm{net}_t}\frac{\partial \mathrm{net}_t}{\partial \mathrm{net}_{t-1}}\frac{\partial \mathrm{net}_{t-1}}{\partial \mathrm{net}_{t-2}}\cdots\frac{\partial \mathrm{net}_{k+1}}{\partial \mathrm{net}_k}\\ &= W \operatorname{diag}\left[f'(\mathrm{net}_{t-1})\right] W \operatorname{diag}\left[f'(\mathrm{net}_{t-2})\right]\cdots W \operatorname{diag}\\ &\quad\cdot \left[f'(\mathrm{net}_k)\right]\delta_t^l = \delta_t^T\prod_{i=k}^{t-1} W \operatorname{diag}\left[f'(\mathrm{net}_i)\right].\end{aligned} \tag{1}$$

The shading in Figure 7 indicates their sensitivity to the input at the first moment (the darker the shadow, the greater the sensitivity). As the new input hidden layer is activated, the network will gradually "forget" the first input, and the sensitivity will decay over time.

For general sequence modeling, LSTM has proven to be a special RNN structure in previous studies: stability and robustness for modeling remote dependencies. The main innovation of LSTM is memory blocks, which basically act as accumulators of state information. Each memory block contains one or more self-connected memory cells and three multiplying cells, an input gate, an output gate, and a forgetting gate, which primarily provide continuous write, read, and reset operations for the cell. The unit is accessed, written, and cleared by several self-parameterized control gates. Whenever there is a new input, if the input gate is activated, its information will be accumulated into the cell. In addition, if the forgotten gate is opened, the past unit state may be "forgotten" in the process. Otherwise, the latest unit output will propagate to the final state and be further controlled by the output gate. One advantage of using memory cells and gates to control the flow of information is that the gradient will be captured in the cell and prevented from disappearing too quickly, which is an important innovation for the RNN model [29, 30]. Figure 8 is an internal structure of a memory block.

The three gates are nonlinear summation units that are activated from inside and outside the block and control the activation of the unit by multiplication (small black circles). The input and output gates multiply the input and output of the unit, while forgetting the gate is multiplied by the previous state of the unit. There is no activation function applied in the cell. The gate activation function is typically a logical S shape, so gate activation is selected between 0 and 1. The unit input and output activation functions are usually tanh or logistic sigmoid. All other connections within the block are unweighted, and the only output from the block to the rest of the network comes from the output gate multiplication.

As previously mentioned, $W_{i,j}$ is the weight of the connection from unit i to unit j, the network input to unit j at time t is represented by a_j^t, and the activation of unit j at time t is b_j^t. In a neural network, forward propagation is generally performed to output the results of the model and to determine whether the results are satisfactory. Backward propagation can find the partial derivative of the error of the weight in the neural network, thereby adjusting the weight.

The LSTM equation is only given for a single memory block. For multiple blocks, you only have to repeat the calculations for each block in any order. The subscript t, φ, ω refers to the input gate, the forgetting gate, and the output gate of the block, respectively. The subscript c refers to one of the C memory units. The weights from unit c to input, forgetting, and output gates are represented as $W_{c,t}$, $W_{c,\varphi}$, and $W_{c,\omega}$, respectively. S_c^t is the state of the unit c at time t. f is the activation function of the gate and g and h are the unit input and output activation functions, respectively.

Let I be the number of inputs, K be the number of outputs, and H be the number of cells in the hidden layer. We use the index h to refer to the unit output of other blocks in the hidden layer, exactly the same as the standard hidden unit. As with the standard RNN, the forward pass of the length T input sequence x is calculated starting from $t = 1$, and the update equation is applied recursively as the t is incremented. The order in which the equations are calculated during forward and backward passes is important and should be performed as follows. As with standard RNN, all states and activations are initialized to zero at $t = 0$, and all δ terms are zero at $t = T + 1$.

The final weight derivative is obtained by summing the derivatives of each time step, where L is the loss function for training.

$$\delta_j^t \overset{\mathrm{def}}{=} \frac{\partial \ell}{\partial a_j^t}. \tag{2}$$

When passing forward, the mathematical expression of each structure inside the memory block:

$$\begin{aligned}\text{Input gate}: a_i^t &= \sum_{i=1}^{I} w_{il}x_i^t + \sum_{h=1}^{H} w_{hl}b_h^{t-1} + \sum_{c=1}^{C} w_{it}s_c^{t-1},\\ b_i^t &= f\left(a_i^t\right).\end{aligned} \tag{3}$$

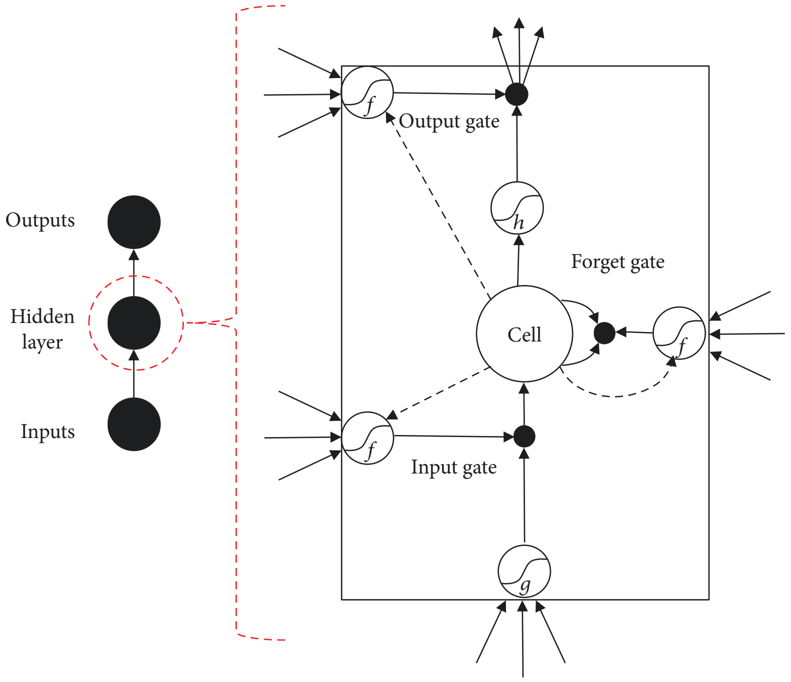

FIGURE 8: Internal structure of the memory block.

$$\text{Forgotten door}: a_{\phi}^{t} = \sum_{i=1}^{I} w_{i\phi} x_i^t + \sum_{h=1}^{H} w_{h\phi} b_h^{t-1} + \sum_{c=1}^{C} w_{i\phi} s_c^{t-1},$$
$$b_{\phi}^{t} = f\left(a_{\phi}^{t}\right). \tag{4}$$

$$\text{Cell}: a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t + \sum_{i=1}^{I} w_{hc} b_h^{t-1},$$
$$s_c^t = b_{\phi}^t s_c^{t-1} + b_i^t g\left(a_c^t\right). \tag{5}$$

$$\text{Output gate}: a_w^t = \sum_{i=1}^{I} w_{iw} x_i^t + \sum_{h=1}^{H} w_{hw} b_h^{t-1} + \sum_{c=1}^{C} w_{iw} s_c^{t-1},$$
$$b_w^t = f\left(a_w^t\right). \tag{6}$$

$$\text{Cell output}: b_c^t = b_w^t h\left(s_c^t\right). \tag{7}$$

Mathematical expression of each structure inside the memory block when passing backwards:

$$\delta_c^t \overset{\text{def}}{=} \frac{\partial \ell}{\partial b_c^t},$$
$$\delta_s^t \overset{\text{def}}{=} \frac{\partial \ell}{\partial s_c^t}. \tag{8}$$

$$\text{Cell output}: \varepsilon_c^t = \sum_{k=1}^{K} w_{ck} \delta_k^t + \sum_{h=1}^{H} w_{ch} \delta_h^{t+1}. \tag{9}$$

$$\text{Output gate}: \delta_w^t = f'\left(a_w^t\right) \sum_{c=1}^{C} h\left(s_c^t\right) \varepsilon_c^t. \tag{10}$$

$$\text{Status}: \varepsilon_s^t = b_w^t h'\left(s_c^t\right) \varepsilon_c^t + b_{\phi}^{t+1} \varepsilon_c^{t+1} + w_{cl} \delta_l^{t+1}$$
$$+ w_{c\phi} \delta_{\phi}^{t+1} + w_{cw} \delta_w^t. \tag{11}$$

$$\text{Cell}: \delta_c^t = b_l^t g'\left(a_c^t\right) \varepsilon_s^t. \tag{12}$$

$$\text{Forgotten door}: \delta_{\phi}^t = f'\left(a_{\phi}^t\right) \sum_{c=1}^{C} s a_c^{t-1} \varepsilon_s^t. \tag{13}$$

$$\text{Input gate}: \delta_l^t = f'\left(a_l^t\right) \sum_{c=1}^{C} g\left(a_c^t\right) \varepsilon_s^t. \tag{14}$$

4.2. LSTM Overall Design. Based on the aforementioned test data characteristics and the theory of LSTM, we designed the network structure required in this paper. For clarity, we analyze the two-layer forward network. Figure 9 is a schematic diagram of a two-layer network LSTM structure.

In this figure, we can see that the input values are transmitted through the input gates to the gate, input and output activation ribbons in the first layer network. After the arithmetic processing of the memory block, the values are output to the gate, input and output activation functional areas of the first and second layers. At the same time, we also observe that the initial input value is output to the second level of the gate, input and output activation ribbons. Finally, the output values of the first and second layers are output to the network. A network larger than two layers can be modeled on a two-layer network. Since the number of arguments in this paper is 20 and there is only one output variable, n in Figure 9 is 20.

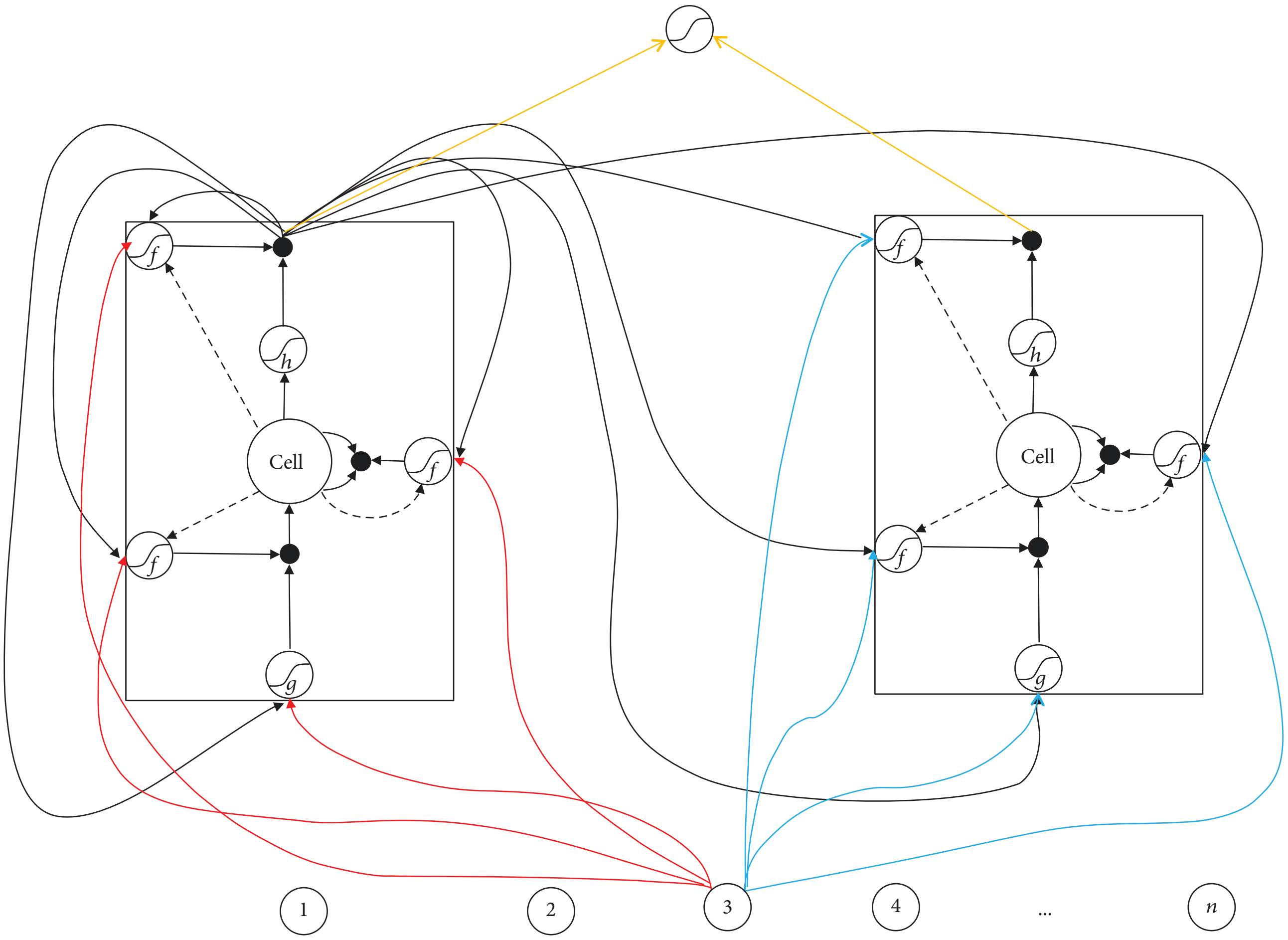

Figure 9: Schematic diagram of two-layer network LSTM structure.

5. Result Analysis and Investment Strategies

5.1. Analysis and Evaluation of Prediction Results. Based on the above analysis, we designed a 50-layer network structure with a step size of 10, an activation function of tanh, a loss function of mse, and a number of learning samples of seven. In order to improve the efficiency of learning, we sacrificed memory to speed up the learning, that is, set batch_size = 72. The values of the various parameters are set as follows: neuralNums = 50, timeStep = 10, dropout = 0.1, loss = "mse," optimizer = "adam," activation = "tanh," output_dim = 1, epochs = 7, and batch_size = 72. Using the grid search method, when neuralNums = 50, epochs = 7, and dropout = 0.1, the prediction effect of LSTM is optimal. In the LSTM network structure, the first layer does not use the activation function, and the other network layer uses the tanh function as the activation function. In this paper, we also use support vector machine regression (SVR) [31–34] and LSTM to predict the stock premium. Figure 10 is a comparison of the SVR and LSTM prediction results.

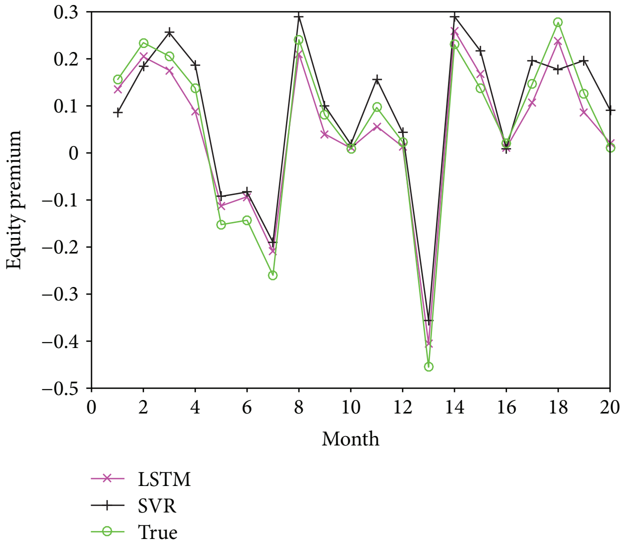

Figure 10: Comparison of SVR and LSTM prediction results.

We use the monthly independent variables and stock premiums from 1947 to 2005 as input samples and put them into the LSTM network for learning. Then, we input the 20-month independent variable into the trained model and get the value of the predicted stock premium. By analyzing Figure 9, it can be seen that, overall, the prediction effect of LSTM is better than that of SVR, and it is stable. In order to more clearly describe the error between the predicted

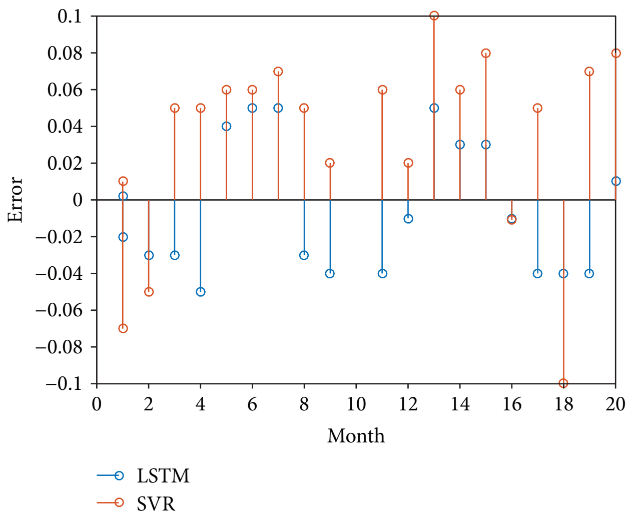

Figure 11: Comparison of prediction errors between SVR and LSTM.

and actual values, we plot the error maps predicted by the two methods, as shown in Figure 11.

RNN does not have long-term and short-term memory characteristics, which has been demonstrated in the existing literature. Therefore, RNN is not applicable for comparison in this paper. As can be seen from Figure 11, the absolute value of the prediction error of the LSTM is smaller than the absolute value of the error predicted by the SVR, that is, its accuracy is high, which also explains the stability of the prediction from the side. In the actual forecast, we predict multiple time nodes in the future instead of one time node in the future. Because predicting multiple time nodes can see a trend in the future, this is more sensible than predicting a time node. Therefore, the number of time steps set in this paper is 10. In order to better evaluate the pros and cons of the prediction, we calculated the size of the SSE, that is, the sum of the squares of the errors of the predicted data and the corresponding points of the original data. The closer the *SSE* is to 0, the better the model selection and fit, and the more successful the data prediction. The *SSE* is defined as follows:

$$\mathrm{SSE} = \sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2. \tag{15}$$

At the same time, the size of the R-square is also calculated, that is, the sum of the squares of the prediction data and the mean difference. It can be seen from the above expression that the normal range of the R-square is [0 1], and the closer to 1, the stronger the explanatory power of the model. Its definition is as follows:

$$\mathrm{SST} = \sum_{i=1}^{n} \left(y_i = \bar{y}_i\right)^2, \tag{16}$$

$$\text{R-square} = 1 - \frac{\mathrm{SSE}}{\mathrm{SST}}. \tag{17}$$

Table 1: Evaluation index calculation result.

	SSE	R-square
LSTM	0.024804	0.953882
SVR	0.075421	0.889393

The calculation results of the evaluation indicators are shown in Table 1.

It can be seen from Table 1 that the SSE value of the LSTM prediction result is smaller than the SSE value of the LSTM prediction result in the SSE evaluation index, indicating that the LSTM prediction effect is better than the SVR; it is also known that the LSTM prediction is in the R-square evaluation index. The R-square value of the result is larger than the R-square value of the LSTM prediction result, indicating that the LSTM prediction is better than the SVR. As we all know, the LSTM algorithm has many parameters to determine. In theory, particle swarm optimization algorithm and ant colony algorithm can be used to automatically find the optimal parameters. However, if the particle swarm algorithm is used, although the optimal parameters can be found automatically, the parameters of the particle swarm optimization algorithm need to be adjusted. This makes the results of the algorithm more uncontrollable. In the future, we can improve the performance of LSTM by using a new activation function, enhancing the link of cell to each gate.

5.2. Risk Investment Advice. It can be seen from the analysis of the results that the prediction effect of LSTM can be better than that of SVR. Based on this, we can design an operating software based on LSTM stock premium real-time forecasting under the financial big data platform. On the other hand, we can advise financial investments based on indicators. When both SSE and R-square are smaller than a certain reference value, we need to sell the stock; when both SSE and R-square are greater than a certain reference value, we can continue to buy the company's stock; when any indicator is less than a certain reference value, we do not need to buy or sell. The reference value can be determined based on the analysis of historical data and the corresponding decision. Determine whether you need to hold a share or sell off by predicting the value and the size of the evaluation indicator. To improve the stability of the forecast, we can get more credible results by increasing the month of the predicted stock premium.

6. Conclusion

First, using the crawler technology, jsoup page analysis technology, Solr search technology, and Hadoop architecture to build a platform for financial big data. Then, based on the block bootstrap resampling technique, the existing data information is utilized to the maximum extent. Finally, based on the LSTM network, the stock premium in 20 months is predicted and compared with the values predicted by support vector machine regression (SVR), and the respective *SSE* and R-square average indicators are calculated. The calculation results show that the *SSE* value of LSTM is 0.024804, and the *SSE* value of SVR is 0.075421; the R-square value of LSTM is 0.953882, and the R-square value of SVR is 0.889393, which means that the effect of LSTM prediction is better than SVR.

Data Availability

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

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The study was supported by the humanities and social science fund of the Ministry of Education of the People's Republic of China: research on the incentive mechanism of economic development of China's model—based on the perspective of stochastic dynamic optimization (Grant No. 15XJC630005).

References

[1] J. Wang, R. Hou, C. Wang, and L. Shen, "Improved v, support vector regression model based on variable selection and brain storm optimization for stock price forecasting," *Applied Soft Computing*, vol. 49, pp. 164–178, 2016.

[2] Z. Fang, G. Luo, F. Fei, and S. Li, "Stock forecast method based on wavelet modulus maxima and Kalman filter," in *2010 International Conference on Management of e-Commerce and e-Government*, pp. 50–53, Chengdu, China, October 2010.

[3] A. M. Yang, X. L. Yang, J. C. Chang, B. Bai, F. B. Kong, and Q. B. Ran, "Research on a fusion scheme of cellular network and wireless sensor for cyber physical social systems," *IEEE Access*, vol. 6, no. 99, pp. 18786–18794, 2018.

[4] S. Aggarwal and S. Aggarwal, "Deep investment in financial markets using deep learning models," *International Journal of Computer Applications*, vol. 162, no. 2, pp. 40–43, 2017.

[5] C. Jiang, Z. Ding, J. Wang, and C. Yan, "Big data resource service platform for the internet financial industry," *Chinese Science Bulletin*, vol. 59, no. 35, pp. 5051–5058, 2014.

[6] J. Woodard, "Big data and Ag-Analytics: an open source, open data platform for agricultural & environmental finance, insurance, and risk," *Agricultural Finance Review*, vol. 76, no. 1, pp. 15–26, 2016.

[7] W. B. Cao and K. J. Zhang, "Research on anti-reptile technology based on automated testing-taking Tmall platform as an example," *Modern Computer*, vol. 11, no. 4, pp. 10–14, 2018.

[8] J. Wang, S. Yang, Y. Wang, and C. Han, "The crawling and analysis of agricultural products big data based on jsoup," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 1197–1202, Xi'an, China, August 2015.

[9] B. Zhou, S. Xue, and G. U. Guang-Li, "Application of Solr in medical big data searching," *China Digital Medicine*, vol. 25, no. 4, pp. 48–53, 2016.

[10] A. Cherniak, H. Zaidi, and V. Zadorozhny, "Optimization strategies for A/B testing on HADOOP," *Proceedings of the VLDB Endowment*, vol. 6, no. 11, pp. 973–984, 2013.

[11] X. Li and J. Feng, "Accurate demand forecasting of financial data based on big data analysis," *Boletin Tecnico/Technical Bulletin*, vol. 55, no. 9, pp. 148–153, 2017.

[12] M. Marcek, J. Petrucha, and D. Marcek, "Statistical and NN forecasting models of financial data: making inferences about the accuracy and risk evaluation," *Journal of Multiple-Valued Logic & Soft Computing*, vol. 17, no. 4, pp. 321–337, 2011.

[13] R. K. Lai, C.-Y. Fan, W.-H. Huang, and P.-C. Chang, "Evolving and clustering fuzzy decision tree for financial time series data forecasting," *Expert Systems with Applications*, vol. 36, no. 2, Part 2, pp. 3761–3773, 2009.

[14] S. Walczak, "An empirical analysis of data requirements for financial forecasting with neural networks," *Journal of Management Information Systems*, vol. 17, no. 4, pp. 203–222, 2015.

[15] Y. Zuo and E. Kita, "Stock price forecast using Bayesian network," *Expert Systems with Applications*, vol. 39, no. 8, pp. 6729–6737, 2012.

[16] E. Abbasi and A. Abouec, "Stock price forecast by using neuro-fuzzy inference system," *Proceedings of World Academy of Science, Engineering and Technology*, vol. 36, pp. 320–323, 2008.

[17] C. H. Cheng, T. L. Chen, and L. Y. Wei, "A hybrid model based on rough sets theory and genetic algorithms for stock price forecasting," *Information Sciences*, vol. 180, no. 9, pp. 1610–1629, 2010.

[18] N. Kshetri, *Big data's role in expanding access to financial services in China*, Elsevier Science Publishers B. V., 2016.

[19] B. Cheng, S. Longo, F. Cirillo, M. Bauer, and E. Kovacs, "Building a big data platform for smart cities: experience and lessons from Santander," in *2015 IEEE International Congress on Big Data*, pp. 592–599, Dalian, China, June 2015.

[20] Z. Wu, J. Wu, M. Khabsa et al., "Towards building a scholarly big data platform: challenges, lessons and opportunities," in *IEEE/ACM Joint Conference on Digital Libraries*, pp. 117–126, London, UK, September 2014.

[21] H. Akima, "A method of bivariate interpolation and smooth surface fitting for irregularly distributed data points," *ACM Transactions on Mathematical Software*, vol. 4, no. 2, pp. 148–159, 1978.

[22] B. Efron, "The jackknife, the bootstrap and other resampling plans," *Siam Monograph*, vol. 38, no. 384, 1982.

[23] B. Efron, "Bootstrap methods: another look at the jackknife," in *Breakthroughs in Statistics*, Springer Series in Statistics, S. Kotz and N. L. Johnson, Eds., pp. 569–593, Springer, New York, NY, 1992.

[24] K. Singh, "On the asymptotic accuracy of Efron's bootstrap," *The Annals of Statistics*, vol. 9, no. 6, pp. 1187–1195, 1981.

[25] S. N. Lahiri, "Theoretical comparisons of block bootstrap methods," *Annals of Statistics*, vol. 27, no. 1, pp. 386–404, 1999.

[26] E. Paparoditis and D. N. Politis, "Tapered block bootstrap," *Biometrika*, vol. 88, no. 4, pp. 1105–1119, 2001.

[27] A. G F, J. Schmidhuber, and F. Cummins, *Learning to forget: Continual prediction with LSTM*, Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, 1999.

[28] S. H. I. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM network: a machine learning approach for precipitation nowcasting," *Advances in Neural Information Processing Systems*, vol. 04214, pp. 802–810, 2015.

[29] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, “LSTM: a search space odyssey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[30] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: continual prediction with LSTM,” *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[31] B. Kichonge, G. R. John, T. Tesha, and I. S. N. Mkilaha, “Prediction of Tanzanian energy demand using support vector machine for regression (SVR),” *International Journal of Computer Applications*, vol. 109, no. 3, pp. 34–39, 2015.

[32] R. M. Balabin and E. I. Lomakina, “Support vector machine regression (SVR/LS-SVM)–an alternative to neural networks (ANN) for analytical chemistry comparison of nonlinear methods on near infrared (NIR) spectroscopy data,” *Analyst*, vol. 136, no. 8, pp. 1703–1712, 2011.

[33] S. Chen, K. Jeong, and W. K. Härdle, “Recurrent support vector regression for a non-linear ARMA model with applications to forecasting financial returns,” *Computational Statistics*, vol. 30, no. 3, pp. 821–843, 2015.

[34] M. Wauters and M. Vanhoucke, “Support vector machine regression for project control forecasting,” *Automation in Construction*, vol. 47, pp. 92–106, 2014.