Hindawi
Scientific Programming
Volume 2021, Article ID 4055281, 8 pages
https://doi.org/10.1155/2021/4055281

Research Article

Stock Prediction Based on Optimized LSTM and GRU Models

Ya Gao,[1] **Rong Wang**,[2] **and Enmin Zhou**[3]

[1]*School of Public Finance and Taxation, Central University of Finance and Economics, Beijing, China*
[2]*School of Computer Science and Technology, Xidian University, Xi'an, China*
[3]*School of Electronics and Information, Xi'an Jiaotong University, Xi'an, China*

Correspondence should be addressed to Ya Gao; 18737165379@163.com

Received 9 August 2021; Accepted 11 September 2021; Published 29 September 2021

Academic Editor: Punit Gupta

Stock market prediction has always been an important research topic in the financial field. In the past, inventors used traditional analysis methods such as K-line diagrams to predict stock trends, but with the progress of science and technology and the development of market economy, the price trend of a stock is disturbed by various factors. The traditional analysis method is far from being able to resolve the stock price fluctuations in the hidden important information. So, the prediction accuracy is greatly reduced. In this paper, we design a new model for optimizing stock forecasting. We incorporate a range of technical indicators, including investor sentiment indicators and financial data, and perform dimension reduction on the many influencing factors of the retrieved stock price using depth learning LASSO and PCA approaches. In addition, a comparison of the performances of LSTM and GRU for stock market forecasting under various parameters was performed. Our experiments show that (1) both LSTM and GRU models can predict stock prices efficiently, not one better than the other, and (2) for the two different dimension reduction methods, both the two neural models using LASSO reflect better prediction ability than the models using PCA.

1. Introduction

The financial market is quite volatile and experiences periods of contraction as well as expansion. The stock market, as a major financial market, is likewise highly volatile. The stock market has the characteristics of high return which has attracted the majority of investors and high risk which puts pressure on investors to sell out at the wrong time. In order to reduce unnecessary losses and obtain higher trading profits, the investors usually except to predict the stock price trend. As a result, stock market forecasting has been a major research topic in the financial area and attracts the attention of investors. In the stock market, the factors affecting the rise and fall of stock prices are complex and diverse. It includes not only the impact of economic factors such as price indicator, circulation indicator, activity degree, and economic uncertainty but also the impact of noneconomic factors such as traders' expectations, traders' psychological factors, and political environment. Therefore, the prediction of stock price has always been a challenging task.

According to the efficient market hypothesis [1], the stock price can be predicted according to the data of historical stocks. Furthermore, in recent years, since the increasing computing power and the decreasing data storage costs, especially the rise and development of innovative technologies such as big data, machine learning, reinforcement learning, and other optimization technologies, researchers have developed various models for predicting stock prices. Machine learning has been widely used in the capital market and plays an indispensable role in predicting future stock prices based on historical data. Traditional stock price forecasting models are mainly linear models, including autoregressive integrated moving average (ARIMA) model [2], multiple linear regression model, and exponential smoothing model [3, 4]. However, those (autoregressive integrated moving average,multiple linear regression model, and exponential smoothing model) linear models play an important role in promoting the progress and development of stock forecasting. Stock prices are typically noisy, fluctuating, and nonparametric, resulting in nonlinear and nonstationary characteristics in the stock market. The standard linear prediction model is unable to produce reliable stock predictions. With the development of deep learning methods, nonlinear neural networks are increasingly employed to predict the stock price for their higher accuracy.

The artificial neural network (ANN) includes MP neural network and back propagation (BP) neural network. However, the structure of ANN model is too single and there are some problems: (1) over fitting leads to the weak ability of the model generalization, (2) local extrema leads to the decline of the prediction ability of the model, and (3) the gradient disappears or explodes due to the excessive weight of neurons in the optimization process, resulting in the failure of prediction. Therefore, relevant scholars introduce deep neural networks (DNN), including convolutional neural network (CNN), recurrent neural network (RNN), long-term and short-term memory neural network (LSTM), and gated recurrent neural network (GRU), to improve the problems existing in the ANN model, so as to improve the accuracy and efficiency of prediction.

CNN is a type of neural network that has been increasingly popular in recent years. A one-dimensional CNN is a neural network that is designed to analyse image data efficiently. CNN can read and automatically extract the most significant features from the original input data for learning. This method feeds the network observed time series value as input and uses a multilayer network to predict the unobserved value. For example, Xu et al. [5] employed CNN to extract important stock features from stock market returns for forecasting stock market trends. Recurrent neural networks (RNN) such as long-term and short-term memory neural networks (LSTM) are another tool for predicting time series [6, 7]. LSTM accurately estimates time series data by using both the historical and the present stock data. In recent years, LSTM has been applied to stock market forecasting in different stock markets around the world. Chen et al. [8] used an LSTM model to predict China's Shanghai and Shenzhen stock markets. Li et al. [9] introduced the stock indicator with investor sentiment based on the LSTM model to predict the CS1300 index value, and the research results showed that the model was better than the support vector machine method in prediction accuracy. However, this model does not reduce the dimension of stock indicator. Jiawei and Murata [10] attempted to identify the influencing factors of stock market trend prediction through the LSTM model, which used a preprocessing algorithm to reduce the dimension of stock features and a sentiment analyzer to present financial news for stock trend prediction. However, only one dimension reduction method is used, and there is no comparison with other methods. Hu [11] reduced the dimension of stock technical analysis indicators by PCA and LASSO methods before using the LSTM model to predict. The results demonstrated that compared with the LASSO-LSTM model, the PCA-LSTM model can significantly reduce data redundancy and enhance prediction accuracy. Although this work used different dimension reduction methods, it only used one model and did not compare with other models.

Cho et al. [12] reduced the LSTM structure and created GRU, a new deep learning architecture that integrates long-term and short-term memory. GRU solves the problem of gradient disappearance and explosion in classic recurrent neural networks (RNNs) when learning long-term reliance. GRU has also been widely used in recent stock forecasting. Shen et al. [13] compared and predicted the trading signals of stock indicator based on the GRU model and SVM. The results demonstrated that the prediction accuracy of the two GRU models is higher than that of other models. However, the emotion indicator was not included in this study. Rahman et al. [14] used the stock data of Yahoo Finance mobile phone and GRU model to predict the stock price. The emotional indicators were not considered in this study, nor were compared with the performance of other models [15].

In this paper, we integrate a variety of technical indicators, such as investor sentiment indicators and financial data based on the Shanghai Composite Index data. We use LASSO and PCA methods to perform dimension reduction on the multiple influencing factors of the extracted stock price. The LSTM and GRU models are then utilized in this paper to forecast the stock price. Most importantly, by comparing the accuracy and stability of the LASSO-LSTM, LASSO-GRU, PCA-LSTM, and PCA-GRU models, the optimal forecasting model may be recommended.

2. Methodology

2.1. LASSO. In empirical analysis, in order to minimize the model deviation due to the lack of important independent variables, we set multidimensional variables. The models need to find the set of independent variables with the strongest explanatory power to the dependent variables. That is, the models need to improve the interpretability and prediction accuracy through independent variable selection (indicator selection and field selection). Indicator selection is an extremely important problem in statistical modelling. LASSO is an estimation method that can simplify the indicator set. It is a compressed estimation. It gets a more refined model by constructing a penalty function, which makes it compress some coefficients and set some coefficients to zero. Therefore, it retains the advantage of subset contraction and is a biased estimation for dealing with complex collinear data.

LASSO's basic idea is to minimize the sum of squares of residuals under the constraint that the sum of absolute values of regression coefficients is less than a constant, so as to produce some regression coefficients strictly equal to 0 and obtain an interpretable model. LASSO adds penalty term to the ordinary linear regression model, and the LASSO estimation of the ordinary linear model is

$$\begin{aligned} \widehat{\beta_{\text{Lasso}}} &= \arg\min_{\beta \varepsilon R^d} \|Y - X\beta\|^2, \\ s.t. &\sum_{j=1}^{d} \left|\beta_j\right| \leq t, t \geq 0, \end{aligned} \tag{1}$$

which is equivalent to

$$\widehat{\beta_{\text{Lasso}}} = \arg\min_{\beta \varepsilon R^d} \left(\|Y - X\beta\|^2 + \lambda \sum_{j=1}^{d} \left|\beta_j\right| \right), \tag{2}$$

where t and λ are said to be in one-to-one correspondence and they are the adjustment coefficients.

Let $t_0 = \sum_{j=1}^{d} |\widehat{\beta}_j(OLS)|$, and when $t < t_0$, some coefficients will be compressed to 0, so as to reduce the dimension of X and the complexity of the model. Finally, the variable selection can be realized by controlling the adjustment coefficient through the λ.

2.2. PCA. Principal component analysis (PCA) is a dimension reduction statistical method. With the help of an orthogonal transformation, it transforms the original random vector whose components are related into a new random vector whose components are not related. This is expressed algebraically as transforming the covariance matrix of the original random vector into a diagonal matrix and geometrically as transforming the original coordinate system into a new orthogonal coordinate system. Then, the multidimensional variable system is reduced, so that it can be transformed into low-dimensional variable system with a high accuracy, and the low-dimensional system can be further transformed into one-dimensional system by constructing an appropriate value function.

(1) Standardized collection of original indicator data p-dimensional random vector $x = (x_1, x_2, x_3, \ldots \ldots, x_p)^T$ and n samples $x_i = (x_{i1}, x_{i2}, x_{i3}, \ldots\ldots, x_{ip})^T$, where $i = 1, 2, \ldots, n (n > p)$. Then, we construct the sample array and carry out the following standardized transformation on the sample array elements: $Z_{ij} = x_{ij} - \overline{x_j}/s_j$, where $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, p$, $\overline{x_j} = \sum_{i=1}^{n} x_{ij}/n$, and $S_j^2 = \sum_{i=n}^{n} (x_{ij} - \overline{x_j})^2/n - 1$. Thus, the standardized matrix Z is obtained.

(2) Find the correlation coefficient matrix for the standardized matrix Z as

$$R = \left[r_{ij}\right]_{p\times p} = \frac{Z^T Z}{n-1}, \tag{3}$$

where $r_{ij} = \sum z_{kj}.z_{kj}/n - 1$; i, $j = 1, 2, \ldots, p$.

(3) Solve the characteristic equation of sample correlation matrix R by $|R - \lambda I_p| = 0$ to get p-characteristic roots, thus determining the principal component. Determine the value of m according to $\sum_{j=1}^{m} \lambda_j / \sum_{j=1}^{p} \lambda_j \geq 0.85$, so that the utilization rate of information can reach more than 85%. For each λ_j, $j = 1, 2, \ldots, m$, we solve the equation $Rb = \lambda_j b$ to obtain the unit eigenvector b_j^o.

(4) Convert the standardized indicator variable into the main component $U_{ij} = z_i^T b_j^o$, $j = 1, 2, \ldots, m$, where U_1 is called the first principal component, U_2 is called the second principal component, and so on. U_p is called the p principal component.

(5) Evaluate m principal components comprehensively. The final evaluation value is obtained by weighted sum of m principal components, and the weight is the variance contribution rate of each principal component.

2.3. LSTM and GRU. LSTM is a special type of recurrent neural network (RNN). The RNN neural network model can recycle the weight parameters of neurons and can effectively employ past data information for prediction. However, RNN can only deal with certain short-term dependence and is prone to gradient explosion and gradient disappearance, that is, long-term dependence on historical data. In order to solve these problems, LSTM was proposed by Hochreiter and Schmidhuber [6] and then improved and promoted by Graves [16]. It has been widely used in a variety of challenges and has yielded impressive outcomes.

Compared with the RNN model, the LSTM model introduces a cell state (C_t) and uses the input gate (i_t), forget gate (f_t), and output gate (O_t). The three gates are used to maintain and control information. At time t, x_t is the input data, h_t represents the current output, c_t is the value from the input gate, tan$h$ is hyperbolic tangent function, σ is the sigmoid function, W represents the matrix weight, and b is the bias. The operation formula of LSTM is as follows.

Forget gate:

$$f_t = \sigma\left(W_f * \left[h_{t-1}, x_t\right] + b_f\right). \tag{4}$$

Input gate:

$$\begin{aligned} i_t &= \sigma\left(W_i * \left[h_{t-1}, x_t\right] + b_i\right), \\ \widetilde{c}_t &= \tanh\left(W_c * \left[h_{t-1}, x_t\right] + b_c\right), \\ c_t &= f_t * c_{t-1} + i_t * \widetilde{c}_t. \end{aligned} \tag{5}$$

Output gate:

$$\begin{aligned} o_t &= \sigma\left(W_o * \left[h_{t-1}, x_t\right] + b_o\right), \\ h_t &= o_t * \tanh\left(c_t\right). \end{aligned} \tag{6}$$

The LSTM model is especially popular in the field of financial forecasting because it effectively deals with the redundancy of relevant information in historical data.

GRU is one of the variants of RNN which is introduced by Cho et al. [12]. By introducing gating structure, it solves the problem that RNN is difficult to deal with long-distance information acquisition. Compared with LSTM, GRU is simplified and only update gate (z_t) and reset gate (r_t) are introduced. In GRU, the update (or input) gate decides how much input (x_t) and previous output (h_{t-1}) to be passed to the next cell and the reset gate is used to determine how much of the past information to forget. The current memory content ensures that only the relevant information needs to be passed to the next iteration, which is determined by the weight W. The main operations in GRU are governed by the following formulae.

Update gate:

$$z_t = \sigma\left(W_z * \left[h_{t-1}, x_t\right]\right). \tag{7}$$

Reset gate:

$$r_t = \sigma\left(W_r * \left[h_{t-1}, x_t\right]\right). \tag{8}$$

After resetting the gate and updating the gate, the candidate status value of GRU unit is $\widetilde{h}_t$ and the final output status value is h_t:

$$\widetilde{h}_t = \tanh\left(W_{\widetilde{h}} * \left[r_t * h_{t-1}, x_t\right]\right),$$
$$h_t = \left(1 - z_t\right) * h_{t-1} + z_t * \widetilde{h}_t. \tag{9}$$

3. Experiment Settings and Results

3.1. Data Source and Indicator Selection. In this paper, the data of the Shanghai Composite Index (000001) from April 11, 2007, to August 3, 2021, are selected as the experimental data. The data comes from NetEase Finance and Economics website, with a total of 3,481 days. In order to evaluate the training effect of the model, we divide the experimental data into training set and test set, of which 80% are used as one training set to train the stock prediction model and the other 20% are used as test sets to verify the prediction effect of the model. In addition, we use Intel Core i9-9900K CPU with memory 64 GB to finish the experiments.

In the selection process of stock technical indicators, this paper considers the factors affecting the stock price as much as possible. Compared with other studies, this paper selects the open price, highest price, lowest price, trading volume, and other common technical indicators, such as OBV, KDJ, BIAS, RSI, CCI, and MFI, as well as other stock price judgment technical indicators and PSY indicators reflecting investors' psychological mood. These indicators comprehensively reflect the information affecting stock price fluctuations and have the strong explanatory power for stock price fluctuations. The selected indicators are described in detail in Table 1.

3.2. Experimental Setup. Different superparameters have a significant impact on the prediction ability of LSTM and GRU models. Therefore, different superparameter data are set in the prediction to compare the prediction results. The number of neuron layers is set to 2 and 3, the number of neurons is set to 8, 16, and 32, the learning rate is usually set to 0.001, and the number of iterations is set to 1000. We can determine the most accurate prediction method by analyzing the prediction accuracy of the experimental results and the degree of fit of the trend between the predicted stock price and the historical stock price. The prediction accuracy is evaluated by mean square error function (MSE), root mean square error (RMSE), and mean absolute error (MAE) at different look-back values. The smaller the value of the three, the more accurate the forecast result is. The full specification of parameters used in these models is listed in Tables 2–5.

3.3. Experimental Results. The experimental results of stock prediction of four models are shown in Tables 2–5. Two different feature sets were obtained in this experiment. Set I is the data obtained from the LASSO dimension reduction method, and set II is the data obtained from the PCA dimension reduction method. These characteristic data are used to train LSTM and GRU models. In the experiment, different backtracking values were set. All parameter specifications used by the four models are shown in Tables 2–5.

Table 1: Explanation of technical indicator variables.

Primary indicator	Secondary indicator	Variable
Momentum indicators	ADX	Z1
	ADXR	Z2
	BOP	Z3
	CCI	Z4
	DX	Z5
	MACD	Z6
	MACDEXT	Z7
	MACDFIX	Z8
	TRIX	Z9
	MFI	Z10
	MOM	Z11
	PPO	Z12
	ROC	Z13
	RSI	Z14
	WILLR	Z15
	DPO	Z16
	DMA	Z17
	EXPMA	Z18
	DMI	Z19
Volume indicator	OBV.OBV	Z20
	OBV.MAOBV	Z21
	ADOSC	Z22
Volatility indicators	ATR	Z23
	NATR	Z24
	TRANGE	Z25
Cycle indicators	HT_DCPERIOD	Z26
	HT_DCPHASE	Z27
	HT_ PHASOR	Z28
	HT_ SINE	Z29
Moving average indicator	BBI	Z30
	SMA	Z31
Overbought and oversold	KDJ.K	Z32
	KDJ.D	Z33
	KDJ.J	Z34
	BOLL	Z35
	BIAS.BIAS1	Z36
	BIAS.BIAS2	Z37
	BIAS.BIAS3	Z38
	WR.WR1	Z39
	WR.WR2	Z40
Volume price indicator	ASI	Z41
	EMV	Z42
Energy indicator	PSY	Z43
	MASS.MASS	Z44
	MASS.MAMASS	Z45
	MTM	Z46
	BRAR-BR	Z47
	BRAR-AR	Z48
	VR	Z49
Stock price basic type	Open price	Z50
	High price	Z51
	Low price	Z52
	Volume	Z53

The results show that, through MAS, RMSE, and MAE indicators, both LSTM and GRU models can predict stock prices effectively, not one is more efficient than the other. However, for different dimension reduction methods, we find that all indicators (except the training time) show that

Table 2: Learning results of LASSO-LSTM.

Number of layers	Number of neurons	Look-back value	Train MSE	Train RMSE	Train MAE	Test MSE	Test RMSE	Test MAE	Train time
3	8	10	1904.5410	43.6410	29.6357	979.393	31.2953	23.7622	132.8780
3	8	20	1272.1884	35.6678	24.2016	904.9543	30.0825	22.6913	307.9990
3	8	30	2142.9192	46.2917	36.9945	957.2454	30.9394	22.6934	430.9928
3	8	40	4629.96388	68.0438	61.9541	5342.1396	73.0899	67.8341	576.7966
3	8	50	1214.7745	34.8536	23.3589	881.0265	29.6821	21.0112	766.2157
2	8	10	1248.6508	35.3363	23.4574	777.5208	27.8841	19.6554	95.1825
2	8	20	1608.0676	40.1007	27.8215	991.3244	31.4853	21.8147	213.3500
2	8	30	1251.1989	35.3723	22.9057	858.0669	29.2928	19.7988	320.6939
2	8	40	1202.1561	34.6721	23.1705	883.7101	29.7273	21.3382	411.6595
2	8	50	1206.7042	34.7376	23.1075	850.2483	29.1590	20.6415	490.2418
3	16	10	1031.0358	32.1097	21.8451	778.9459	27.9096	19.9938	376.8092
3	16	20	6555.5376	80.9663	75.5480	9468.3418	97.3054	93.4085	779.2415
3	16	30	1254.3915	35.4174	23.0273	733.8773	27.0902	18.6517	1189.4983
3	16	40	1015.5285	31.8674	21.3504	872.5764	29.5394	20.9425	1603.3826
3	16	50	984.6891	31.3798	22.6411	892.3263	29.8718	20.5884	2065.8919
2	16	10	1051.3169	32.4240	21.6973	786.5796	28.0460	19.8271	212.8883
2	16	20	1100.5430	33.1744	21.7411	734.0122	27.0927	18.7960	442.2047
2	16	30	1854.9745	43.0694	34.8938	1331.5276	36.4901	30.3015	725.7830
2	16	40	984.7316	31.3804	20.8187	803.8667	28.3525	20.9327	1080.9957
2	16	50	1006.7289	31.7290	21.1893	846.4849	29.0944	21.3891	1346.3720
3	32	10	787.5522	28.0634	20.2811	852.4590	29.1969	20.1759	306.4265
3	32	20	816.3383	28.5716	20.2925	1199.1475	34.6287	23.9940	636.0178
3	32	30	908.5170	30.1416	20.8915	1850.9948	43.0232	30.6120	970.8293
3	32	40	844.3165	29.0571	20.1408	2720.0981	52.1546	32.7257	1291.5366
3	32	50	1057.6416	32.5214	21.6094	2061.0061	45.3983	29.4876	1356.4536
2	32	10	1117.4025	33.4276	22.2174	799.0801	28.2680	19.8247	454.6504
2	32	20	919.8626	30.3292	21.2240	751.9146	27.4211	19.1840	929.6708
2	32	30	1096.1162	33.1076	23.0584	2220.7078	47.1244	31.1358	1358.4529
2	32	40	1124.6395	33.5356	23.5911	1125.7322	33.5519	25.2733	1385.7139
2	32	50	1442.5688	37.9812	29.5802	1449.2459	38.0690	29.8898	1497.0255

The number of epochs is 1000, learning rate is 0.001, and the activation function is *tanh*.

Table 3: Learning results of PCA-LSTM.

Number of layers	Number of neurons	Look-back value	Train MSE	Train RMSE	Train MAE	Test MSE	Test RMSE	Test MAE	Train time
3	8	10	1690.9171	41.1208	28.1397	10971.7773	104.7463	72.8242	94.6602
3	8	20	2939.5332	54.2175	36.5242	42278.7188	205.6179	176.8455	198.6585
3	8	30	3896.0701	62.4185	50.7139	5464.5537	73.9226	54.9697	286.6041
3	8	40	2811.6401	53.0249	35.2591	37730.9688	194.2446	160.7494	410.7228
3	8	50	2484.0608	49.8404	38.5345	30902.8711	175.7921	154.2262	541.6164
2	8	10	1619.0308	40.2372	27.5215	19627.638	140.0987	116.2432	60.6061
2	8	20	2298.5891	47.9436	34.5403	17482.4219	132.2211	102.2909	128.5815
2	8	30	2054.7803	45.3297	32.6867	24208.5762	155.5911	125.7793	191.2626
2	8	40	2300.3635	47.9621	34.4281	27250.1621	165.0762	143.2569	252.3014
2	8	50	1614.6926	40.1832	26.5031	27625.6934	166.2098	140.6607	313.6309
3	16	10	1022.4178	31.9753	23.6002	19104.1875	138.2179	105.8687	278.2536
3	16	20	1744.8813	41.7718	31.4407	100281.7656	316.6730	261.6820	543.4168
3	16	30	932.1148	30.5306	23.0709	5806.1943	76.1984	58.5185	846.4980
3	16	40	1688.8053	41.0951	31.0483	72460.7422	269.1853	225.7833	1090.1818
3	16	50	1101.5414	33.1895	25.5879	4054.6895	63.6764	42.8904	1654.7813
2	16	10	755.3936	27.4844	19.6168	7201.1304	84.8595	50.9353	93.1576
2	16	20	642.6140	25.3498	18.2508	3843.0637	61.9924	48.0259	207.6672
2	16	30	734.1151	27.0946	19.6506	13394.8955	115.7363	89.6245	311.5668
2	16	40	737.1022	27.1496	19.1860	5955.7412	77.1734	59.1523	427.4087
2	16	50	941.9175	30.6907	21.1289	4154.1406	64.4526	50.9237	666.0416
3	32	10	599.8295	24.4914	19.4298	35741.8242	189.0551	124.2747	514.5233

Table 3: Continued.

Number of layers	Number of neurons	Look-back value	Train MSE	Train RMSE	Train MAE	Test MSE	Test RMSE	Test MAE	Train time
3	32	20	594.4351	24.3810	18.6597	45484.7734	213.2716	169.4778	1064.2589
3	32	30	455.7460	21.3482	16.1783	17365.6328	131.7787	94.3915	1647.6020
3	32	40	805.4067	28.3797	23.0945	7770.9268	88.1529	58.4716	2789.5281
3	32	50	672.4887	25.9324	19.2775	91975.9453	303.2754	242.9042	2610.5209
2	32	10	495.8636	22.2680	16.6603	15152.0000	123.0935	90.1184	347.5115
2	32	20	627.7170	25.0543	18.4374	6696.0894	81.8296	50.5065	721.5676
2	32	30	454.6801	21.3232	15.7141	5454.8228	73.8568	51.7249	1089.6257
2	32	40	548.5927	23.4221	17.2341	17594.1758	132.6430	105.7155	1437.5246
2	32	50	407.0175	20.1747	14.6608	8100.0464	90.0003	66.1121	1808.6861

The number of epochs is 1000, learning rate is 0.001, and the activation function is *tanh*.

Table 4: Learning results of LASSO-GRU.

Number of layers	Number of neurons	Look-back value	Train MSE	Train RMSE	Train MAE	Test MSE	Test RMSE	Test MAE	Train time
3	8	10	1293.7802	35.9692	23.8084	831.7062	28.8393	20.3225	117.7369
3	8	20	1444.4125	38.0054	25.5757	910.9886	30.1826	21.6005	249.3114
3	8	30	1123.8232	33.5235	22.7495	826.1986	28.7437	20.0954	402.0947
3	8	40	1482.5836	38.5043	27.8070	943.8668	30.7224	20.9493	508.5391
3	8	50	1280.0657	35.7780	24.5826	978.3127	31.2780	24.1942	657.0113
2	8	10	1377.2985	37.1120	24.9828	834.2753	28.8838	20.0803	90.3936
2	8	20	1136.9731	33.7190	22.8381	799.1834	28.2698	19.5676	200.8435
2	8	30	1245.3809	35.2900	23.5962	864.6091	29.4042	20.8457	276.3479
2	8	40	1500.8726	38.7411	26.3263	1093.2498	33.0643	23.7652	350.2043
2	8	50	1239.8575	35.2116	23.4242	845.8825	29.0841	20.5453	431.4074
3	16	10	1138.9615	33.7485	22.0896	753.7004	27.4536	19.1470	378.2382
3	16	20	1192.8514	34.5377	24.7956	1015.7639	31.8711	23.3095	783.4717
3	16	30	985.8597	31.3984	21.1888	883.4285	29.7225	22.0179	1198.1594
3	16	40	1038.7191	32.2292	21.6453	802.5121	28.3286	19.8618	1622.0595
3	16	50	1116.0420	33.4072	22.7048	792.4473	28.1504	20.0095	2032.0963
2	16	10	1233.5467	35.1219	23.9346	833.0342	28.8623	19.9071	207.6725
2	16	20	1104.4722	33.2336	23.2089	869.4269	29.4860	21.8385	433.1125
2	16	30	1114.1602	33.3790	22.0850	796.8630	28.2288	20.7113	668.5479
2	16	40	1072.4717	32.7486	23.0368	802.4998	28.3284	19.7236	1073.6516
2	16	50	979.3882	31.2952	21.5892	844.8095	29.0656	19.7065	1373.0419
3	32	10	924.5891	30.4071	20.6728	831.0146	28.8273	20.2939	277.7194
3	32	20	862.2056	29.3633	20.4895	1081.1123	32.8803	24.4835	526.1457
3	32	30	907.1031	30.1182	21.2814	1127.4957	33.5782	23.5790	797.8424
3	32	40	847.7834	29.1167	21.0260	1012.2365	31.8157	21.7214	1066.6936
3	32	50	810.5232	28.4697	20.3346	1196.4927	34.5904	24.4721	1349.8547
2	32	10	969.9576	31.1441	20.7674	777.3851	27.8816	19.5336	367.9321
2	32	20	985.3215	31.3898	21.8014	829.0216	28.7927	20.1274	890.5871
2	32	30	861.6357	29.3536	20.5396	952.9534	30.8699	22.5985	1362.8664
2	32	40	1149.7027	33.9073	24.3771	821.3757	28.6597	20.0752	1392.1843
2	32	50	883.3071	29.7205	20.3320	833.8439	28.8764	20.4838	1596.1727

The number of epochs is 1000, learning rate is 0.001, and the activation function is *tanh*.

Table 5: Learning results of PCA-GRU.

Number of layers	Number of neurons	Look-back value	Train MSE	Train RMSE	Train MAE	Test MSE	Test RMSE	Test MAE	Train time
3	8	10	1369.6727	37.0091	27.3800	29235.9590	170.9853	152.2538	72.1253
3	8	20	2434.9895	49.3456	35.6683	12723.6592	112.7992	93.6385	168.9686
3	8	30	2101.9541	45.8471	33.8847	34108.7148	184.6854	166.6717	254.5802
3	8	40	2574.5256	50.7398	37.5205	21786.1641	147.6014	127.3568	333.7451
3	8	50	1194.8505	34.5666	25.1312	29938.3730	173.0271	151.4763	458.2760

Table 5: Continued.

Number of layers	Number of neurons	Look-back value	Train MSE	Train RMSE	Train MAE	Test MSE	Test RMSE	Test MAE	Train time
2	8	10	1196.4231	34.5893	23.0424	10500.0391	102.4697	82.4290	54.2256
2	8	20	2299.9802	47.9581	36.0542	29103.7676	170.5983	151.9113	112.4777
2	8	30	1426.4019	37.7677	26.6629	21631.0195	147.0749	121.6393	167.5188
2	8	40	1835.9618	42.8481	29.4047	9878.1016	99.3886	80.6860	225.1572
2	8	50	1503.7362	38.7780	26.7672	10440.7920	102.1802	86.5779	275.0539
3	16	10	943.2319	30.7121	21.8401	29209.2168	170.9070	148.9597	116.7384
3	16	20	816.0226	28.5661	20.8915	12768.4766	112.9977	99.0165	255.2906
3	16	30	965.2081	31.0678	23.1747	14524.5684	120.5179	99.9182	815.1597
3	16	40	980.0072	31.3051	22.1477	12230.3311	110.5908	84.3423	1096.2136
3	16	50	803.9420	28.3539	20.7994	9752.1914	98.7532	81.7505	1354.4110
2	16	10	776.4768	27.8653	19.3489	7601.5073	87.1866	74.7424	70.5103
2	16	20	784.2062	28.0037	21.0686	18093.4902	134.5120	119.4448	176.9984
2	16	30	747.7553	27.3451	20.0675	8023.9351	89.5764	77.0783	236.3435
2	16	40	720.6558	26.8450	19.3404	18241.6797	135.0617	117.5537	378.0467
2	16	50	713.9094	26.7191	19.0031	9412.2773	97.0169	79.6811	484.1664
3	32	10	453.6670	21.2995	15.6519	6122.3311	78.2453	61.3247	469.7184
3	32	20	456.4868	21.3656	15.9169	11951.0684	109.3209	90.6603	977.3712
3	32	30	626.6472	25.0329	19.3384	11865.4746	108.9288	86.8948	1483.5679
3	32	40	529.3604	23.0078	17.0553	9151.2109	95.6620	76.8583	1991.7930
3	32	50	504.7110	22.4658	16.9049	30787.4473	175.4635	153.6052	3289.7475
2	32	10	425.4210	20.6257	15.0201	6595.3359	81.2117	63.7287	268.0692
2	32	20	653.8412	25.5703	18.7442	10183.2344	100.9120	84.6054	539.5263
2	32	30	508.9430	22.5598	16.2366	6050.9692	77.7880	61.9705	856.6547
2	32	40	630.0575	25.1009	18.9956	7276.7456	85.3038	69.0245	1345.9522
2	32	50	623.7015	24.9740	18.0729	9594.8418	97.9533	80.5411	1678.2588

The number of epochs is 1000, learning rate is 0.001, and the activation function is *tanh*.

the prediction results of the two neural network models using LASSO dimension reduction are mostly better than those using PCA dimension reduction data. In other words, under the same network model, the prediction performance of LASSO-LSTM model is better than PCA-LSTM and the prediction performance of LASSO-GRU is better than PCA-GRU.

4. Conclusion

This study innovatively integrates a variety of technical indicators such as investor sentiment indicators and financial data and carries out dimension reduction on the multiple influencing factors of the extracted stock price through LASSO and PCA analysis approaches. This work carries out a comparison on the performances of LSTM and GRU for stock market forecasting under the different parameters. Our experimental results show that (1) both LSTM and GRU models can be used to predict stock prices effectively and (2) for different dimension reduction methods, the prediction results of the two neural network models using LASSO dimension reduction are mostly better than those using PCA dimension reduction data.

Data Availability

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

Conflicts of Interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as potential conflicts of interest.

References

[1] E. F. Fama, "Efficient capital markets: a review of theory and empirical work," *The Journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[2] P.-F. Pai and C.-S. Lin, "A hybrid ARIMA and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497–505, 2005.

[3] X. B. Huang, C. F. Wang, Z. M. Fang, and C. L. Xiong, "Detection of Chinese stock information based on hidden Markov model," *Syst. Eng.-Theory Pract*, vol. 32, no. 4, pp. 713–720, 2012.

[4] J. J. Shi and T. Song, "Analysis of long-term fluctuation trends and influencing factors in stock market-based on spline-GARCH model," *J. Appl. Stat. Manag*, vol. 34, no. 1, pp. 175–182, 2015.

[5] B. Xu, D. Zhang, S. Zhang, H. Li, and H. Lin, "Stock market trend prediction using recurrent convolutional neural networks," in *Proceedings of CCF International Conference on Natural Language Processing and Chinese Computing*, pp. 166–177, Hohhot, China, August 2018.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: current status and

future directions," *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, 2021.

[8] K. Chen, Y. Zhou, and F. Dai, "A LSTM-based method for stock returns prediction: a case study of China stock market," in *Proceedings of the 2015 IEEE international conference on big data (big data)*, pp. 2823-2824, IEEE, Santa Clara, CA, USA, October 2015.

[9] J. Jiahong Li, H. Junjie Wu, and J. Wu, "Sentiment-aware stock market prediction: a deep learning method," in *Proceedings of the 2017 international conference on service systems and service management*, pp. 1–6, IEEE, Dalian, China, June 2017.

[10] X. Jiawei and T. Murata, "Stock market trend prediction with sentiment analysis based on LSTM neural network," in *Proceedings of the International Multi-Conference of Engineers and Computer Scientists (IMECS)*, pp. 13–15, Hong Kong, China, March 2019.

[11] Y. W. Hu, "Stock forecast based on optimized LSSVM model," *Computer Science*, vol. 48, no. S1, pp. 151–157, 2021.

[12] K. Cho, B. Van Merriënboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.

[13] G. Shen, Q. Tan, H. Zhang, P. Zeng, and J. Xu, "Deep learning with gated recurrent unit networks for financial sequence predictions," *Procedia computer science*, vol. 131, pp. 895–903, 2018.

[14] M. O. Rahman, M. S. Hossain, T. S. Junaid, M. S. A. Forhad, and M. K. Hossen, "Predicting prices of stock market using gated recurrent units (GRUs) neural networks," *Int. J. Comput. Sci. Netw. Secur*, vol. 19, no. 1, pp. 213–222, 2019.

[15] H. Wang, X.-M. Zhang, G. Tomiyoshi et al., "Association of serum levels of antibodies against MMP1, CBX1, and CBX5 with transient ischemic attack and cerebral infarction," *Oncotarget*, vol. 9, no. 5, pp. 5600–5613, 2017.

[16] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, Springer, Berlin/Heidelberg, Germany, 2012.