Unveiling the Predictive Power: Comparative Analysis of Cutting-Edge Deep Learning Models for Stock Price Forecasting

Minghao Guan [1, *, #], Yuanjin Zhu [2, #], Bo Xiao [3, #]

[1] School of mathematics and statistics, Central South University, Changsha, China, 410012

[2] School of Finance, Central University of Finance and Economics, Beijing, China, 102206

[3] School of Business, Ningbo University, Ningbo, China, 315211

* Corresponding Author Email: 8201210506@csu.edu.cn

[#]these authors contributed equally.

Abstract. Accurate stock price prediction plays a fundamental role in informing government financial regulations and facilitating effective arbitrage strategies for investors. With the application of deep learning algorithms in finance, significant progress has been made to improve the accuracy of stock price prediction. In this paper, first, we collected stock price data from four listed companies from different sectors. Then, we used four competitive methods for prediction, namely LSTM, GRU-LSTM, Attention-LSTM and Transformer-LSTM. The validity of the study is supported by multiple sets of comparative experiments. Our experimental results show that LSTM shows superiority in predicting stock prices, while Transformer-LSTM model has better generalization ability.

Keywords: Stock price prediction; LSTM, GRU; Attention; Transformer.

1. Introduction

The forecast of stock prices has always been an important yet challenging task due to the volatility in financial markets [1]. The past century has witnessed the emergence of various methodologies of forecasting stock price, including technical analysis, fundamental analysis, quantitative analysis, and sentimental analysis [2]. Currently, as time series prediction algorithms continue to improve, stock price prediction using deep learning algorithms plays an increasing critical role in quantitative trading of stocks. It's highly necessary to improve the accuracy of stock price prediction through advanced mathematical models for the following reasons. The stock price serves as a pivotal signal to guide the appropriate flow of capital in the securities market, as well as providing a practical basis for the government's regulation of financial markets and the arbitrage activities of investors. In consequence, there is a need for academia to identify feasible prediction methods using deep learning and to obtain the optimal model by comparing them. The main contribution of this paper lies in figuring out the most competitive prediction model through comparative experiments, which also offers a useful reference for the selection of specific methods to determine entry and exit points in quantitative trading. More importantly, the conclusions of this study can provide insights into the development direction of financial forecasting.

In this paper, we divided the data from different stocks into three groups and utilized four distinct methods based on deep learning to make predictions. We then compared the results to actual data in order to evaluate the effectiveness of each model. The four stocks selected are from Apple Inc., Johnson & Johnson, Tesla Inc., and Procter & Gamble Co. respectively. Apple Inc. is a high-tech company renowned for innovation. Johnson & Johnson is the world's largest healthcare company for diversified medical health and care products. Tesla Inc. is an electric car company known for its environmentally friendly philosophy. Procter & Gamble is currently one of the world's largest daily necessities companies. The stocks selected in this article have good representativeness and are suitable for research on industry heterogeneity as the companies aforementioned are industry leaders and demonstrate different styles in terms of stock price trends.

This paper consists of two parts, literature review and empirical analysis. In the literature review part, we summarized relevant studies on the application of deep learning in stock price prediction, in order to determine the mathematical model for this experiment. In the empirical analysis part, we compared the prediction accuracy of the stock trend under different models and then selected the most suitable method. We also compared the performance of these models on stocks from different industries to enhance the reliability of the experimental results.

2. Literature Review

With the application and development of artificial intelligence in speech and image recognition, deep learning algorithms have made remarkable progress. Leveraging its advantages in achieving high prediction accuracy, deep-learning-based analysis using historical data of stocks have also emerged as a breakthrough innovation in the domain of financial forecasting. One common choice is the RNN family, such as RNN, LSTM and GRU. Zhang et al. utilized LSTM networks to predict stock trends [3]. LSTM networks are more appropriate for dealing with complex financial time series due to the nonlinear and non-stationary nature of stock price data, as opposed to other artificial neural networks. Li et al. selected 18 stocks in the Shanghai Stock Exchange, and used a GRU recurrent neural network to predict the closing price for the next 10 days [4]. Empirical results show that compared to the other two models, GRU has smaller testing and validation errors, which reflects the strong learning and generalization ability of GRU.

To further improve the model aforementioned, a considerable body of literature has introduced Attention. Zhao et al. proposed a hybrid model (LSTM-CNN-CBAM), which incorporates the Convolutional Attention Block Module (CBAM) into the network structure [5]. Based on comparative experiments with the Shanghai Stock Exchange Index, it was found that the model proposed can effectively improve the ability of feature extraction and stock price prediction. In order to extract stock price trend features for prediction purposes, Lin et al. proposed an Attention-based LSTM model [6]. Based on a sample set of 42 Shanghai Stock Exchange 50 stocks, experimental results demonstrate that the model proposed exhibits superior predictive capabilities when compared to both traditional SVM models and single LSTM models. Qiu et al. developed a forecast model using LSTM and an Attention mechanism [7]. Their experiments on the S&P 500 and DJIA datasets indicated that their model performed better than three other models, including the LSTM model, the LSTM model with wavelet denoising, and the GRU neural network model. Li et al. established a novel Attention-based multi-input LSTM model, which incorporates extra input gates controlled by convincing factors to eliminate harmful noise [8]. The approach proposed is evaluated on stock data from the China stock market, and experimental results demonstrate its superiority over state-of-the-art methods. Lee designed a deep neural network model that combines a GRU-Attention mechanism [9]. The model exhibits high accuracy in forecasting essential stock price movements.

However, the main limitation of these techniques is that the RNN family faces challenges in capturing exceedingly long-term dependencies. In contrast, the Transformer model, a renowned sequence-to-sequence approach [10], has recently produced impressive results in natural machine translation tasks. Unlike RNN-based models, the Transformer leverages a multi-head self-Attention mechanism to grasp global positional relationships, thereby strengthening its ability to learn long-term dependencies. Fu et al. conducted several experiments to evaluate the efficacy of the Transformer model in predicting IBM's stock price trend [11]. Ding et al. introduced a new methodology utilizing the Transformer with several enhancements [1]. The experimental findings demonstrate that the model proposed surpasses several other competitive approaches, including RNN, in effectively capturing significant long-term dependencies present in financial time series data from both the NASDAQ exchange market and the China A-shares market. In summary, numerous scholars have made attempts to utilize deep learning methods in stock price prediction. However, rare articles have implemented controlled experiments to assess the strengths and weaknesses of these contemporary prediction methods.

3. Methods

3.1. Long Short-Term Memory (LSTM)

Firstly proposed by Hochreiter and Schmidhuber [12], LSTM neural network is a variant of traditional RNN. Compared with classical RNN, it can efficiently capture semantic associations between long sequences, and alleviate the phenomenon of gradient vanishing or explosion. Meanwhile, LSTM is more complex, and its core structure can be divided into four parts, including forgetting gate, input gate, cell state, output gate, etc., as shown in Fig. 1. Since LSTM is good at dealing with time series data prediction, this paper adopts LSTM model to predict stock price.

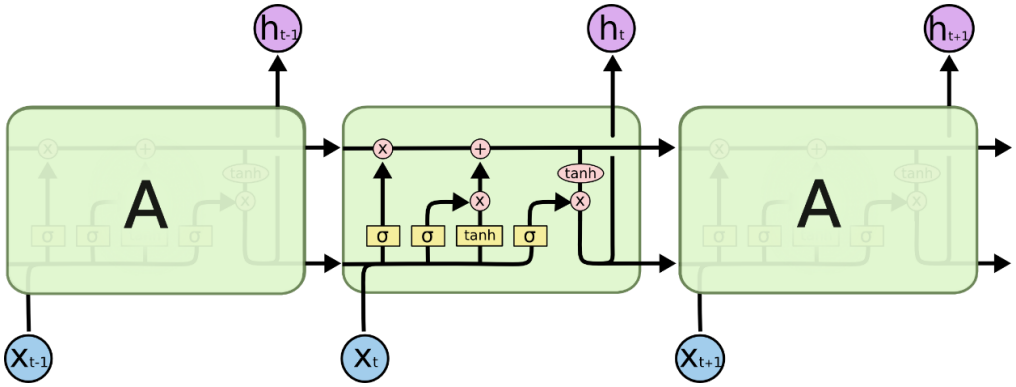

Fig 1. LSTM structure.

The general model of LSTM consists of four basic steps as follows:

(1) In the forgetting gate, the model first splices the input x_t at the current time step with the implied state h_{t-1} at the previous time step to obtain $[h_{t-1}, x_t]$, and then transforms it through a fully-connected layer, and ultimately activates it through a sigmoid function to obtain f_t, i.e., the gate value, which is the value of the gate that will be acted on the tensor passing through the gate. The value of the forgetting gate, which will be acted on the cellular state of the previous layer, represents how much information about the past has been oblivious to it.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

(2) There are two formulas for input gates. The first is the formula that produces the input gate value, which is quite similar to the forget gate formula, while the only difference is the target they will act on afterward. This formula implies how much of the input information needs to be filtered. The second formula for the input gate is the same as the internal structure of a traditional RNN. In the case of LSTM, it gets the current cell state instead of the implicit state as in RNN.

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

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

(3) During the training process, the model cell state needs to be constantly updated. There is no fully connected layer in the structure of the cell update. It is just the result of multiplying the forgotten gate value just obtained with the C_{t-1} obtained in the previous time step, plus the input gate value multiplied with the un-updated C_t obtained in the current time step. The updated C_t is finally obtained as part of the input for the next time step. The entire cell state update process is an application of the forgetting and input gates.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

(4) The final part of the output gate has two formulas. The first one is to calculate the gate value of the output gate, which is calculated in the same way as the forgetting gate and the input gate. The second one is to use this gate value to generate the implied state h_t, which will be applied to the updated cell state C_t, and activated by the tanh function, and finally get h_t as part of the input of the next time step. The whole process of output gating is to generate the implied state h_t.

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

$$h_t = o_t * \tanh(C_t) \tag{6}$$

3.2. Gated Recurrent Unit (GRU)

GRU (Gated Recurrent Unit) was firstly proposed in 2014 [13], which is also a variant of traditional RNN, and like LSTM, is able to effectively capture semantic associations between long sequences and alleviate the phenomenon of gradient vanishing or explosion. At the same time, its structure and computation are simpler than LSTM, which reduces the training parameters and guarantees the prediction accuracy. Its core structure can be divided into two parts, including update gate and reset gate. The update gate controls the extent to which state information from the previous moment is retained in the current state, with larger values indicating that more state information from the previous moment is retained. The reset gate controls the extent to which the current state is combined with previous information, with smaller values indicating more information is ignored.

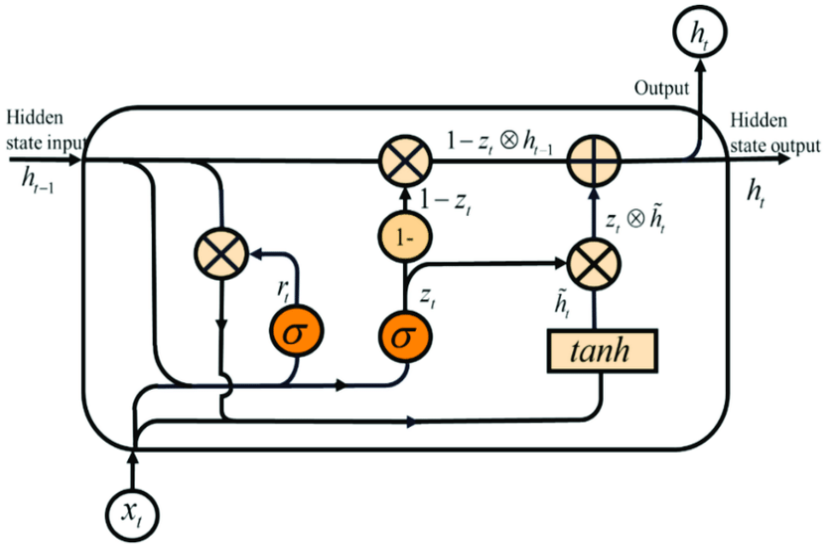

Fig 2. GRU structure.

The general model of GRU consists of four basic steps as follows:

(1) As with gating in LSTMs previously analyzed, the gate values for the update and reset gates are first calculated as z_t and r_t respectively, which are shown in (7) and (8). The update gate determines how much past information can be continued into the future. The information from the previous and current moments are linearly transformed, i.e., the weight matrices are right-multiplied separately, and the summed data is fed into the update gate, i.e., it is multiplied with a sigmoid function, which yields a value between [0, 1].

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

(2) The reset gate r_t determines how much historical information cannot be passed on to the next moment. As with the data processing of the update gate, the information from the previous and current moments are linearly transformed, i.e., the weight matrices are right-multiplied separately, and then

the summed data is fed into the reset gate, i.e., multiplied with a sigmoid function, which yields a value between [0, 1]. It's just that the values and usefulness of the weight matrices are different in the two times.

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

(3) Next, the memory information is reset utilizing a reset gate. The GRU no longer uses separate memory cells to store the memory information, but instead records the historical state directly utilizing hidden cells. The reset gate is utilized to control the amount of data for the current information and the memory information, and new memory information is generated to continue forward. As in Equation (9), since the output of the reset gate is in the interval [0, 1], the reset gate is utilized to control the amount of data that the memory information can continue to be passed forward, when the reset gate is 0 it means that the memory information is cleared in its entirety, and vice versa, when the reset gate is 1, it means that the memory information passes through in its entirety.

$$\tilde{h}_t = tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{9}$$

(4) Finally, the output of the hidden state at the current moment is computed using the update gate. The output information of the hidden state consists of the hidden state information h_{t-1} of the previous moment and the hidden state output h_t of the current moment, and the update gate is utilized to control the amount of data that these two pieces of information are passed to the future, which is shown in (10).

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{10}$$

3.3. Attention

The attention mechanism was originally applied in the field of computer vision and later found its significance in NLP. It gained substantial attention in the NLP community when models like BERT and GPT achieved remarkable performance in 2018. As a result, core components such as Transformer and Attention started to receive significant focus.

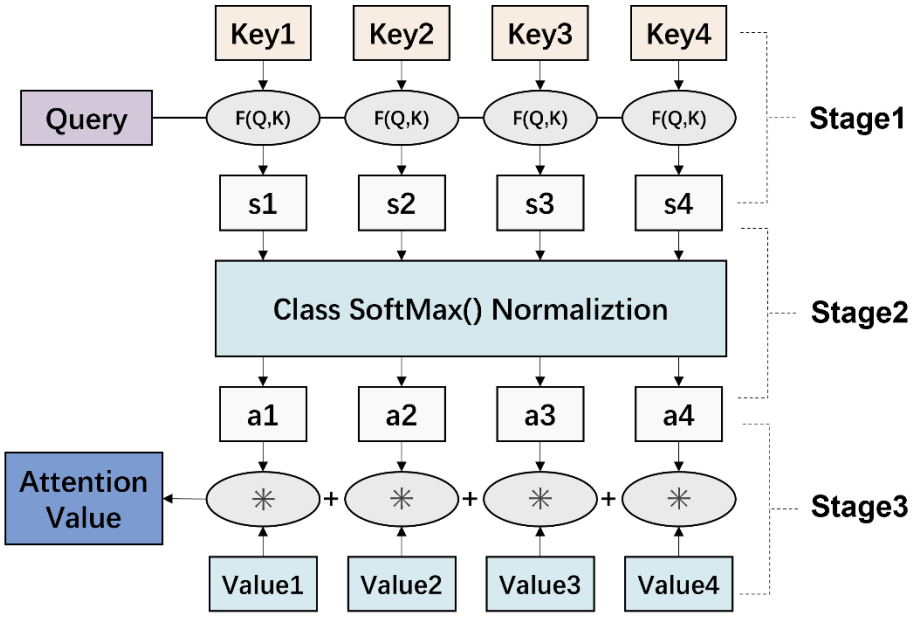

Fig 3. Calculation process of the attention mechanism.

The calculation process of the attention mechanism is illustrated in Fig. 3.

In the first stage, the similarity between a query and a specific key is calculated. Common methods include taking the dot product of their vectors, calculating the cosine similarity of their vectors, or

employing additional neural networks for evaluation. In this study, we adopt the approach of first calculating the dot product of the two vectors, followed by applying the tanh activation function to obtain the similarity:

$$Sim_i = tanh(Query \cdot Key_i^T) \tag{11}$$

In the second stage, a calculation method Softmax normalization is introduced to transform the scores obtained in the first stage. On one hand, it allows normalization by organizing the original score calculations into a probability distribution where the sum of all element weights is equal to 1. On the other hand, it emphasizes the weights of important elements through the intrinsic mechanism of SoftMax. The general formula commonly used for calculation is as follows:

$$a_i = Softmax(Sim_i) = \frac{e^{Sim_i}}{\sum_{j=1}^{L_x} e^{Sim_j}} \tag{12}$$

The result of the second stage of the calculation a_i, represents the weighting coefficients corresponding to each value. Attention values are obtained by weighted summation:

$$Attention(Query, Source) = \sum_{i=1}^{L_x} a_i \cdot Value_i \tag{13}$$

By performing the three stages of calculation above, the attention value for a given query can be obtained.

3.4. Transformer

The study by Ding et al. emphasizes that Transformer-based models are more effective in meeting the challenges of financial time series forecasting. This is because Transformer models excel at capturing the long-term complex structure inherent in financial time series data [14]. Compared to RNN models, Transformer models enhance the ability to learn long-term dependencies by employing a multi-head self-attention mechanism. Therefore, we chose to use the Transformer model for our experiments.

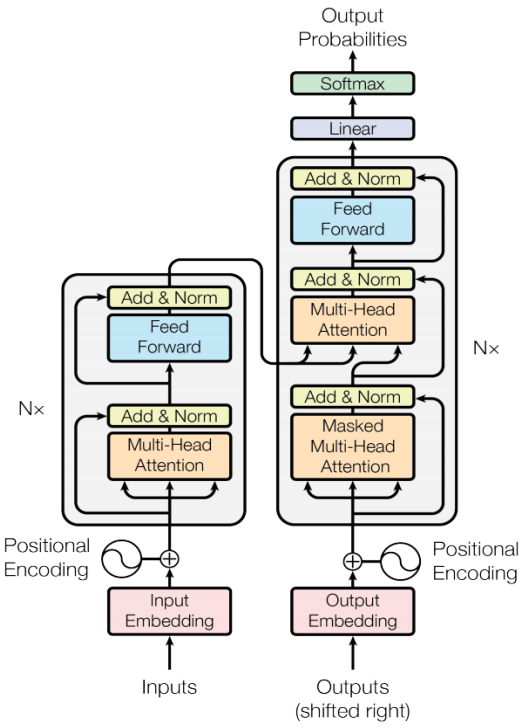

Fig 4. The Transformer - model architecture.

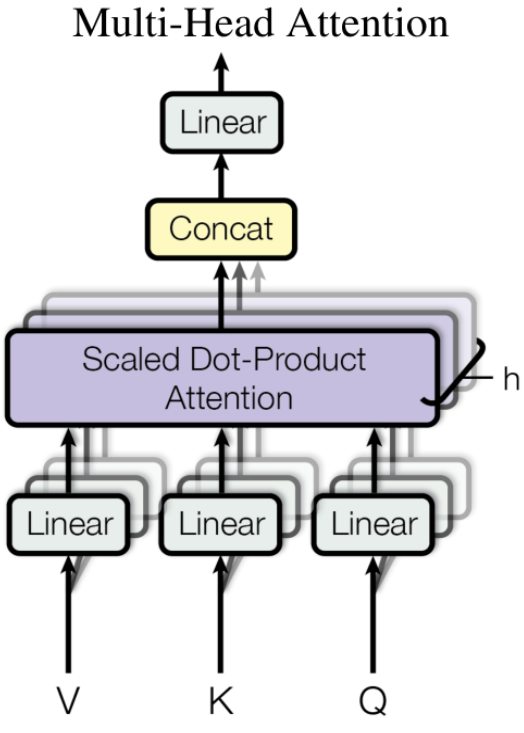

Fig 5. Multi-Head Attention.

Fig. 4 illustrates the structure of the Transformer model applied in language translation.

Transformer is an Encoder-Decoder architecture. The Encoder consists of 6 layers, each composed of two sub-layers: 1. Multi-Head Attention, which comprises multiple Self-Attention mechanisms; 2. Feed Forward layer. Each sub-layer is augmented with an Add&Norm layer.

The Decoder also has 6 layers, with each layer containing three sub-layers: 1. Masked Multi-Head Attention layer, used for Self-Attention. Unlike the Encoder, the Decoder operates in a sequence generation process, where at time t and beyond, no results are available. Only time steps smaller than t have results. Therefore, a Mask is applied. 2. Multi-Head Attention, used to calculate the attention between Encoder and Decoder, different from Self-Attention; 3. Feed Forward layer. Similar to the Encoder, each sub-layer in the Decoder is followed by an Add&Norm layer.

(1) Multi-Head Attention

Multi-Head Attention is essentially composed of multiple independent and parallel Attention mechanisms, allowing us to capture related information across different subspaces. This is depicted in Fig. 5.

(2) Input in Transformer

The basic Transformer model requires encoding words through Embedding, converting them into word embedding vectors. However, since our inputs are time series data instead of words, we do not need the Input Embedding and Output Embedding steps.

4. Experiments

4.1. Data preparation

We have extracted the adjusted stock price data of Tesla, Procter & Gamble, Apple and Johnson & Johnson from their respective listing dates until July 19, 2023. The dataset includes opening price, highest price, lowest price, closing price, and volume data. There are a total of 3285, 10921, 10731 and 10917 valid data points for the four companies respectively. We have selected the closing price as the prediction indicator, and the visualizations of the results are shown below.

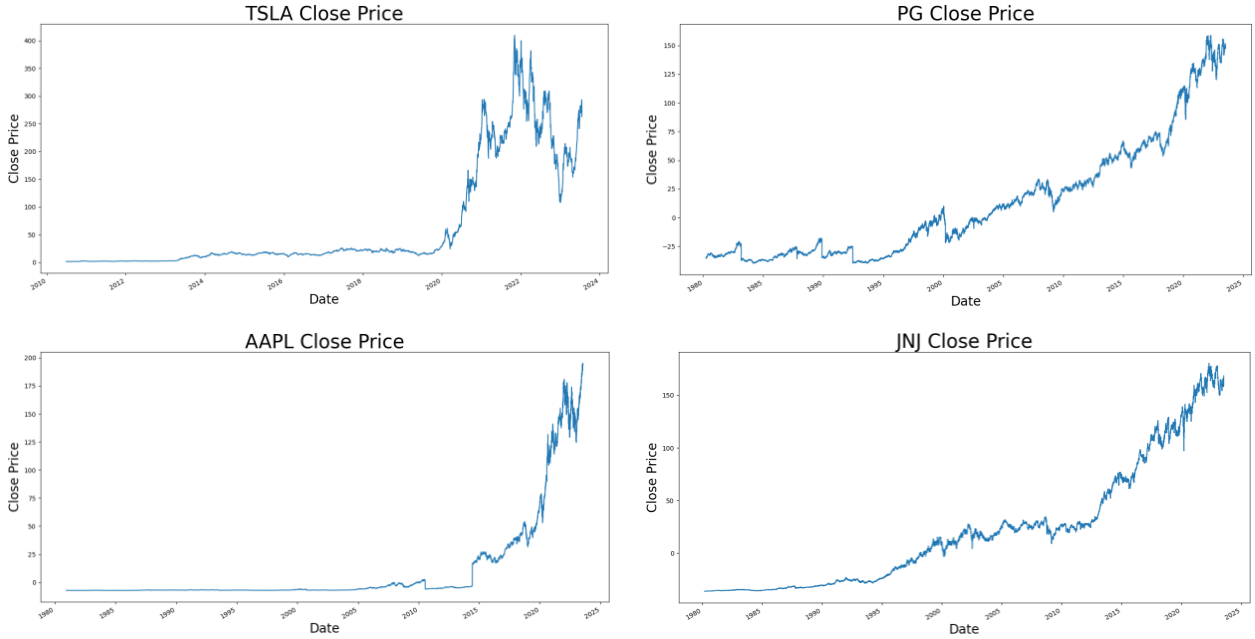

Fig 6. Split-adjusted close price data of four stocks.

4.2. Basic analysis

According to the TSLA Close Price in Fig. 6, it can be seen that from 2010 to early 2019, Tesla's stock price remained relatively stable and failed to attract widespread attention. However, from early 2019 to early 2020, Tesla released sales data for the Model 3 that exceeded expectations and achieved

consecutive quarterly profits, triggering market attention and causing the stock price to rise rapidly. After 2020, Tesla announced successful entry into the Chinese market and achieved profitability, attracting more investors and leading to a continuous increase in the stock price. From mid-2022 to early 2023, Tesla's stock price experienced a certain degree of correction due to the drag of global economic recovery uncertainties and supply chain issues.

According to the PG Close Price Fig. 6, it can be seen that during its early listing period until the late 1990s, Procter & Gamble (P&G) experienced relatively stable stock performance with a slight upward trend. The stable revenue and market position in the daily consumer goods industry provided support for the stock price. In the early 2000s, P&G's stock price began to rise due to a series of strategic initiatives that enhanced the company's profitability and market share, thus driving the stock price upward. In 2008, during the global economic downturn and a decline in consumer confidence, P&G experienced a significant drop in sales, leading to a substantial decline in its stock price during the financial crisis. After 2010, P&G's stock price entered a relatively stable growth phase. During the global outbreak of the COVID-19 pandemic, P&G's stock price once again saw an increase. This was attributed to the increased demand for essential everyday products as consumers prioritized these items, providing strong support for P&G as a leading consumer goods company.

According to the AAPL Close Price Fig. 6, Apple’s stock price has gone through several phases since it went public in 1980. Before 2000, Apple's stock price did not rise or fall much and was in a relatively stable state. From 2001 to 2007, Apple launched products such as iPod and iPhone, and its performance and stock price grew rapidly. After the financial crisis in 2008, the stock price fluctuated considerably, but the overall trend was upward. The stock price has been on an upward trend in general. However, Apple's share price began to fall in 2012, and its market value evaporated by tens of billions of dollars. After that, Apple launched new products, such as iPad, Apple Watch, etc., and the stock price gradually rebounded. In 2019, Apple's stock price fell again due to the trade war and other factors. Apple's market capitalization is largely affected by the product launch situation. By exploring the many possibilities of scene interaction and maximizing the product experience, Apple has been able to lead the industry change and define people's lifestyles by taking a backseat.

According to the JNJ Close Price Fig. 6, as the world's most famous healthcare and consumer care Products Company, Johnson & Johnson's stock price has been rising steadily for nearly a century. Overall, Johnson & Johnson's valuation is low and its performance has remained strong. Two acquisitions in 1959 and 1961 established Johnson & Johnson's brand strength in prescription drug development. After the 1980s, Johnson & Johnson moved further into consumer healthcare by acquiring pharmaceutical companies that produced products including glucose meters and mouthwashes. Mergers and acquisitions have given Johnson & Johnson a diversified portfolio, reduced the company's business risk, and increased its revenue streams. However, there have been times in the past few decades when negative press has led to a brief dip in the stock price. In December 2018, there was a relatively large drop in Johnson & Johnson's stock price due to a Reuters report on baby talcum powder containing asbestos, which can cause cancer, causing panic among consumers, and investors. Facing personal injury lawsuits from more than 100,000 plaintiffs, the lawsuit-plagued Johnson & Johnson stock price rose modestly for most of the period.

Next, we normalized and divided the data into three groups, train_data (70%), val_data (20%), and test_data (10%), and trained them in the four selected models.

4.3. Experiment analysis

4.3.1. LSTM

The yellow line in Fig. 7 shows the results of 20 epochs using the LSTM model, and the blue line is the original data. The training result images above shows that for the three stocks TLSA, PG and JNJ, the LSTM_20 prediction result is more accurate, for the AAPL stock, the prediction value is quite precise at the beginning, but then the deviation gradually becomes bigger. Although the fluctuation trend is basically the same as the real value, it is always higher than the real value.

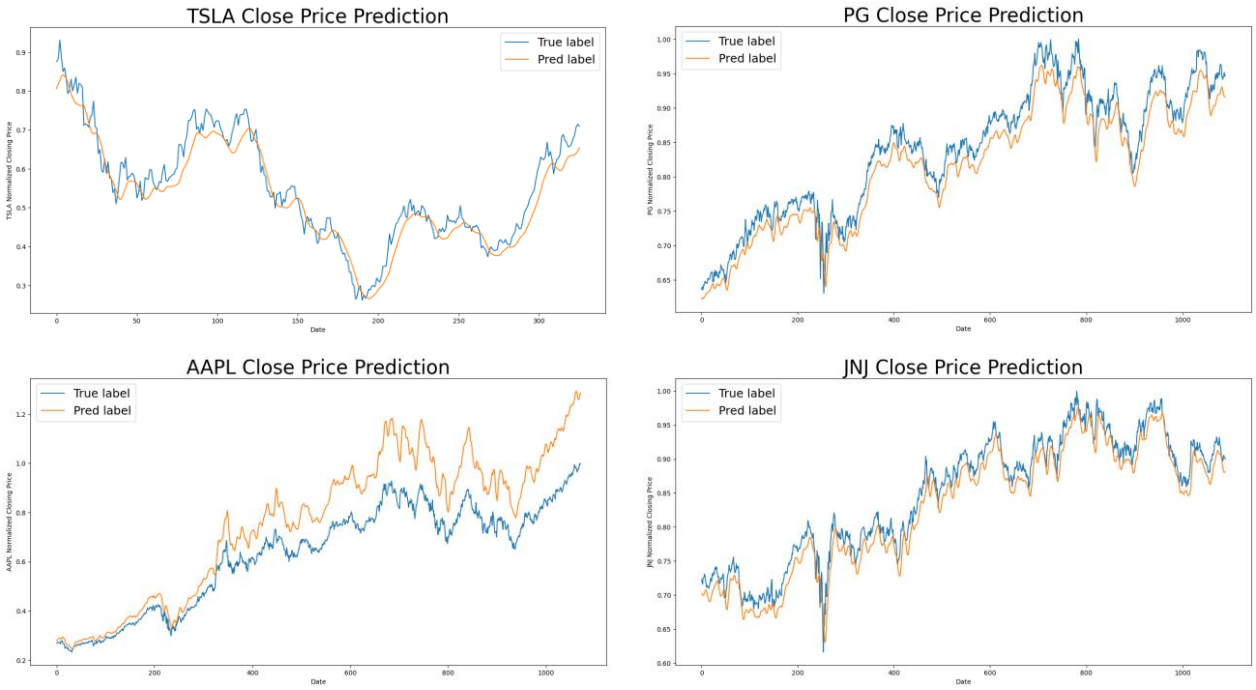

Fig 7. Four stocks' close price prediction using the LSTM_20.

Since the prediction was good, we tried to train the LSTM model further by setting the epoch to 100, and the results are shown below.

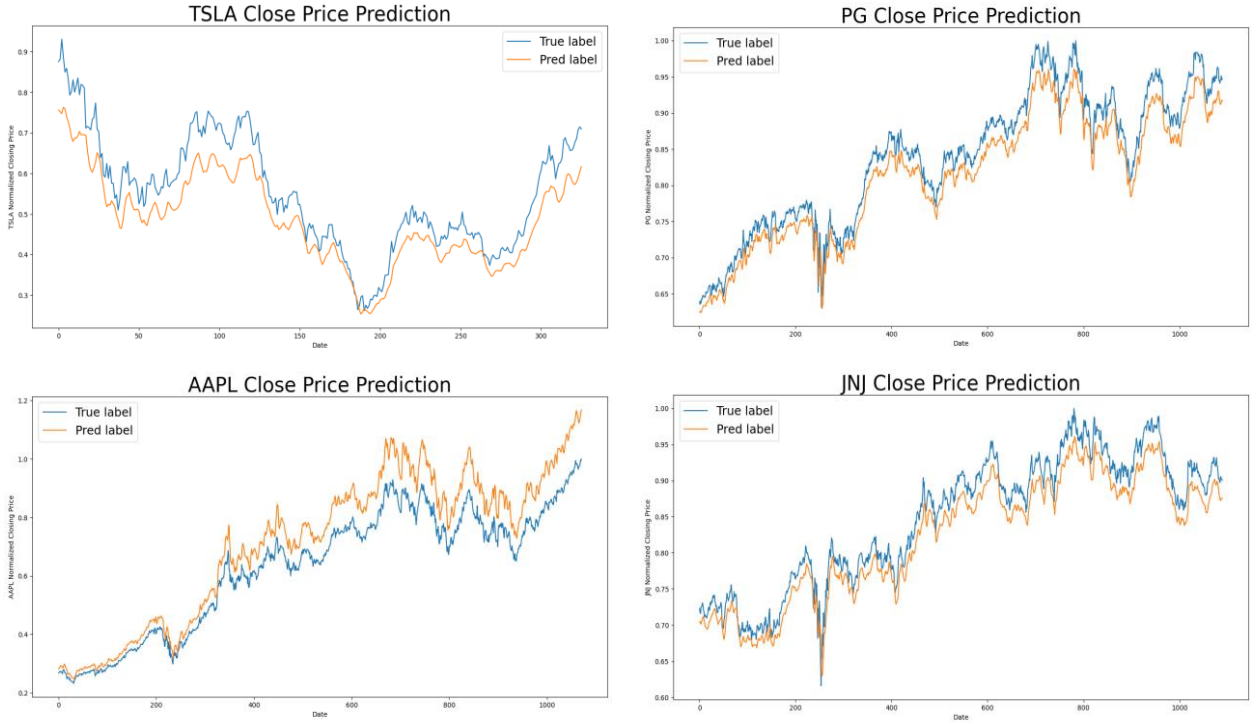

Fig 8. Four stocks' close price prediction using the LSTM_100.

However, when the number of Epoch of LSTM was changed from 20 to 100, we found that the val loss fluctuated more drastically for the four stocks. Among them, there is an increase in the val loss fluctuation of TSLA, which may be due to the overfitting problem of the model.

As shown in Fig. 8, for the three stocks TLSA, PG, JNJ, the prediction of LSTM_100 is not as good as that of LSTM_20. For the AAPL stock, the fit of LSTM_100 is better, but it still has the same problem as the prediction of LSTM_20. In addition, we find that the predicted value of AAPL is always larger than the true value, contrary to the other three stocks.

4.3.2. GRU-LSTM

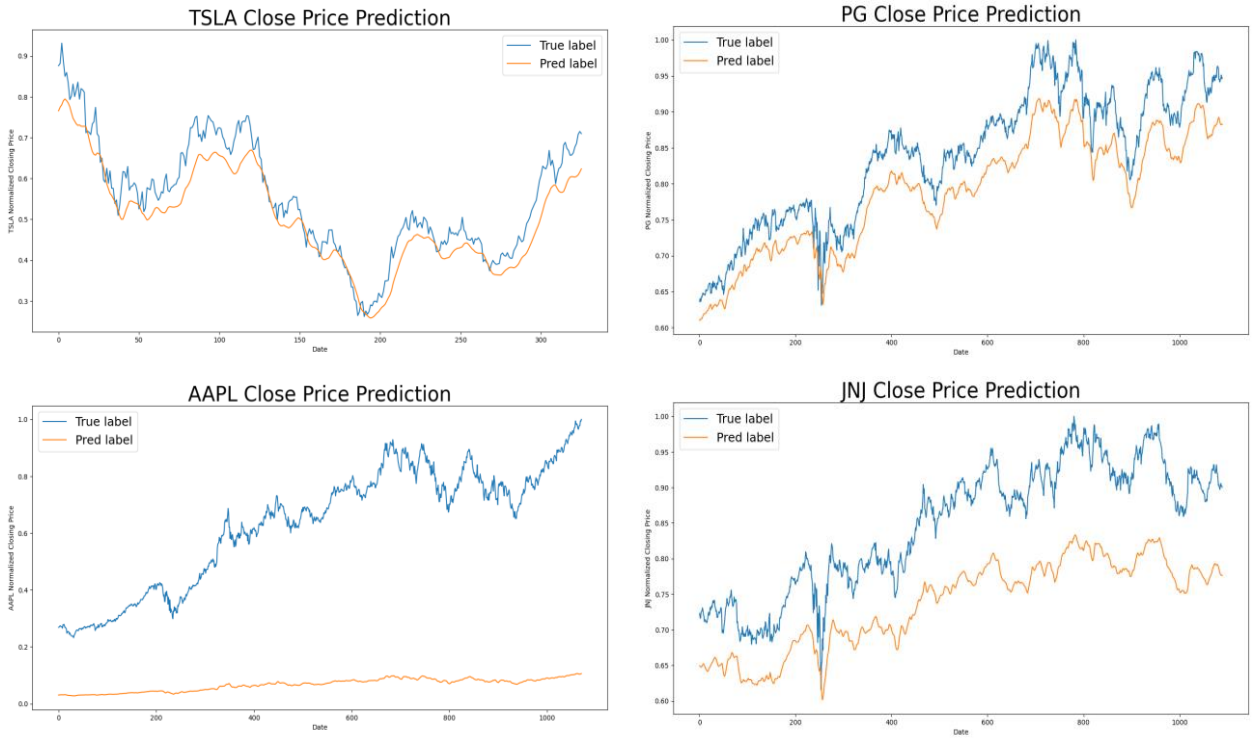

Fig 9. Four stocks' close price prediction using the GRU_LSTM.

However, the training effect of this model in Fig. 9 is not ideal, and the prediction of AAPL and JNJ has a very large deviation. Overall, it seems that the GRU-LSTM model is only good for the prediction of TSLA, and the prediction of the other three stocks is obviously inferior to the LSTM model and does not realize the effect that we want.

4.3.3. Attention-LSTM

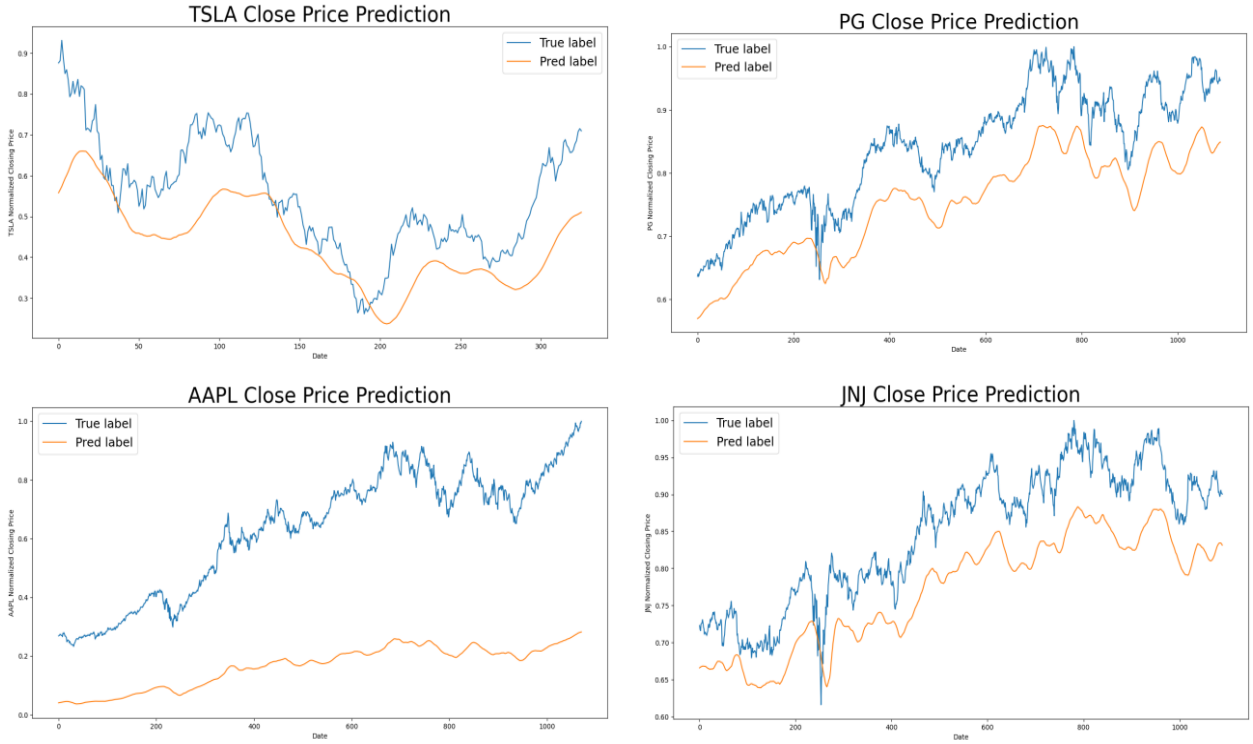

Fig 10. Four stocks' close price prediction using the A_LSTM.

We continue to try to predict stock prices using the Attention-LSTM model with 20 epochs. The training results in Fig. 10 are a little similar to those of the GRU-LSTM model, with generally poorer fitting results. It can be seen that although the Attention model has had great success in areas such as natural language processing, there are still difficulties in applying it to stock price prediction.

4.3.4 Transformer-LSTM

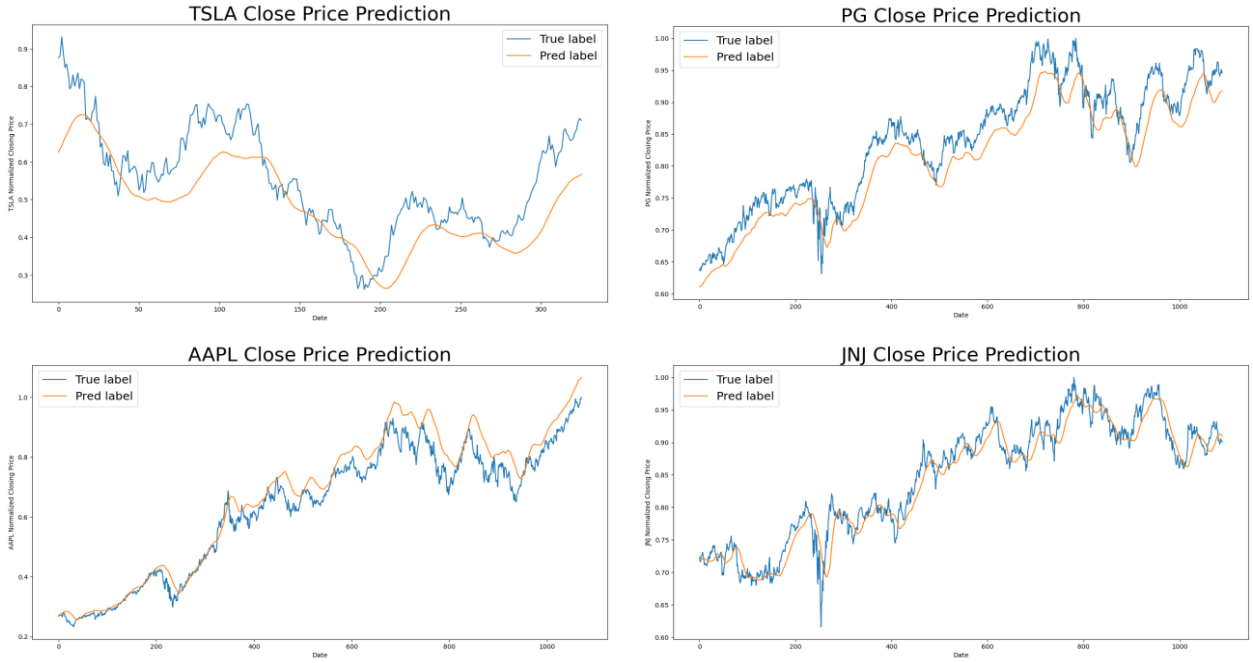

Fig 11. Four stocks' close price prediction using the Trans_LSTM.

Fig. 11 shows the prediction results for the four stocks when we use the Transformer-LSTM model with 20 epochs. The prediction results show that PG, AAPL and JNJ have better fitting results, while the predicted series of TSLA seems to have some lags and is slightly lower than the true value. In addition, we can also observe that the result curves predicted by the Transformer-LSTM model are smoother, which is obviously a good shielding effect for the noise in the stock market, and we speculate that it may be due to the Multi-Head Attention layer in the Transformer-LSTM model plays a role.

4.4. Error analysis

Table 1. Error analysis table of PG.

Model	LSTM_20	LSTM_100	GRU_LSTM	A_LSTM	Trans_LSTM
R^2	0.9134	**0.9174**	0.6730	0.0298	0.8488
MSE	0.0007	**0.0007**	0.0027	0.0265	0.0012
RMSE	0.0272	**0.0265**	0.0528	0.1630	0.0359

Table 2. Error analysis table of TSLA.

Model	LSTM_20	LSTM_100	GRU_LSTM	A_LSTM	Trans_LSTM
R^2	**0.9179**	0.7375	0.8309	0.0684	0.5285
MSE	**0.0015**	0.0050	0.0032	0.0180	0.0091
RMSE	**0.0398**	0.0713	0.0572	0.1344	0.0956

Table 3. Error analysis table of AAPL.

Model	LSTM_20	LSTM_100	GRU_LSTM	A_LSTM	Trans_LSTM
R^2	0.4777	0.8047	-7.0275	-4.3778	**0.9292**
MSE	0.0229	0.0085	0.3521	0.3521	**0.0031**
RMSE	0.1513	0.0925	0.5934	0.5934	**0.0557**

Table 4. Error analysis table of JNJ.

Model	LSTM_20	LSTM_100	GRU_LSTM	A_LSTM	Trans_LSTM
R^2	**0.9380**	0.9131	-0.6495	0.1806	**0.9334**
MSE	**0.0004**	0.0006	0.0129	0.0129	**0.0005**
RMSE	**0.0220**	0.0261	0.1140	0.1140	**0.0228**

To further compare the advantages and disadvantages of each model, we calculated the R^2, MSE and RMSE of the training results of different models, and the results are shown in the tables above.

Apparently, PG, TSLA and JNJ are best fitted with the LSTM_20 model and AAPL is best fitted with the Trans_LSTM model. For both AAPL and JNJ, the R^2 of GRU_LSTM becomes negative, indicating that the nonlinear model does not capture any feature information, but only fits random noise, which is not suitable for fitting these two stock data. In addition, one of the R^2 of A_LSTM is negative and the other is close to 0, indicating that the model is also poorly fitted for both.

Taking into account the effectiveness of the fit, the generalization ability, and the compatibility for long and short time series, we believe that the Trans_LSTM model has wider adaptability, because although LSTM_20 performs well in stock price prediction for the three stocks, it is not good in predicting the stock price of AAPL with an R^2 lower than 0.5, whereas the Trans_LSTM model predicts the stock price of each stock with an R^2 higher than 0.5, which is more stable. And A_LSTM has the worst prediction for all four stocks, probably due to the fact that the attention mechanism of additive addition is not suitable for this kind of financial time series prediction and needs to be changed to other kinds.

5. Conclusions

In this paper, we utilized four models: LSTM, GRU-LSTM, Attention-LSTM, and Transformer-LSTM, to predict stock prices of publicly listed companies in four different industries with the aim of providing better assistance to investors in making investment decisions, asset allocation, and risk management. The experiment revealed that the LSTM model has good fitting ability, but it may suffer from overfitting, while the Transformer-LSTM model not only exhibits good fitting ability but also possesses good generalization ability. In addition, the fitting performance of the other two models was unsatisfactory. The findings of this study complement the gaps in previous researches by comparing the overall fitting performance of these commonly used models, as well as clarifying which industries each model is more suitable for predicting stock prices of relevant companies. Inevitably, our research has certain limitations, including possible noise in the data leading to decreased prediction accuracy, limited evaluation of models without exploring the combination of multiple models that may yield better performance, and potentially insufficient training which may reduce the universality of the findings. In the future, improvements can be made in the following aspects to enhance the accuracy and practical value of stock price prediction:

(1) Utilizing reliable fundamental data for multidimensional inputs.

(2) Exploring regularization methods to avoid overfitting of financial data.

(3) Finding more effective combinations of multiple models to optimize prediction performance.

References

[1] Ding, Q., Wu, S., Sun, H., Guo, J., & Guo, J. (2020). Hierarchical multi-scale Gaussian transformer for stock movement prediction. In IJCAI International Joint Conference on Artificial Intelligence, 2021-January, 4640-4646.

[2] Rathina, S. V., Vinayakumar, R., & Kayalvily, T. (2023). Multi-Lexicon Classification and Valence-Based Sentiment Analysis as Features for Deep Neural Stock Price Prediction. Sci, 5(1).

[3] Zhang, Y., Chu, G., & Shen, D. (2021). The role of investor attention in predicting stock prices: The long short-term memory networks perspective. Finance Research Letters, 38, Article 101484.

[4] Li, L., Chen, A. X., Li, W. S., et al. (2018). Research on prediction of stock closing price based on GRU recursive neural network. Computer and Modernization, (11), 103-108.

[5] Zhao, H., & Xue, L. (2021). Research on stock prediction based on LSTM-CNN-CBAM model. Computer Engineering and Applications, 57(03), 203-207.

[6] Lin, J., & Kang, H. (2020). Research on LSTM stock price trend prediction based on attention mechanism. Shanghai Management Science, 42(01), 109-115.

[7] Qiu, J., Wang, B., & Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. PLOS ONE, 15(1), e0227222.

[8] Li, H., Shen, Y., & Zhu, Y. (2018). Stock Price Prediction Using Attention-based Multi-Input LSTM. Proceedings of Machine Learning Research, 95, 454-469.

[9] Lee, M. C. (2022). Research on the Feasibility of Applying GRU and Attention Mechanism Combined with Technical Indicators in Stock Trading Strategies. Applied Sciences, 12(3), 1007.

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems, 5999-6009.

[11] Fu, S., Tang, Z., & Li, J. (2023). IBM Stock Forecast Using LSTM, GRU, Attention and Transformer Models. In 2023 IEEE International Conference on Control, Electronics and Computer Technology, ICCECT 2023, 167-172.

[12] Hochreiter S & Schmidhuber J. (1997). Long short-term memory. Neural computation (8). doi:10.1162/NECO.1997.9.8.1735.

[13] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho & Yoshua Bengio. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. CoRR.

[14] Fukui H., Hirakawa T., Yamashita T., & Fujiyoshi H. (2019). Attention Branch Network: Learning of Attention Mechanism for Visual Explanation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).