Proceedings of the 2nd International Conference on Financial Technology and Business Analysis
DOI: 10.54254/2754-1169/47/20230370

Stock Price Prediction During the COVID-19 Pandemic Based on the GRU Algorithm

Zirui Wu[1,a,*], Mingtao Huang[2,b]

[1]*School of Economics and Statics, Guangzhou University, 230 Zhonghuan West Road, Guangzhou, China*

a. 2064200170@e.gzhu.edu.cn, b. 2064200033@e.gzhu.edu.cn

**corresponding author*

Abstract: Stock price prediction during the Covid-19 pandemic has emerged as a significant research domain within the financial sector, giving rise to a multitude of neural network-based methods aimed at forecasting stock prices. This paper uses multiple machine learning models and analyzes the stock fund flows to attempt to learn and predict price trends from a dynamic perspective. The data used in this study includes the closing price, change rate, and fund flow of an A-share Pharmaceutical stock in the most recent 1000 trading days. The Mean Squared Error (MSE) is used as the model evaluation metric, and the model's MSE can reach 0.013 after training. The predictions generated by the model exhibit a high degree of alignment with the actual price trends, indicating its accuracy in short-term price trend prognostication. These findings substantiate the efficacy of the model for stock price prediction during the pandemic period, thereby contributing to the body of knowledge within the field of financial forecasting.

Keywords: GRU, RNN, stock price prediction

1. Introduction

The COVID-19 pandemic has exerted a pronounced influence on the A-share market, especially on companies involved in developing COVID-19 treatments. The volatility and uncertainty of the market during the pandemic has made it challenging for investors to make informed decisions.

Deep learning has experienced remarkable advancements in recent years, which have resulted in the emergence of several mainstream technologies [1-3], such as Gated Recurrent Units (GRU), Ordinary Backpropagation (BP) and Transformer. The Transformer model, a deep learning architecture built upon the self-attention mechanism, was first introduced by Vaswani et al. Since its inception, this model has garnered significant attention and found extensive applications within the domain of natural language processing [4]. Compared to traditional models, Transformer employs a novel mechanism to handle sequence data. Its self-attention mechanism can capture the relationships between different positions in the input sequence, enabling it to better process long-distance dependencies in sequences. Transformer is a deep learning model that has been widely adopted for its high performance in processing text sequences. However, when it comes to time series analysis, the performance of the Transformer model may not be as exceptional as other models that have been specifically designed for this task.

DOI: 10.54254/2754-1169/47/20230370

The use of GRUs for stock price prediction also has yielded impressive results, highlighting the model's efficacy in processing time series data, particularly in the domain of stock price prediction [5, 6]. However, the limitation of prior research is that their methods solely focused on stable periods where stock prices did not experience significant fluctuations, hence failing to fully demonstrate the advantages of the GRU. To address this limitation and to fully showcase the strengths of the GRU, many researchers have attempted to utilize Recurrent Neural Networks (RNNs) and GRUs to predict stock prices during unique periods such as the COVID-19 pandemic. Although their results were promising, their models did not account for the dynamics of price fluctuations.

To address this issue, this paper employs a Recurrent Neural Network (RNN) architecture [7, 8], incorporating GRU and examines the stock fund flows in order to acquire insights and forecast price trends within a dynamic context, specifically during a unique period. The dataset utilized in this investigation encompasses the closing price, change rate, and fund flow of a COVID-19 therapeutic drug stock within the A-share market over the preceding 1,000 trading days. In an effort to enhance the training efficacy and robustness of deep neural networks, the approach incorporates gradient clipping and weight normalization techniques in conjunction with GRUs.

2. Method

2.1. Data Set Preparation

The data used in this study is sourced from Sina Finance and was obtained using the request package to scrape the fund flow data of a stock of companies developing COVID-19 treatments for the past 1000 trading days.

2.2. RNN

For time series data shown in equation (1) [9], it can be assumed that the distribution of $X_t$as equation (2).

$$x_{t-1}, \dots, x_1 \tag{1}$$

$$P(x_t | x_{t-1}, \dots, x_1) \tag{2}$$

There are two effective strategies for estimating (2): The first strategy assumes that long sequences may not be necessary in real-world situations, so satisfying a time span of length τ is only needed, using the observation sequence$x_{t-1}, \dots, x_{t-\tau}$. The most obvious advantage of this is that training a deep network is feasible because the number of parameters is always constant, at least when $t > \tau$. As a result of performing regression on itself, this kind of model is known as an autoregressive model.

The second approach, as depicted in Figure 1, involves updating both the prediction $\hat{x}_t$ and the summary h_t while keeping some of the summary h_t of prior observations. This results in a model where $\hat{x}_t = P(x_t \mid h_t)$ is used to estimate x_t and $h_t = g(h_{t-1}, x_{t-1})$ is used to update the model. Such models are known as latent autoregressive models because h is never observed.

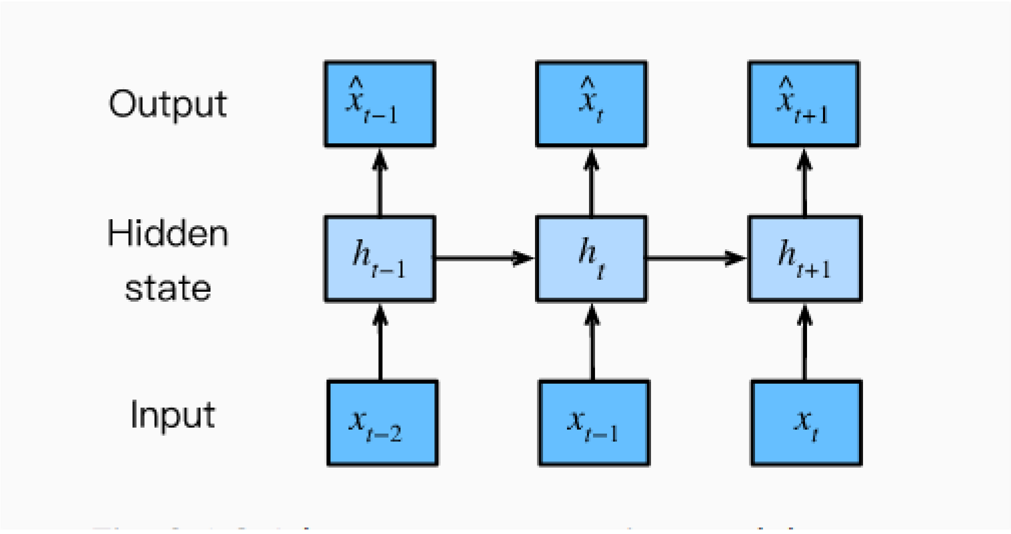

Figure 1: A latent autoregressive model [9].

A neural network with hidden states is called RNN. Based on the $X_t \in R^{n\times d}$ as input. The hidden variable can be represented by the variable $H_t \in R^{n\times h}$. In particular, the current time step's hidden variable is calculated using the input and the hidden value from the last time step:

$$H_t = \phi(X_t W_{xh} + H_{t-1} W_{hh} + b_h) \tag{3}$$

As with the present state of the network, the relationship between neighboring hidden variables H_t and H_{t-1} preserves the previous information of the sequence up to this time step. These concealed variables are hence referred to as hidden states. The calculation is recurring because the definition of the concealed state used at the present time step is the same as the definition used at the previous time step.

The output of the output layer is calculated similarly to a multilayer perceptron for time step t:

$$O_t = H_t W_{hq} + b_q \tag{4}$$

The weight $W_{xh} \in R^{d\times h}$ of the hidden layer, the weight $W_{hh} \in R^{h\times h}$ of the hidden layer, the bias $b_h \in R^{1\times h}$, the weight $W_{hq} \in R^{h\times q}$ of the output layer, and the bias $b_q \in R^{1\times h}$ are among the parameters of the RNN.

Considering the following for calculating the hidden state at any time step t: Concatenate the input X_t from the current time step with the hidden state H_{t-1} from time step t-1, and then insert the result into a completely connected layer with an activation function ϕ. The output of the completely linked layer is the hidden state H_t at the current time step t.

The calculating logic of a recurrent neural network for three consecutive time steps is shown in the following Figure 2:

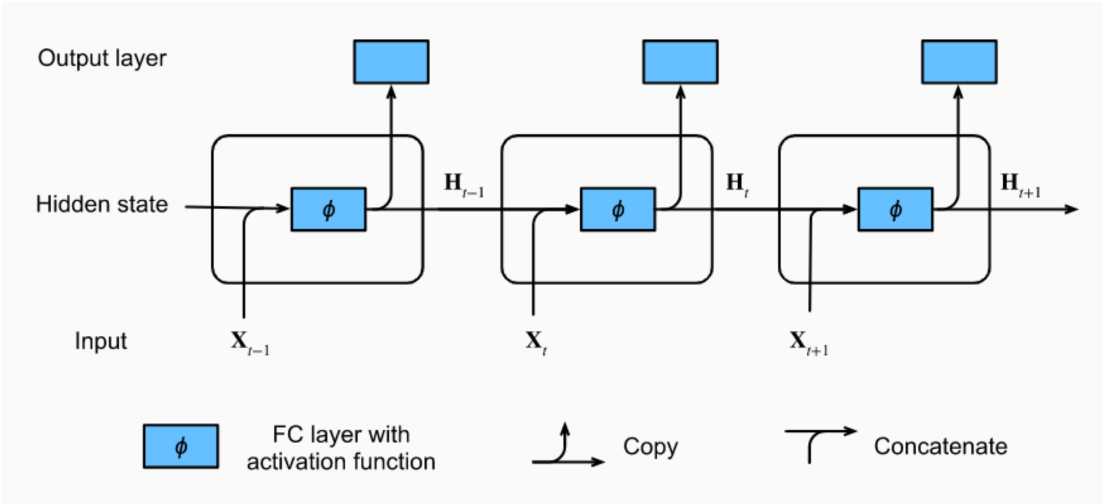

Figure 2: An RNN with a hidden state [9].

2.3. GRU

The GRU, is a type of RNN [10]. The gating module enhances its ability to process sequential data effectively, making it a valuable tool in deep learning research. These gating mechanisms enable the network to selectively forget or remember previous inputs, making the GRU suitable for tasks that require modeling temporal dependencies and long-term dependencies. The GRU consists of a set of recurrent units that update their hidden state based on the input at each time step. The two gating systems in charge of managing these units are an update gate and a reset gate. The reset gate and update gate are fundamental components within the GRU architecture. The reset gate determines the extent to which the previous hidden state should be disregarded, while the update gate governs the portion of the new input that should be incorporated into the current hidden state. The presence of gating mechanisms in the GRU enables it to selectively retain or discard information from preceding timesteps. Consequently, this characteristic renders the GRU well-suited for tasks involving the modeling of long-term dependencies in sequential data.

A conventional RNN and GRU differ primarily in that the former allows gated hidden states. This implies that the model includes a method to choose when to update and when to reset the hidden state. Reset gates and update gates are used by GRU to regulate the information flow in the recurrent network. The reset gate plays a crucial role in capturing short-term dependencies within a sequence, whereas the update gate is instrumental in capturing long-term dependencies within the same sequence. These two features, present in the architecture of the Gated Recurrent Unit (GRU), are of utmost significance in modeling sequential data.

Considering a situation where the prior hidden state is $X_t \in R^{n\times d}$(with h hidden units) and the input is a mini-batch $H_{t-1} \in R^{n\times h}$(with n samples and d inputs). Reset gate $R_t \in R^{n\times h}$ and update gate $Z_t \in R^{n\times h}$ are calculated as follows:

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r) \tag{5}$$

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z) \tag{6}$$

The weight parameters are $W_{xr}, W_{xz} \in R^{d\times h}$ and $W_{hz} \in R^{h\times h}$, and $b_r, b_z \in R^{1\times h}$ are bias parameters.

DOI: 10.54254/2754-1169/47/20230370

The candidate hidden state $\tilde{H}_t \in R^{n\times h}$ at time step t is obtained by integrating the reset gate R_t with the RNN's traditional hidden state updating mechanism:

$$\tilde{H}_t = \tanh(X_t W_{xh} + (R_t \odot H_{t-1}) W_{hh} + b_h) \quad (7)$$

Here, $W_{xh} R^{d\times h}$ and $W_{hh} R^{h\times h}$ are weight parameters, $b_h \in R^{1\times h}$ is a bias term, and the symbol $\odot$ denotes element-wise multiplication. The $\tan h$ function is used here to limit the values between -1 and 1.

After calculating the candidate hidden state, the effect of the update gate Z_t is integrated to determine the new hidden state $H_t \in R^{n\times h}$, which determines the extent to which the new state comes from the old state H_{t-1} and the candidate state $\tilde{H}_t$. The update gate Z_t simply performs an element-wise convex combination between H_{t-1} and $\tilde{H}_t$. The following is the gated recurrent unit's final update formula:

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t \quad (8)$$

2.4. Implementation Details

In accordance with the study objectives, a five-layer neural network model was constructed utilizing TensorFlow's deep learning framework. The initial four layers were implemented as GRU layers, each comprising 128 hidden units and employing a dropout rate of 0.2. The final layer consisted of a fully connected layer housing a single unit. There were 10 epochs in the training process, with a batch size of 16. The Adam algorithm was employed as the optimizer, utilizing Mean Squared Error (MSE) as the chosen loss function, with a learning rate set at 0.001.

3. Results and Discussion

In this study, different algorithms, including GRU, standard BP neural network, and CNN were experimented, to predict the price trend of a single stock shown in Figure 3, Figure 4 and Figure 5. Due to its capacity to identify long-term dependencies in time-series data, the GRU model fared better than the other models in forecasting the long-term trend of stock prices, according to the results. On the other hand, the ordinary BP neural network and CNN models were better suited for handling local features and short-term trend prediction, resulting in relatively poorer overall predictive performance.

MSE was selected as the model evaluation metric. The GRU model achieves an MSE of 0.014 after training, and its predicted price trends closely align with actual prices. Thus, it can be considered accurate in short-term price trend prediction.

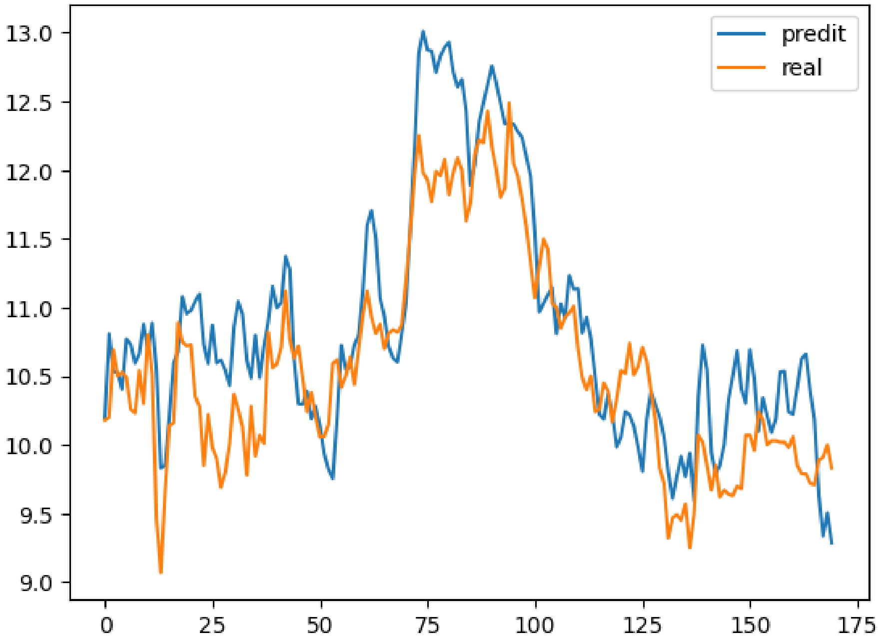

Figure 3: Prediction performance using GRU.

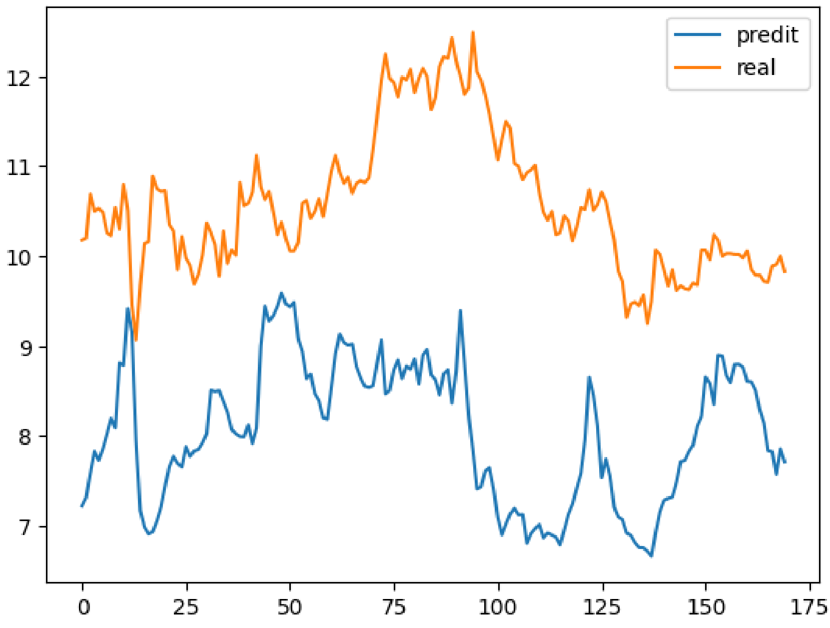

Figure 4: Prediction performance using ordinary BP neural network.

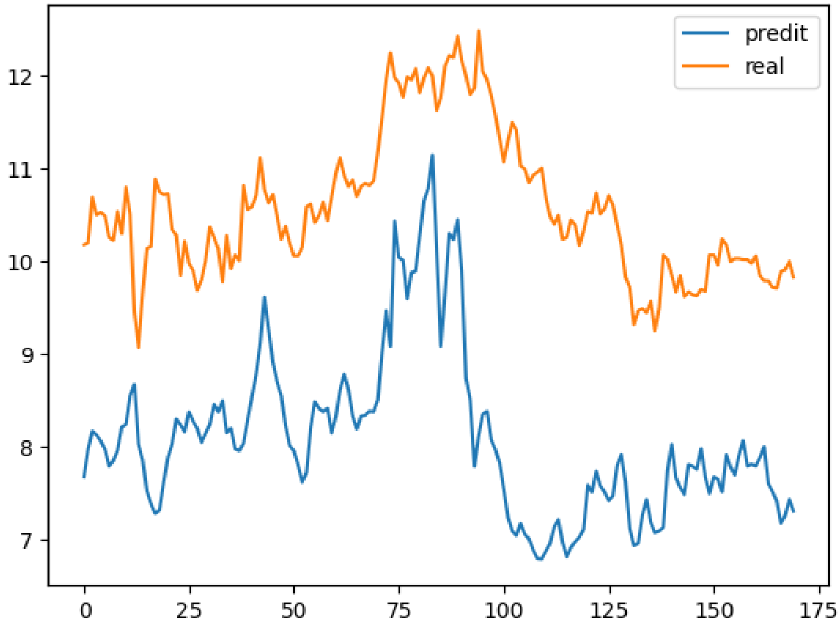

Figure 5: Prediction performance using CNN.

The disparity between the projected outcomes of the CNN and the conventional BP neural network, as evidenced by the graphical representation, indicates a substantial deviation from the observed real-world circumstances. Consequently, it is not doubted the efficacy of these models in generating accurate prognostications. Overall, the findings of this study suggest that the GRU model is more accurate in predicting the long-term trend of stock prices. Thus, the selection of the appropriate algorithm and parameters should be carefully considered based on the specific context. It is suggested that future research could explore more advanced deep learning models or integrate other factors, such as market sentiment and company performance, to improve the accuracy and usefulness of stock price prediction.

4. Conclusion

This study employed the GRU algorithm to predict the stock price trends during the pandemic period, in order to assist in investment decision-making. In order to reduce the problems of gradient explosion and decay and thus increase the accuracy of the model predictions, weight normalization and gradient clipping were used. The suggested approach was put to the test using several simulation trials, and the outcomes showed that the GRU algorithm performed better than the other algorithms in terms of fitting accuracy. In the future, further study plans to further improve the predictive performance by integrating the GRU algorithm with market sentiment and company performance factors. The findings of this study suggest that the GRU algorithm is a promising approach for predicting the long-term trend of stock prices during pandemic periods. The proposed method can potentially assist investors in making informed decisions and minimize risks in uncertain market conditions.

References

[1] Liu, T., Siegel, E., Shen, D. (2022) Deep learning and medical image analysis for COVID-19 diagnosis and prediction. Annual review of biomedical engineering, 24: 179-201.

[2] Qiu, Y., Yang, Y., Lin, Z., et al. (2020) Improved denoising autoencoder for maritime image denoising and semantic segmentation of USV. China Communications, 17(3): 46-57.

[3] D'Amato, V., Levantesi, S., Piscopo, G. (2022) Deep learning in predicting cryptocurrency volatility. Physica A: Statistical Mechanics and its Applications, 596: 127158.

[4] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017) Attention is all you need. Advances in neural information processing systems, 30.

[5] Akhil, S. & Purva, R. (2018) Application of LSTM, GRU and ICA for Stock Price Prediction Information and Communication Technology for Intelligent Systems pp 479–487

[6] Ya, G., Rong. W., and Enmin, Z. (2021) Stock Prediction Based on Optimized LSTM and GRU Models Scientific Programming/2021/Article.

[7] Sherstinsky, A. (2020) Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404: 132306.

[8] Williams, G., Baxter, R., He, H., et al. (2022) A comparative study of RNN for outlier detection in data mining, 2002 IEEE International Conference on Data Mining, Proceedings. IEEE, 709-712.

[9] Aston, Z., et al. (2021) Dive Into Deep Learning, arXiv:2106.11342

[10] Dey, R., Salem, F. M. (2017) Gate-variants of gated recurrent unit (GRU) neural networks, 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS). IEEE, 1597-1600.