The Superiority of XGboost Model in the Forecast of Medical Stock Price in the Period of COVID-19

Zhaofeng Ma[1,*]

[1]*International Business School, Gengdan Institute of Beijing University of Technology, Beijing, 101301, China*
[*]*Corresponding author. Email: mazf@imnu.edu.cn*

ABSTRACT
In this paper, the stock data of Fosun Pharma since the COVID-19 outbreak are used to fit the stock through the traditional statistical analysis model ARIMA, machine learning model XGboost, and deep learning model LSTM, and the future stock prices trend is predicted. It is shown that since the regular term is added to the machine learning model to control the complexity of the model, the generalization ability of the model is improved, and thus making the prediction effect better than the other two models. What's more, it explains that during the special epidemic period, due to the lack of data and many uncontrollable factors, the quoted data of the deep neural network model are often unrepresentative, resulting in the inconsistent characteristic distribution of the training set and the new data, leading to a series of problems such as over-fitting in the process of predicting individual stocks. Finally, it discusses the differences and advantages of the machine learning model compared with the traditional statistical learning model in the financial field.

Keywords: *LSTM, ARIMA, XGBoost, Stock price forecast, COVID-19*

1. INTRODUCTION

1.1. Background

Since the outbreak of COVID-19 in early 2020, it has spread to the whole country in just a few months. The outbreak has had a certain impact on national life and economic development, while the investment market is a microcosm of the economy, and the stock market is a barometer of economic development. Moreover, the impact of the epidemic on the capital market was gradually revealed in January 2020, with the stock market being the most affected by COVID-19. The development potential of medical stocks has received widespread attention from investors and stock prices have continued to rise, but the general market trend remains relatively weak as its rise is mainly due to the epidemic demand incentives and related policy benefits [1]. It is evident that the rise of medical stocks is affected by the epidemic in the short term, and share prices may gradually retrace to avoid causing excessive bubbles, and investment also needs to be combined with market laws. Therefore, it is of far-reaching significance for the stock investment market to be able to accurately predict the share prices of listed vaccine companies.

1.2. Related research

Traditional methods of technical analysis of stocks include the moving average method, point chart method, K-line chart method, etc. The prediction means are complicated and the predicted data cannot meet the needs of investors. Consequently, several in-depth studies have been conducted by many scholars. Qu analyzed the short-term trend of stock prices by using GM (1,1) isodimensional metabolic model, and according to the trend of GM (1,1) isodimensional metabolic model shows that the best time to buy shares [2]. Along with the time series forecasting method, Wu and Wen predicted the pattern and trend of stock price movements in the GEM market by building an ARIMA model. What they found is that the model has a better short-term dynamic and static forecasting effect, which can provide a useful reference for investors and enterprises in making relevant decisions [3]. To address the time lag of the traditional time series stock price forecasting model, Luo and Chen proposed a stock price forecasting model based on wavelet and dynamic GM(1, 1)-ARIMA. Wavelet analysis was used to preprocess the stock price data, and an ARIMA model was developed for the wavelet reconstruction series. The dynamic GM(1, 1) model is developed by considering the influence of future factors on the system during the forecasting process. And finally, the proposed wavelet and dynamic GM(1, 1)-ARIMA model was found to have the highest prediction accuracy compared with the traditional stock price prediction model [4].

As artificial intelligence technology and big data technology continue to be applied and developed, traditional time series models are limited by a fixed model framework and cannot make accurate predictions for complex financial time series, and RNNs (recurrent neural networks) are frequently used to analyze forecast series data. However, RNNs are usually difficult to train, and the gradient tends to disappear in most cases after many cycles, and there are also fewer cases where the problem of gradient explosion occurs. Responding to the problems of RNNs in practical applications, long short-term memory (LSTM) networks have been proposed, which have attracted much attention for their ability to maintain long-term storage of information, and a prediction method based on recurrent neural networks (RNNs), long short-term memory neural network (LSTM), arises spontaneously [5]. Ning and Zhang proposed a hybrid LSTM-Adaboost based network model using Adaboost integrated learning to train the optimal parameters, obtain the optimal forecasting model, and finally generate forecasting results [6]. Song et al. performed the optimization of key parameters of the LSTM model through the PSO algorithm with an adaptive learning strategy to match the stock data features with the network topology and improve the stock price prediction accuracy [7].

XGBoost algorithm, as a newly proposed algorithm in 2015, has the advantages of high computing efficiency and accuracy. Sun et al. applied the XGBoost model based on the grid search algorithm to the prediction of stock prices and concluded that the algorithm based on grid search optimization parameters can improve the prediction performance of the XGBoost model. However, the overall improvement of the prediction performance of the model is not high, and other methods need to be combined to improve the quality of data in the later stage [8]. In response to the low accuracy of previous stock prediction models due to the ponderous data and the shortcomings of traditional neural networks that are complex and time-consuming to train, Chen used Pearson's correlation coefficient analysis to extract the relevant features in the ponderous data that most affect the closing price trend and generate a dataset with high correlation features. Based on the Xgboost framework, which performed extremely well in data mining algorithms, combined with feature engineering processing, good training prediction results were achieved [9]. Wang and Guo constructed GS-XGBoost financial forecasting model by parameter optimization of the XGBoost model through a grid search algorithm. This included the application of the model to short-term stock forecasting. Finally, it was verified that the GS-XGBoost financial forecasting model has better fitting performance in short-term stock forecasting [10].

1.3. Objective

Based on the excellent performance of the machine learning model in stock prediction and the particularity of the LSTM network in data prediction, this paper further applies the machine learning model to stock prediction. It compares the three models. First, recurrent neural network, including LSTM model. Second, traditional statistical analysis models, including the ARIMA model. Third, integration algorithm, including XGBoost model. Observe the differences of machine learning models in the financial field compared to traditional statistical learning models and neural network models.

2. METHOD

2.1. LSTM

In 1997, Hochreiter and Schmidhuber proposed the LSTM neural network model, known as Long Short Term Memory networks, to solve the gradient disappearance problem faced by RNNs, which is a special type of RNN capable of learning long-term dependencies. The key of LSTM is the cell state, which consists of four main parts: the input gate (i_t) that controls the input of new information, the forgetting gate (f_t) that decides how much information is discarded, the output gate (o_t) that filters the final output information, and the cell state (c_t) that determines the current moment, where tanh, as an activation function in the LSTM model, is used to process the state and output of data,as shown in Figure 1.

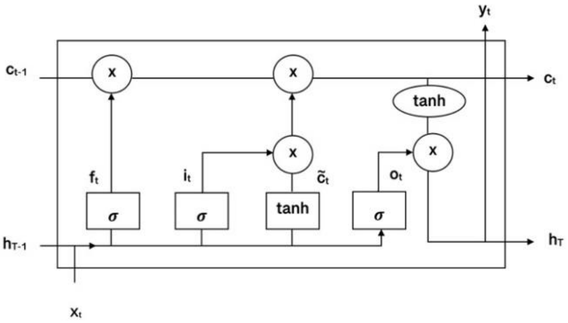

Figure 1 Structure of LSTM network model.

The first step of the LSTM is to decide which information that is going to throw away from the cell state. This decision is controlled by a Sigmoid layer called "Forget Gate". The Forget Gate looks at h_{t-1} and x_t and outputs a number between 0 and 1 for each element in the cell state c_{t-1}. 1 means "keep the information completely" and 0 means "discard the information completely". Where h_{t-1} represents the output of the previous cell and x_t represents the input of the current cell. σ represents the sigmod function.

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \quad (1)$$

The next step is to decide which new information will be stored in the cell state. This step is divided into two parts. First, a sigmoid layer called "Input Gate" determines what information we want to update. Next, a *tanh* layer creates a new candidate value, $\tilde{c}_t$, that may be added to the cell state. In the next step, the two values will be combined and used to update the tuple state.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$\widetilde{c_t} = tanh(w_c[h_{t-1}, x_t] + b_c) \quad (3)$$

Multiply the old state, c_{t-1}, by f_t, forgetting what has been decided to forget. Then add $i_t*\tilde{c}_t$, a value consisting of the new candidate value ($\tilde{c}_t$) multiplied by the degree to which each state of the candidate value decides to update it.

$$c_t = f_t c_{t-1} + i_t\widetilde{c_t} \quad (4)$$

Finally, the final output needs to be decided. The output will be based on the current tuple state and some filtering will be added. First, an output gate of the Sigmoid layer is created to decide which parts of the cell will be output. Then the cell state is passed through the *tanh* (so that the output value is between -1 and 1) and multiplied by the output gate so that only the desired part is output.

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t tanh(c_t) \quad (6)$$

2.2. ARIMA

2.2.1. Autoregressive model AR

The autoregressive model describes the relationship between the current value and the historical value and predicts itself with the historical time data of the variable itself. The autoregressive model must meet the requirements of stability.

The autoregressive model first needs to determine an order p, indicating that the current value is predicted by several historical values. The formula of the p-order autoregressive model is defined as :

$$y_t = \mu + \sum_{i=1}^{p} \gamma_i y_{t-i} + \varepsilon_t \quad (7)$$

In the formula y_t is the current value, μ is a constant, p is the order, ε_t is the error.

Autoregressive models have many limitations: 1. Autoregressive models predict with their own data.2. Time series data must be stationary.3. Autoregression can only be used to predict pre-related phenomena.

2.2.2. Moving average model MA

The moving average model is concerned with the accumulation of errors in the autoregressive model. The formula of the q-order autoregressive process is defined as follows :

$$y_t = \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-1} \quad (8)$$

Moving the average method can effectively eliminate random fluctuations in prediction.

2.2.3. Autoregressive Moving Average Model ARMA

Combining autoregressive model AR with moving average model MA, we obtain autoregressive moving average model ARMA (*p, q*), and the calculation formula is as follows :

$$y_t = \mu + \sum_{i=1}^{p} \gamma_i y_{t-1} + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-1} \quad (9)$$

2.2.4. ARIMA model of differential autoregressive moving average

By combining the autoregressive model, the moving average model, and the difference method, we obtain the ARIMA (*p, d, q*) of the difference autoregressive moving average model, where *d* is the order of difference of data.

2.3. XGBoost

XGBoost is one of the boosting algorithms. The idea of Boosting algorithm is to integrate many weak classifiers together to form a strong classifier. Since XGBoost is a boosting tree model, it is integrating many tree models together to form a very strong classifier.

2.3.1. XGBoost algorithm idea

The idea of the algorithm is to keep adding trees and keep splitting features to grow a tree. Each time a tree is added, it actually learns a new function to fit the residuals of the last prediction. When we finish training to get *k* trees, we want to predict the score of a sample, in fact, according to the characteristics of this sample, in each tree will fall to the corresponding leaf node, each leaf node corresponds to a score, and finally just add up the corresponding scores of each tree is the predicted value of the sample.

$$\hat{y} = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i) \quad (10)$$

$$whereF = \{f(x) = w_{q(x)}\}(q: R^m \underset{0}{\rightarrow} T, w\epsilon R^T) \quad (11)$$

Note: $w_{q(x)}$ is the *score of leaf node q, and f(x) is* one of the regression trees

2.3.2. XGBoost Principle

The XGBoost objective function is defined as

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{12}$$

$$where \Omega(f) = \sqrt{T} + \frac{1}{2}\lambda||w||^2 \tag{13}$$

The XGBoost objective function consists of two parts, the first part is used to measure the difference between the predicted and true scores, and the other part is the regularization term. The regularization term also contains two parts, *T* denotes the number of leaf nodes and *w* denotes the score of leaf nodes. *γ* controls the number of leaf nodes and *λ* controls that the fraction of leaf nodes is not too large to prevent overfitting.

The newly generated tree is to fit the residuals of the last prediction, i.e., when t trees are generated, the predicted scores can be written as:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{14}$$

Also, the objective function can be rewritten as:

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{15}$$

The next step is to go about finding an ft that minimizes the objective function. The idea of XGBoost is to approximate it using its Taylor second-order expansion at f_t = 0. So, the objective function is approximated as:

$$L^{(t)} \simeq \sum_{i=1}^{n}[l(y_i, \hat{y}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \tag{16}$$

Where g_i is the first-order derivative and h_i *is the second-order derivative.*

$$g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}) \tag{17}$$

$$h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}) \tag{18}$$

Since the predicted scores of the first *t-1* trees and the residuals of *y* have no effect on the optimization of the objective function, they can be removed directly. Simplifying the objective function as:

$$\tilde{L}^{(t)} = \sum_{i=1}^{n}[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \tag{19}$$

The above equation adds up the loss function values of each sample, and each sample will eventually fall into a leaf node, so we can reorganize so the samples of the same leaf node in the following process.

$$Obj \simeq \sum_{i=1}^{n}[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$

$$= \sum_{i=1}^{n}[g_i w_q(x_i) + \frac{1}{2} h_i w_q^2(x_i)] + \sqrt{T}$$

$$+ \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

$$= \sum_{j=1}^{T}[(\sum_{i \epsilon 1_j} g_i) w_j + \frac{1}{2}(\sum_{i \epsilon 1_j} h_i + \lambda) w_j^2] + \sqrt{T} \tag{20}$$

Thus, by rewriting the above equation, we can write the objective function as a quadratic function about the fraction w of leaf nodes, and solving for the optimal *w* and objective function values is now easy. In this way, the best w and objective function formulas are now known.

$$w_j^* = -\frac{G_j}{H_j + \lambda} \tag{21}$$

$$Obj = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \sqrt{T} \tag{22}$$

3. RESULTS AND DISCUSSION

3.1. LSTM model building and analysis

3.1.1. Data sources

The data in this article is obtained from the Tushare database, and the stock opening price, closing price, high price, low price, volume, and turnover of Fosun Pharma from January 2, 2020, to December 13, 2021, are selected from Tushare as the original data, totaling 472 trading days. The sample size basically covers the data changes for all trading days since the Fosun Pharma outbreak.

3.1.2. Pre-processing of data

3.1.2.1. Division of data

In this paper, we divide the data into training data set and testing data set according to the ratio of 9:1, i.e., the data of all trading days of Fosun Pharma on and before September 30, 2021, as the training set, totaling 425 trading days; the data of all trading days of Fosun Pharma on and after October 1, 2021, as the testing set, totaling 47 trading days. The chart below shows the closing index of Fosun Pharma for all trading days from January 1, 2020, to mid-December 13, 2021.

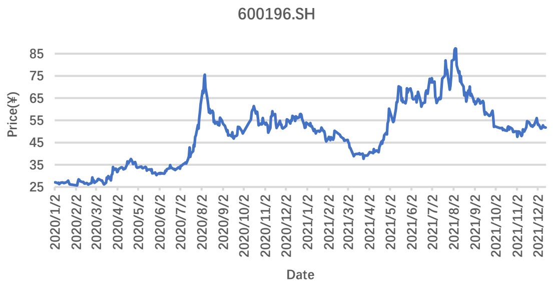

Figure 2 Fosun Pharma Closing Index Trend.

3.1.2.2. Standardization of data

After dividing the data into training and test sets, the data needs to be normalized. If no normalization is performed, the values of different features will vary greatly, which will cause the objective function to become "flat", and when gradient descent is performed, the direction of gradient update will deviate from the direction of the minimum value, making the training time grow. In this paper, the minimum-maximum scaling method is adopted to scale the data between 0 and 1. The maximum value of the minimum value of the training set data is first extracted, and then both the training set data and the test set data are processed using Equation (23). Both the training set and the test set must be normalized based on the data in the training set, because the information in the test set is usually assumed to be unknown in advance when building the model, and the sample size of the test set is small, so if the mean and standard deviation of the test set is used to normalize the data in the test set, it may cause the normalized data to deviate from the overall data.

$$x_{std} = (x - x_{min})/(x_{min}) - (x_{max}) \qquad (23)$$

3.1.3. Single-featured LSTM networks

3.1.3.1. Scrolling time window setting

In this paper, we first train the LSTM model using the closing price data of Fosun Pharma as the time-series data. Before training the network, the data need to be further processed - set the rolling time window, the size of the time window set in this paper is 5, i.e., each rolling 5 days, and use the closing price data of the 5 days as the feature and the closing price of the next day as the label. After processing, the training set feature dimension is (420, 5, 1), the training set label dimension (420, 1), the test set feature dimension (42, 5, 1), and the test set label (42, 1).

3.1.3.2. LSTM network training

In this experiment, two layers of LSTM nets are stacked, each layer is specified as 50 neurons, and the dropout layer is set with parameters set to 0.2, the optimizer is specified as adam, and the loss function is mse. When training the network, specify epochs as 200 and batch_size as 128.

3.1.3.3. Test set prediction results

The trained model was used to make predictions on the test set. Computing the RMSE predicted by the model on the test set as: 1.627 with the following fitted trend graph.

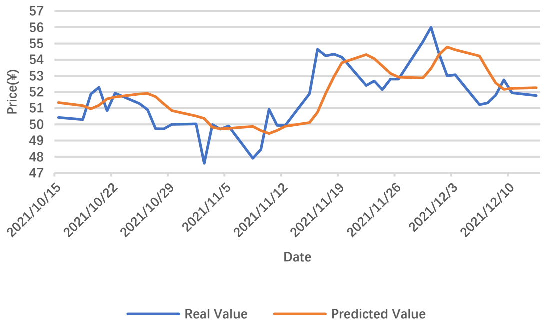

Figure 3 Single Feature LSTM model forecasts for the most recent quarter.

3.1.4. LSTM networks with multiple features

3.1.4.1. Scrolling time window setting

This step is roughly the same as the single-feature LSTM modeling, and it is worth noting that because it is multi-feature modeling, i.e., not only the closing price data is used, but also the opening price, the highest price, the lowest price, the volume and the turnover information, the data dimensions of the input model are changed, and after processing, the training set feature dimensions are (420, 5, 6), the training set label dimensions (420, 1), test set feature dimension (42, 5, 6), and test set label (42, 1).

3.1.4.2. LSTM network training

This step specifies the same parameters as the single feature, the only difference is that the specified dimension input_shape of the input data is changed. where the FEATURE changed from 1 to 6.

3.1.4.3. Test set prediction results

The trained model was used to make predictions on the test set. Computing the RMSE predicted by the model on the test set as: 3.3 Test set prediction results.

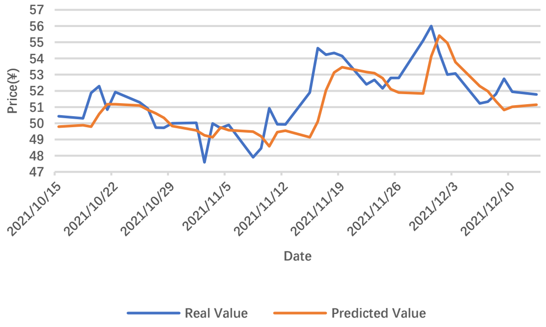

Figure 4 Multi-feature LSTM model forecasts for the most recent quarter.

3.1.5. Analysis of experimental results

By comparing the results of single-feature and multi-feature LSTM models, this paper reveals that the introduction of multiple features does not improve the prediction accuracy of the model.

Table 1. The results of single-feature and multi-feature LSTM models

Models	MSE
Single Feature LSTM	1.627
Multi-feature LSTM	1.748

3.2. Establishment and analysis of ARIMA model

3.2.1. Data source

The data in this article are derived from the Tushare database, and the closing prices of Fosun Pharmaceuticals from January 2, 2020 to December 13, 2021 are used as the original data. There are a total of 472 data samples in this article. The sample size essentially encompasses the closing prices of all Fosun Pharma trading days since the outbreak.

3.2.2. Unit root test and smoothing of the data

Through Python visualization, we observe the time-series images of the closing prices of Fosun Pharmaceuticals, and we find that the closing price time series data of Fosun Pharmaceuticals exhibit non-smooth characteristics. Therefore, we performed a unit root test on the closing price data, and after performing a unit root test on the data, we obtained a p-value of 0.2992 greater than 0.05 for the ADF test. It can be concluded that the original series is non-stationary. The non-smoothness of the time series data can be solved by the difference method, so we perform the first-order difference on the original data and conduct the ADF test again to obtain the P-value, which is 4.38733424269911e-13 less than 0.05 after the first-order difference processing of the data. This indicates that the series meets the requirement of smoothness after the first-order differencing of the data.

3.2.3. Establishment of ARIMA model and parameter estimation

The key to establishing the ARIMA model is the need to determine the p and q parameters. Usually, the p and q parameters can be determined by observing the ACF and PACF plots, but for the rigor of the parameters, the BIC or AIC information criterion is generally used for judgment. In this paper, the minimum value of BIC information is found by the toolkit of statsmodels. According to the output of Python, the optimal p and q values are (3, 2) respectively. We select several sets of data from the output of the ARMA_order_select_ic tool class as follows, among which the smallest BIC value corresponding to ARIMA (3, 2) is visible. Finally, combining the differential information, we choose to fit the data with ARIMA (3, 1, 2) for model construction.

Table 2.BIC values for different ARIMA models

	ARIMA (3, 0)	ARIMA (3, 1)	ARIMA (3, 2)	ARIMA (0, 0)	ARIMA (0, 1)	ARIMA (0, 2)
BIC value	1990.007	1994.951	1974.742	1979.310	1985.420	1984.243

3.2.4. ARIMA model testing

Following the estimation of the model parameters, we must test the model. That is, in order to determine whether the residual term follows the normal distribution, we typically visualize the residuals. The model fits correctly if the probability density distribution of the residuals is normal. Following the execution of the Python program, the following output visualization results are displayed.

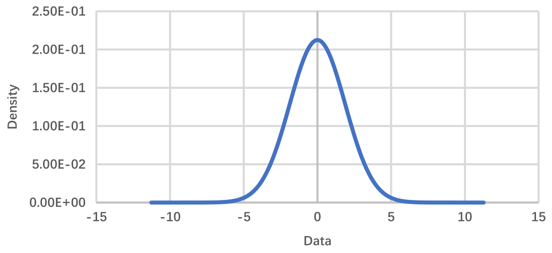

Figure 5 Probability density distribution of residuals.

Note: The line chart is the residual of fitting the model and plots its probability density.

The mean value is 0.000609 and the standard deviation is 1.880288, which meets the requirements of normal distribution.

3.2.5. ARIMA model prediction results and analysis

In this paper, the data are divided into the training set and test set, and October to December 2021 is taken as the test set, and the rest is the training set. The sample size of the training set is 425, and the sample size of the test set is 47.

Generally, the ARIMA model is only suitable for short-term forecasting and has a poor effect on long-term forecasting. To enhance the model's predictive ability, this paper chooses to use a rolling fit for the test data. The ARIMA model predicts the data for the first day of the test set using the training data, then the real data for that day of the test set is added to the training data set, and the model is fitted to the training data set to predict the data for the second day of the test set, and so on, until all predictions for the test set are complete.

The RMSE of the model's prediction on the test set is calculated as: 1.287, and the fitted trend graph is as follows.

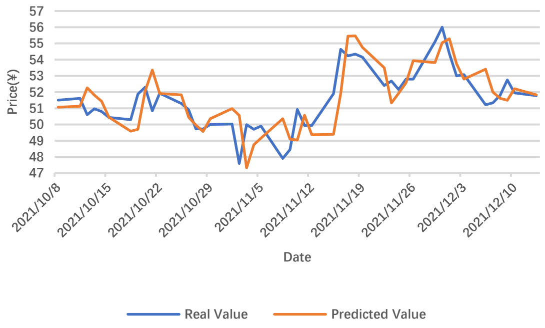

Figure 6 ARIMA model forecasts for the most recent quarter.

3.3. XGboost model development and analysis

3.3.1. Data sources

The data in this paper is sourced from the Tushare database. The stock opening price, closing price, high price, low price, volume, and turnover of Fosun Pharma from 2 January 2020 to 13 December 2021 were selected from Tushare as the original data, with a total of 472 trading days data. The sample size basically covers the data changes for all trading days since the Fosun Pharma outbreak.

3.3.2. Pre-processing of data

3.3.2.1. Division of data

In this paper, the data is divided into training data set and test data set according to the ratio of 9:1, i.e. the data of all trading days of Fosun Pharmaceuticals on and before 30 September 2021 as the training set, totaling 425 trading days of data; the data of all trading days of Fosun Pharmaceuticals on and after 1 October 2021 as the test set, totaling 47 trading days of data.

3.3.2.2. Feature and label selection

XGboost, being a supervised learning algorithm with excellent performance, requires that we specify features and labels. In this paper, we select the daily opening price, closing price, high price, low price, volume, and turnover of stock as features and the closing price of the next day as the label.

3.3.3. boost model construction

3.3.3.1. Selecting model hyperparameters using GridSearchCV

In machine learning models, the parameters that need to be chosen manually are called hyperparameters. For example, the depth, number of trees, learning rate, etc. in the XGboost model, all need to be specified in advance. If the hyperparameters are not selected properly, underfitting or overfitting problems can occur. When choosing hyperparameters, there are generally two ways, one is to specify the parameters manually by experience, and the other is to have a fixed algorithm automatically choose the hyperparameters.

GridSearchCV, as a commonly used method of automatic parameter tuning, is made up of two main components: GridSearch and CV, i.e. grid search and cross-validation. GridSearch, which searches for parameters, means that the parameters are tuned sequentially in steps within a specified range of parameters, and the tuned parameters are used to train the learner to find the parameters with the highest accuracy on the validation set from all the parameters. The advantage of GridSearchCV lies in the fact that it can guarantee to find the parameters with the highest accuracy within the specified parameter range.

In this paper, the tree depth, learning rate, and the number of trees are selected as the adjusted hyperparameters, and after specifying the parameter steps, a hyperparameter dictionary is formed and the model is trained using the parameters selected by GridSearchCV.

3.3.3.2. XGboost training

The XGboost model and hyperparameter dictionary are passed into GridSearchCV, and then the training data can be trained on the model. GridSearchCV will return the best hyperparameters and the best model, with the best hyperparameters as follows.

Table 3. The best hyperparameters of GridSearchCV

learning_rate	0.1
max_depth	10
n_estimators	100

3.3.3.3. Test set prediction results

Predictions were made on the test set using the trained model. The RMSE predicted by the computational model on the test set was: 1.241 and the fitted trend graph is as follows.

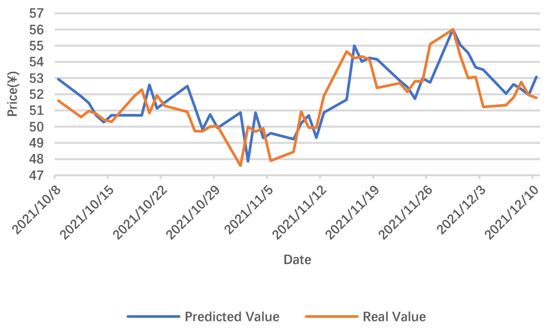

Figure 7 XGboost model forecasts for the most recent quarter.

3.4. Performance of the model on the test set

The comparison in the table shows that the best prediction result is the XGboost model, followed by the ARIMA model and finally the LSTM model. The single-feature LSTM again outperforms the multi-feature LSTM, and it is speculated that this may be due to the introduction of other features that bring in noise and so affect the model effect

Table 4. Prediction results of different models

Models	RMSE
ARIMA	1.287
Single Feature LSTM	1.627
Multi-feature LSTM	1.748
XGboost	1.241

4. CONCLUSION

This paper uses the traditional statistical analysis model ARIMA, the machine learning model XGboost, and the deep learning model LSTM to fit stocks since the revival of the pharmaceutical epidemic and learn the future trend of stocks through the model. For the ARIMA model, the main learning is done using information from one latitude of the stock closing price. Two approaches are used for the LSTM model, one with a single feature and the other with multiple features, and the results are compared. The main approach for the XGboost is to find features to build a supervised learning approach. In contrast, the XGboost model has a better prediction effect on individual stocks in one and a half years than the other two models. Especially compared with the LSTM model, the XGboost model adds regular terms to control the complexity of the model, which prevents over-fitting and improves the generalization ability of the model. In intricate stock markets, individual stocks bear the shortcoming of many uncertainties. As a result, the data of individual stocks are often unrepresentative with only a few training set data, which results in the inconsistent characteristic distribution of training set and new data characteristics, which often occurs in the case of individual stock prediction. To sum up, the XGboost model has incomparable advantages over the other two models in predicting individual stock prices during a specific epidemic period.

REFERENCES

[1]Y.L. Xia,Bio-pharmaceutical stocks under the COVID-19 epidemic Investment value analysis—An Empirical Study Based on Factor Analysis,Modern Business Trade Industry,vol.41,no.33,2020,pp.109-110.DOI:10.19311/j.cnki.1672-3198.2020.33.051

[2]S.D. Qu,GM (1,1) Model Analyses Stock Price Trend,Journal of Guangxi University for Nationalities (Natural Science Edition),2001,pp.170-173.DOI:10.16177/j.cnki.gxmzzk.2001.03.004

[3]Y.X. Wu and X. Wen, Short-term Stock Price Forecast Based on ARIMA Model, Statistics and Decision,2016,pp.83-86.DOI:10.13546/j.cnki.tjyjc.2016.23.051

[4]H. Luo and Y.F. Chen,Based on Wavelet and Dynamic GM (1,1)-ARIMA Research on Stock Price Forecast Based on the Model,Journal of Zhejiang Sci-Tech University(NaturalSciences),vol.37,no.4,2017,pp.575-579.DOI:10.3969/j .issn . 1673-3851 . 2017 . 07 . 018

[5]D.X. Zhang,Y.T. Chen and J. Meng,Synthetic well logs generation via Recurrent Neural Networks,Petroleum Exploration and Development,vol.45,no.4,2018,pp.598-607.DOI: 10.11698/PED.2018.04.06

[6]Y.B. Ning, Y.J. Zhang,LSTM-Adaboost stock price forecast model,Journal of University of Science and Technology Liaoning,vol.42,no.5,2019,pp.383-388,DOI:10.13988/j.ustl.2019.05.012

[7]G. Song, Y.F. Zhang,F.X. Bao and C. Qin,Stock Forecast Model Based on Particle Swarm

Optimization LSTM, Journal of Beijing University of Aeronautics and Astronautics, vol. 45,no.12,2019, pp. 2533-2542.DOI:10.13700/j.bh.1001-5965.2019.0388

[8] L.L. Sun,H.B. Fang,X.X. Zhu,L.M. Hu and L.W. Qi,Stock prediction using XGBoost model based on grid search optimization,Journal of Fuyang Normal University(Natural Science),vol.38,no.2,2021,pp.97-101.DOI:10.14096/j.cnki.cn34-1069/n/2096-9341（2021）02-0097-05

[9]Y.S. Chen, Z.J. Tang, Y. Luo and J. Yang, Research on Stock Price Forecast Based on Pearson Optimization and Xgboost Algorithm, Information Technology,2018,pp. 84-89.DOI: 10. 13274 /j. cnki. hdzj. 2018. 09. 019

[10]Y. Wang and Y.K. Guo , Application of Improved XGBoost Model in Stock Forecast, Computer Engineering and Applications,2019, pp.202-207.DOI:10.3778/j.issn.1002-8331.1904-0007