Prediction of A-share Prices Based on K-means Clustering Algorithm and LSTM

Xingyu HE, XH, He
University of Shanghai for Science and Technology, Shanghai, China
hownsit@163.com

ABSTRACT

In this paper, a long and short-term memory neural network is used to predict the future trend of A-share prices [2, 3, 5–8, 10], while the K-means clustering algorithm reduces the number of models in the whole process and achieves high compatibility of the model with different stocks. The individual stocks of the A-share Shanghai exchange in February 2021 are divided into 31 clusters according to their time-of-day K-chart data, and LSTM neural networks are used to construct respective models for the logical prime stocks of these 31 clusters. The effect of using a small number of models to predict the prices of multiple stocks is finally achieved. Therefore, when implementing the prediction function for a new A-share, it is possible to use an existing compatible generalized model, and there is no need to build a separate model for the stock before determining whether it has predictive value, which reduces the initial cost in implementing the prediction function.

CCS CONCEPTS

• **Networks**; • **Network algorithms**; • **Network economics**;

KEYWORDS

Machine Learning, K-means, LSTM, Stock Price Prediction

ACM Reference Format:
Xingyu HE, XH, He. 2022. Prediction of A-share Prices Based on K-means Clustering Algorithm and LSTM. In *2022 the 5th International Conference on Big Data and Internet of Things (BDIOT 2022), August 12–14, 2022, Chongqing, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3561801.3561804

1 INTRODUCTION

Stock is a multiplayer game, and its price fluctuations are often affected by a variety of complex factors: economic cycles, changes in the country's financial situation; national policy adjustments or changes; the issuing company's operating performance, and other factors of the company itself; changes in the industry's position in the national economy; investor movements, the intentions and manipulations of large investors; and the psychological state of investors after being influenced by various aspects of change. Some factors are difficult to measure and record, such as human investment psychology or the manipulation intentions of large investors. Some factors are difficult to quantify, so it is extremely difficult to try to predict changes in stock prices by these factors. When the causal mapping process is ignored, the difficulty of prediction can be reduced by directly studying the results of the combination of these factors and exploring the patterns between the different results. Currently, the use of long and short-term memory neural networks in stock price prediction is a feasible approach that is commonly accepted.

So far, some are also scholars who use long and short-term memory neural networks to do stock price prediction [11, 13], but theirs are more about stock-specific and do not give a solution when dealing with a large number of stocks. When dealing with a large number of different stocks, their solutions require building models for each stock. This paper achieves a fast prediction of individual stock prices of the A-share Shanghai exchange by using the K-means unsupervised clustering method and long and short-term memory neural network.

Figure 1: Steps and processes of the simulation.

2 ANALYSIS

2.1 Process Introduction

The simulation steps and process of predicting the relevant price signals at the next moment based on the A-share quotation data of the Shanghai stock exchanges are shown in Figure 1.

2.2 Data Acquisition

Because LSTM neural network is mainly used to deal with problems with long-term dependence, this paper will be based on a certain amount of historical data, and here 1090 individual stocks' time-of-day K-line chart data of A-shares in January 2021 are chosen, which needs to include 7 feature values [1]: trading date, latest price, highest price, lowest price, opening price, trading volume, and yesterday's closing price.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
BDIOT 2022, August 12–14, 2022, Chongqing, China
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9036-1/22/08…$15.00
https://doi.org/10.1145/3561801.3561804

3 DATA PRE-PROCESSING

Missing values are processed for the obtained historical data which are an important problem that often occurs in time-series data. Mean-complement method is chosen in this paper.

To implement K-means clustering of stocks through historical data, a set of values that can reflect the true level of this stock needs to be determined, which are the latest price, highest price, lowest price, opening price, trading volume, and yesterday's closing price. Here, the weighted average of the individual stock's eigenvalues in all time-of-day K-lines over one month is chosen. A weighting rule should be specified to eliminate the influence generated by the maximum and minimum values. As in equation 1), the size of the weight w_i depends on the distance of the value x_i from its mean. The point far from the mean group has the highest probability of being an outlier; therefore, its weight should be the smallest. And $max((x - mean(x))^2)$ is the farthest distance from the mean in this group of values.

$$w_i = 1 - \frac{(x_i - mean\,(x))^2}{\max\left((x - mean\,(x))^2\right)} \tag{1}$$

In stock trading, the price and the trading volume are often closely related. To improve the specificity and accuracy of the model. The stocks are chosen to be discarded with small trading volume fluctuations and small trading volume representative values. The coefficient of variation of each stock is calculated, and the principle of the minimum number of discards is observed. Finally, the stocks with a coefficient of variation greater than 2.38 and the representative value of trading volume less than 10000 are chosen to be discarded.

4 K-MEANS CLUSTERING

4.1 Method Introduction.

K-means is an iterative solution clustering analysis algorithm, in which K objects (points) are randomly selected as the initial cluster centers, and then the distance between other objects (points) and each cluster center is calculated, and each object (point) is assigned to the cluster center nearest to it. The cluster centers and the objects (points) assigned to them represent a cluster. After each cluster is assigned, the cluster centers of each cluster are recalculated based on the existing objects (points) in the cluster. This process is repeated until a termination condition is met. The termination conditions can be that no (or a minimum number of) objects are reassigned to different clusters, no (or a minimum number of) cluster centers change again, and the error sum of squares is locally minimized.

4.2 Cluster Processing.

K-means clustering is used to classify the retained stocks and group the ones with similar trends into one category. Therefore members in a group could share an initial growth model and refer to the parameters in this model when training a new model or updating it later.

To evaluate the effect of K-means clustering, this paper uses the concept of high cohesion and low coupling. In classification, the smaller the distance between members, and the larger the distance between different classes, the better it is. In Figure 2, which is an example of clustering results for eight two-dimensional coordinate points, the obtained classification has clear boundaries and high clustering, which are good classification results.

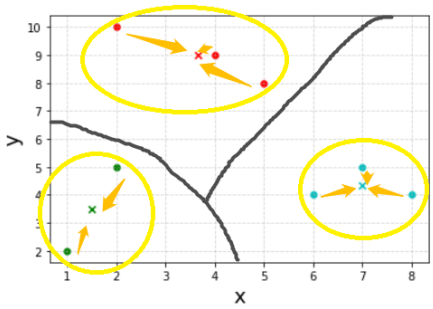

Figure 2: An example of a good clustering effect.

The *score* shown in equation 2) is used as an index to evaluate the effect of clustering, and the better the classification is when the value of the *score* is smaller. Where D_{inner} is the average Euclidean distance between class members and D_{outer} means the average Euclidean distance between class prime centers.

$$score = \frac{D_{inner}}{D_{outer}} \tag{2}$$

The Euclidean distance is calculated as:

$$D\,(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3}$$

To be able to obtain the exact number of classifications, the individual classification effect evaluation was obtained by traversing the number of clusters from 2 (inclusive) to 38 (inclusive). As in Figure 3, the X-axis is the number of different clusters(same in Figure 4 & Figure 5), and the Y-axis is the value of the score of the evaluation index of the clustering effect. The figure shows that the best result is achieved when the number of clusters is 31. Although the score value is also close to this optimal value when the number of clusters is 34 and 35, it is important to give preference to those with fewer clusters, not only to reduce the complexity but also to prevent over-classification. In Figure 4, the Y-axis is the "cohesion" distance (the distance to the respective center of mass) of all members(Clustering is oriented to all members, and the total number of members in different categories is constant, so there is no need to average it.). The figure shows that the best cohesiveness is achieved when the number of clusters is also 31. In Figure 5, the Y-axis is the "coupling" distance (the distance between different classes) between all classes. The figure shows that the coupling is lowest when the number of clusters is also 31.

The actual center of mass is abstract and does not necessarily correspond to stock in the data set, so the stock closest to the actual center of mass is chosen as the logical center of the class. As in Figure 6, the point selected by the blue rectangle will replace the "fork" (actual center of mass) in the class as the center of mass (logical center of mass) of the class.

The final clustering results are shown in Figure 7, with the logical center of mass on the X-axis and the number of cluster members on the Y-axis.

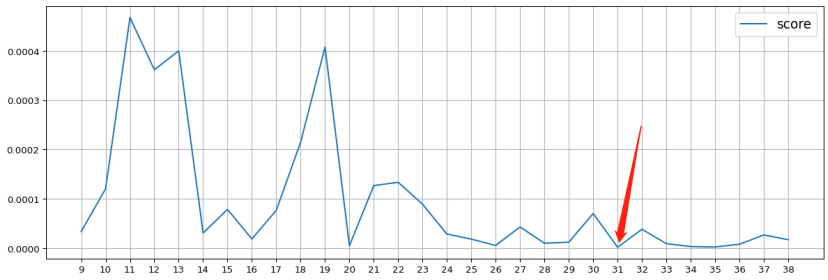

Figure 3: Evaluation indicator "score" chart.

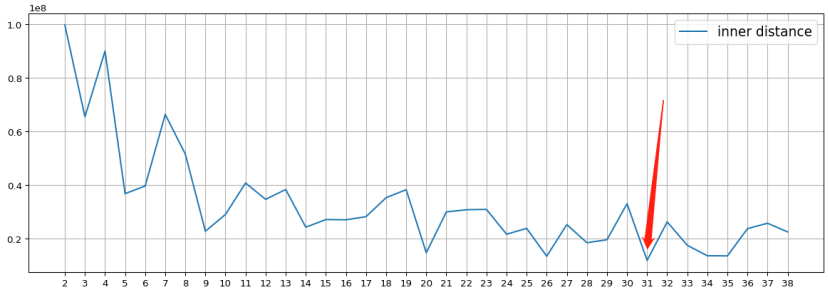

Figure 4: "Cohesive" distance.

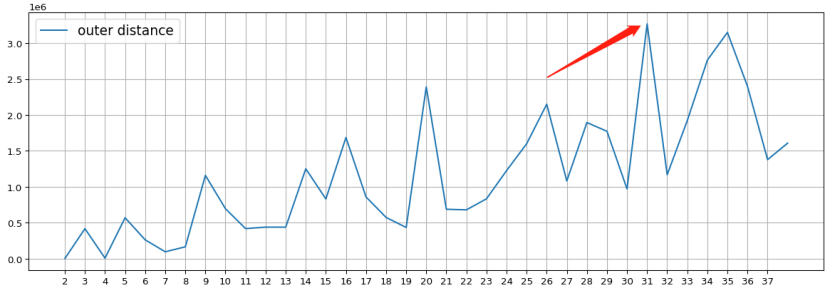

Figure 5: "Coupling" distance.

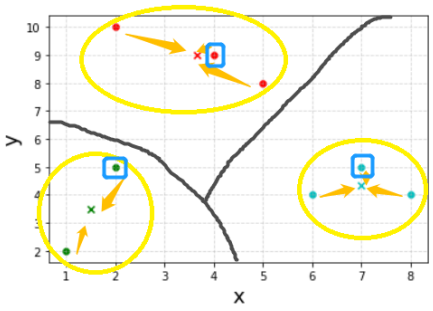

Figure 6: Example of the selection of a logical center of mass.

5 BUILDING MODELS

5.1 Building LSTM Models

In this paper, the Sequential model in “Keras” is chosen, which enables a linear stacking of multiple network layers. By adding a LSTM neural network to it, long-term dependencies can be learned, with a more complex internal structure that can choose to adjust the transmitted information by gating the state, remembering the information that needs to be remembered for a long time and forgetting the unimportant information [14].

LSTM is a deformation of recurrent neural networks that avoids gradient disappearance and gradient explosion like RNNs produce in long-distance transmission. Therefore, it has an inherent advantage in solving long-term timing problems. As shown in Figure 8, the control flow of LSTM is similar to RNN in that they both process the data flowing through the cells during forwarding propagation, with the difference that the structure and operation of the cells are

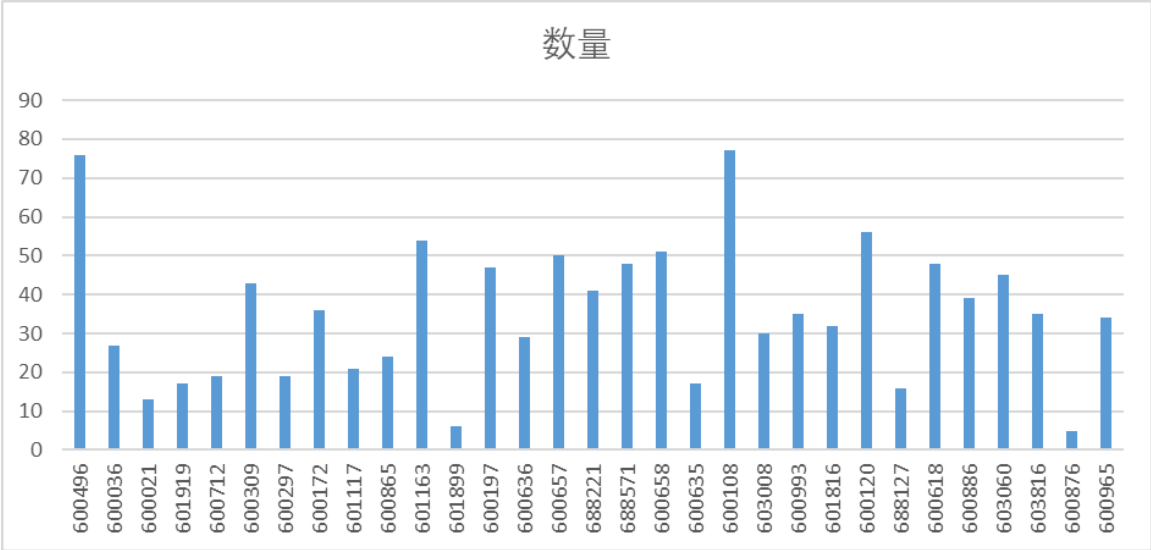

Figure 7: Clustering results.

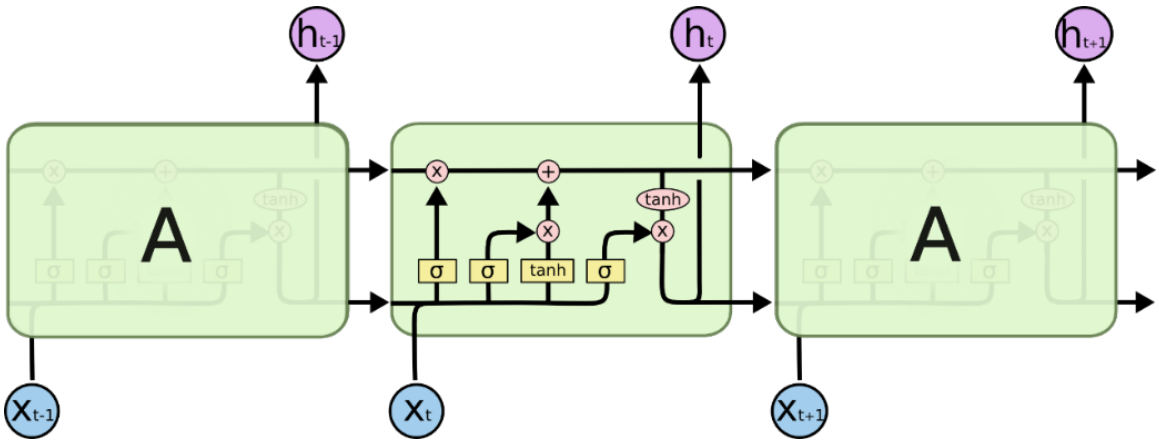

Figure 8: The structure of LSTM. (https://img-blog.csdnimg.cn/20190317220528691.png)

changed in LSTM. The horizontal line across the cell at the top is the cell state, which is the direction of information flow.

The core concept of LSTM is the cell state and the "gate" structure. These are the forgetting gate, the input gate, and the output gate, respectively. The cell state is equivalent to the pathway of information transmission, allowing information to be passed on in a sequence. The function of the forgetting gate is to decide which information should be discarded or retained; the input gate is used to update the cell state; and the output gate is used to determine the value of the next hidden state, which contains the previously entered information. The gate structure contains the sigmoid activation function. The sigmoid activation function is similar to the tanh function, except that the sigmoid compresses the value between 0 and 1 instead of between -1 and 1. This helps to update or forget information, because any number multiplied by 0 gives 0, and this information is eliminated. Similarly, any number multiplied by 1 gets itself, and that part of the information is perfectly preserved. This way the network can understand which data is to be forgotten and which data is to be saved.

5.1.1 Data Pre-Processing. Because of the different magnitudes of the data eigenvalues, to improve the accuracy of the model, it needs to be standardized first, and the standardization formula is shown in equation 4).

$$x_s = \frac{x_i - x_{mean}}{x_{std}} \tag{4}$$

In equation 4), x_s means the normalized data, x_i means the original data, x_{mean} means the mean of the original data, and x_{std} means the standard deviation of the original data.

5.1.2 Building LSTM Networks. To obtain the best number of prediction retracements and model parameters for each class, this paper finds the best-evaluated model by setting several possible

Table 1: Table of possible values of the parameters

Parameter Name	Optional parameters
Number of retrospective time	[1–4]
Number of neurons	[96,112,128]
Number of LSTM layers	[1–4]
Number of fully connected layers	[1–4]

optimal parameters and keeps its parameters as the optimal combination of parameters for that class to predict its eigenvalues. The optional parameters are shown in Table 1, in addition to which an additional LSTM layer is set as the input layer and a fully connected layer is set as the output layer.

To prevent overfitting, the corresponding Dropout layer is also added, and the threshold value is set to 0.1. The activation function selected in this paper is "RELU", and "ADAM" is used as the optimizer, "MSE" as the loss function, and "MAPE" as the performance evaluation function [4, 9, 12]. Only the model weights and the optimal model are saved during the model training, while the detection value criterion is set to the average error, the mode is set to the minimum value, batch size is 32 and the epoch is 50.

5.2 Model Evaluation

During the experiments, the data set was divided into test and training sets in the ratio of 4:6.

The first test indicator is the coefficient of determination (R^2), the proportion of the total variation of the dependent variable that can be explained by the independent variable through the regression relationship. The larger the R^2 value is, the better the explanation of the model is. the formula for calculating R^2 is shown in equation 5).

$$R^2 = 1 - \frac{\sum_{i=1}^{m} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{m} (y_i - \overline{y_i})^2} \tag{5}$$

In equation 5), y_i is the true value of sample i, $\hat{y}_i$ is the predicted value of sample i, and m is the number of samples.

The second test indicator is accuracy deviation (AD), the weighted average of the ratio of the absolute value of the difference between the predicted value and the real value, is used to evaluate the deviation degree between the predicted value and the real value. The smaller the value of AD is, the more balanced and stable the distribution of results predicted by the model is. The calculation formula of AD is shown in Equation 6)

$$D = \frac{\sum_{i=1}^{n} \frac{|preY_i - Y_i|}{Y_i}}{n} \tag{6}$$

In equation 5), Y_i is the true value and $preY_i$ is the predicted value of the model.

The third test indicator is overall deviation (OD), the weighted average of the difference between the predicted value and the real value and the ratio of the real value, is used to evaluate the fitting degree of the predicted whole and the real whole. The closer the OD value is to 0, the more accurate the model is. The OD calculation formula is shown in Formula (7).

$$D = \frac{\sum_{i=1}^{n} \frac{preY_i - Y_i}{Y_i}}{n} \tag{7}$$

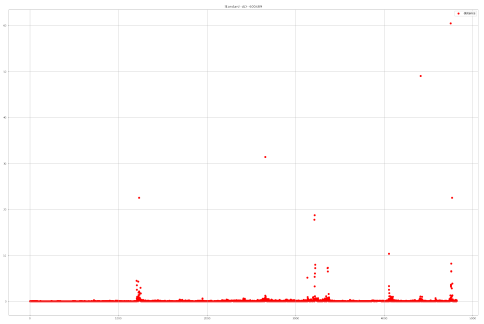

Figure 9: 600489 AD values at each point after fitting the latest price to the 600108 model.

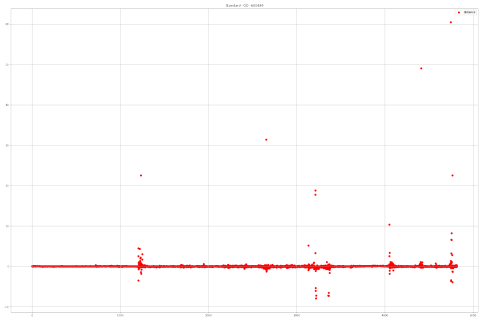

Figure 10: 600489 OD at each point after fitting the latest price to the 600108 model.

The four clusters with the largest number of members are selected for comparative analysis. As shown in Figure 7, the centroids are 600108, 600120, 601163, and 600496. The model they trained to predict the latest price is used to fit the latest price of the stock with code 600489 (belonging to the cluster with stock code 600108 as centroid), as shown in Table 2. It can be seen that the model for 600108 has the highest explanation, which indicates that the inclusion of K-means is a correct approach.

Figure 9 and Figure 10 show the valuation distribution of AD and OD reviews for stock 600489. The x-axis is the index value of the trading time points of stock 600489 in February 2021 time-series alignment(same in Figure 11), and the y-axis is the corresponding valuation value. It can be seen that both the AD point distribution and the OD point distribution are concentrated in the horizontal line with a value of 0. This indicates that the results of the model fit are stable and accurate.

Table 2: R^2-values for different models fitted to 600489

Model Code	R2
600108	0.9964551220445029
600120	0.9842140712094754
601163	0.98482835043915
600496	0.801106297932982

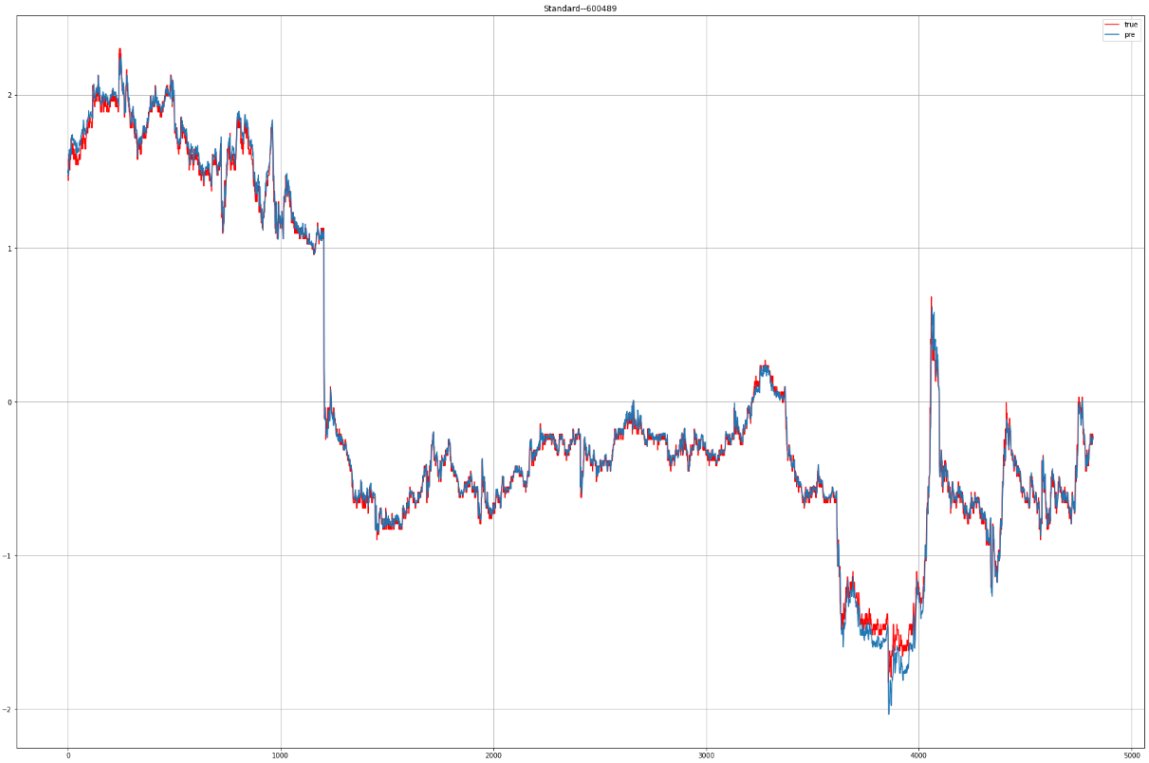

Figure 11: Fit of model 600108 to the latest price of 600489.

6 USING MODELS FOR PREDICTION

After reducing the predicted value to the true value by the inverse normalization formula, which is shown in equation 8), it is compared with the true value, and the result is shown in Figure 11. The Y-axis is the value of the latest price. In which the red line is the true value and the blue line is the predicted value.

$$x_i = x_s \times x_{std} + x_{mean} \tag{8}$$

The model successfully fits abrupt change points as well as sustained upticks, and still maintains a fairly close trend of change in areas where accuracy is poor. In the figure all trading times are manually stitched together, so that when on different trading days, the data predicted based on the last trading time point of that trading day is the first trading date of the next trading day.

7 CONCLUSION

Using the February 2021 A-share time-of-day K-line as experimental data, combining the K-means clustering algorithm and LSTM neural network can reduce the difficulty of making batch predictions for stocks. The following conclusions are obtained.

- By placing different weights on each time-line data of a stock according to its distance from the mean, the weighted average result obtained can effectively represent the true level of this data set.
- K-means is a label-free classification algorithm that uses a high "cohesion" and low "coupling" judging rule to obtain the best number of classifications.
- By finding the commonality among different stocks to achieve reusability of the model, the prediction results are evaluated by R^2, AD, and OD, and it is found that the model can still maintain high stability and accuracy, which can save resources such as server and time consumption.

In this report, only the clusters with a large number of members were selected for verification. In subsequent studies, each one should be verified. In a persistent run, when a brand new stock

is added to the forecasting system, it is hard to determine which parameters in the existing model the new model can refer to because it does not have a certain amount of historical data. Due to the stock is constantly changing, the prediction model needs to constantly evolve itself. Meanwhile, although the members of a cluster are very similar, there are still differences between them. Therefore, independent differentiation is necessary to ensure the accuracy of prediction in the future.

REFERENCES

[1] D. Wei, "Prediction of Stock Price Based on LSTM Neural Network", 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (ALAM), pp. 544-547, 2019.

[2] Fang Yichou, Lu Zhuang, Ge Junwei. Joint RMSE loss LSTM-CNN model for stock price prediction[J]. Computer Engineering and Applications, 2022,58(09):294-302.

[3] I. Parmar *et al.*, "Stock Market Prediction Using Machine Learning", 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), pp. 574-576, 2018.

[4] K.M. Inumula, "Application of optimized technical indicators: MACD and RSI", Paripex-Indian Journal of Res., pp. 636-640, 2017.

[5] Lin X, Zhu S. Attention mechanism-based LSTM stock price prediction model[J]. Journal of Chongqing University of Technology and Business (Natural Science Edition), 2022,39(02):75-82.DOI:10.16055/j.issn.1672-058X.2022.0002.011.

[6] Peng Y, Liu YH, Zhang RF. Modeling and analysis of stock price prediction based on LSTM[J]. Computer Engineering and Applications, 2019,55(11):209-212.

[7] P. Piravechsakul, T. Kasetkasem, S. Marukatat and I. Kumazawa, "Combining Technical Indicators and Deep Learning by using LSTM Stock Price Predictor," 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2021, pp. 1155-1158, doi: 10.1109/ECTI-CON51831.2021.9454877.

[8] Priya, R.S., Sruthi, C. (2022). Stock Price Prediction Based on Deep Learning Using Long Short-Term Memory. In: Sivasubramanian, A., Shastry, P.N., Hong, P.C. (eds) Futuristic Communication and Network Technologies. Lecture Notes in Electrical Engineering, vol 792. Springer, Singapore.

[9] R. Vaidya, "Moving Average Convergence-Divergence(MACD) Trading Rule: An Application in Nepalese Stock Market", Quantitative Economics and Management Studies, pp. 366-374, 2020, [online] Available: https://doi.org/10.35877/454RI.qems197.

[10] Sarkar A, Sahoo AK, Sah S, Pradhan C (13–14 Mar 2020) "LSTMSA: a novel approach for stock market prediction using lstm and sentiment analysis". International conference on computer science, engineering and applications (ICCSEA), India.

[11] Silva TR, Li AW, Pamplona EO (19–24 Jul 2020) "Automated trading system for stock index using LSTM neural networks and risk management". International joint conference on neural networks (IJCNN), United Kingdom.

[12] S. Nisarg and P. Manubhai, "A Comparative Study on Technical Analysis by Bollinger Band and RSI", Proc. IJMSS Conf, vol. 3, Jun. 2015.

[13] Wu, Y., Yang, H., Zhou, K., Wang, Y., Zhu, Y. (2022). Application of Bidirectional LSTM Neural Network in Grain Stack Temperature Prediction. In: Pan, L., Cui, Z., Cai, J., Li, L. (eds) Bio-Inspired Computing: Theories and Applications. BIC-TA 2021. Communications in Computer and Information Science, vol 1566. Springer, Singapore.

[14] Yu Qiongfang, Niu Dongyang. Mine pressure space-time hybrid prediction based on LSTM networks [J/OL]. Electronic science and technology: 1-7 [2022-05-22]. DOI: 10.16180 / j.carol carroll nki issn1007-7820.2023.02.010.