Stock Ranking Prediction Based on an Adversarial Game Neural Network

SHIZHAO WEI, SHUNGANG WANG, SIYI SUN, AND YAN XU

1School of Mathematics and Physics, University of Science and Technology Beijing, Beijing 100083, China
2School of Statistics, Beijing Normal University, Beijing 100875, China
3The Graduate School of Art and Science, New York University, New York, NY 10003, USA

Corresponding author: Yan Xu (xuyan@ustb.edu.cn)

This work was supported in part by the National Key Research and Development Program of China (2020AAA0105103).

ABSTRACT With the globalization of the economy and financial markets today, predicting stock rankings and constructing appropriate portfolio strategies have become hot research topics for many scholars. However, because the stock market has different styles in different periods, market-style switching will seriously affect the prediction performance of the model. To eliminate the influence of style exposure and make the stock selection performance of the deep learning model more balanced, we propose an adversarial game neural network model based on LSTM and an attention mechanism for stock ranking prediction. We also combined trading tasks to construct an MS-WRSE loss function that considers stock rankings to optimize the network. Compared with classic time series prediction models, the adversarial game neural network can eliminate the influence of market-style factors on stock ranking predictions through the mutual game between the main neural network and the auxiliary neural network. This can strengthen the stock selection performance of the model.

INDEX TERMS Stock ranking prediction, adversarial game neural network, MS-WRSE loss.

I. INTRODUCTION

In recent decades, the stock market is easily affected by various factors, including the sentiments and expectations of traders, macroeconomic conditions, government policies and major events of listed companies, which make stock price forecast very challenging. Roy et al. used LASSO linear regression to forecast the stock market [1]. Idrees et al. applied an autoregressive integrated moving average model to predict stock market volatility [2]. Bose et al. proposed a model based on multivariate adaptive regression splines and deep neural networks to predict the stock price movement [3]. Improved models based on long short-term memory neural networks are used for time series forecasting. Chen et al. proposed a hybrid deep learning model that integrated an attention mechanism, a multi-layer perceptron, and a bidirectional long short-term memory neural network (LSTM) to forecast stock prices [4]. As one of the powerful data mining tools, deep learning plays an important role in academia and industry, especially in the stock market [5]. With its powerful representation ability, deep learning defines and predicts real tasks by minimizing the loss function and subsequently optimizing the actual problem.

In recent years, some studies have considered stock return forecasts as ranking forecasts [6], [7]. They ranked stocks according to the expected return and selected the top stocks for long and the bottom stocks for short. The key to ranking prediction lies in the selection and evaluation of models. How to choose a suitable model from many parameters and iterations is very important to prevent overfitting while ensuring model convergence. There are many studies on model performance evaluation indicators. Apart from mean square error (MSE) [8], economic indicators for model selection also provide help, such as investment return ratio (IRR) [7] and Sharpe ratio [9], which can make people more objectively evaluate the model.

Game theory is the study of competition or cooperation between individuals. Combined with theoretical computer science, it has jointly contributed to the development of algorithmic game theory [10]. Inspired by the idea of game theory, many algorithms have been developed, such as the generative adversarial network (GAN) [11], which establishes a game relationship between the generator and the discriminator and finally reaches the equilibrium state of both parties.
Oliehoek et al. directly used game theory techniques to solve the problem of difficult GAN training [12]. In the field of multiagent reinforcement learning (MARL), multiple agents interact with the same environment and finally reach an equilibrium state. In addition, He et al. combined game theory and machine learning, proposed a game machine learning framework and applied it to auction problems with variable data environments [13]. The idea of game theory is to use the constant interaction between multiple players to achieve the optimal equilibrium state. Adversarial generative neural network is a relatively successful case of applying game theory to the field of machine learning. Therefore, the introduction of game theory into machine learning can play an important role.

Combining the above research content, this work proposes an adversarial game neural network model for the task of stock ranking prediction. Compared with the classic time series prediction model, such as autoregressive integrated moving average model (ARIMA), the advantage is that through the mutual game between the main network and the auxiliary network, the influence of market-style factors on the stock ranking prediction is gradually eliminated.

The remaining part of this study is structured as follows: Section 2 introduces the research methods of this study, including the model framework, MS-WRSE loss, and valuation indicators. Section 3 explains the dataset and three experiments. Section 4 shows the performance evaluation of the proposed adversarial game neural network on stock ranking prediction, including the experimental setting, data description, model comparison and discussion. Section 5 summarizes the experimental results and provides appropriate suggestions for future research content.

II. METHODS AND MATERIALS
Because stock forecast cannot be simply considered price forecast, this study designs a new loss function based on the idea of stock ranking to optimize the model training. At the end of this section, some classic evaluation indicators in stock prediction tasks are given for the trained models.

A. FACTOR NEUTRALIZATION
In quantitative trading, professional investors often use indicators to choose stocks, and these indicators for stock selection are called factors. When these factors are used for investment transactions, they are often interfered with by other factors, such as market style and cyclical laws, which results in a certain bias in the selected stocks (or style exposure) and makes the selection of stocks too concentrated. Simultaneously, a certain risk is increased. To address the problem of factor style exposure, methods of neutralizing factors are preferable. This section mainly adopts linear regression. For a given factor sequence \( \{Y_i^t\} \), \( i = 1, \ldots, m \), \( t = 1, \ldots, T \), where \( t \) is time, and \( i \) is the number of stocks at time \( t \). Assuming the style exposure factor sequence \( \{X_i^t\} \), \( i = 1, \ldots, m \), \( t = 1, \ldots, T \), linear regression is performed using ordinary least squares (OLS) to obtain the residual sequence \( \{M_i^t\} \), and the formula is as follows:

\[
M_i^t \triangleq \varepsilon_i^t = Y_i^t - \alpha X_i^t
\]

where \( \alpha \) is the coefficient of the linear regression independent variable, and \( \varepsilon_i^t \) is the residual, which is defined as the neutralized factor \( M_i^t \).

B. ADVERSARIAL GAME NEURAL NETWORK
Although the neutralization method of OLS linear regression can eliminate the style exposure, it also greatly changes the position relationship of the original sequences, which results in a large difference in the sequences before and after neutralization and introduces unnecessary invalid information. Therefore, we propose the adversarial game neural network (AGNN), which can largely retain the original ranking information and reduce the burden of model learning for high-SNR (signal noise ratio) data.

The main idea of an adversarial game neural network is to assign multiple neural networks to different tasks, and there is a certain correlation between these tasks. A balanced state for all networks was achieved by optimizing the loss function of each network. In this state, each neural network performs its own task using other networks. Compared with the optimization of a neural network, the adversarial game neural network introduces additional information from the non-feature level; thus the model can focus on its own task while taking into account the random intervention of other networks on the task, thereby improving the representation ability of the model and the flexibility of solving tasks.

The AGNN is mainly divided into two parts, as shown in Figure 1. The left part is the auxiliary neural network, whose inputs are exposure factors such as circulating market value and industry. The auxiliary neural network is responsible for learning the influence of exposure factors in given labels. The right part is the main neural network, and its inputs are basic features of stocks in a period such as the opening price, closing price, highest price, and lowest price. The main neural network is responsible for learning the relationship between the features and the labels after removing the exposure factor. The deep learning model optimizes the network parameters by minimizing the loss function; i.e., the primary task of model training is to find the functional relationship between inputs and labels. This paper designs a framework for the interactive training of two neural networks with different inputs. The main network and auxiliary network use different input features to improve their respective prediction accuracy. However, in the optimization process, the label learned by the two networks is dynamically changed, which is derived from the training states of the two networks. Finally, when the loss functions of the two networks are in a stable state, the interactive training of the model is in a balanced state, where the main network learns the label that removes the representation of the auxiliary network. Therefore, in this paper, the architecture is named the adversarial game neural network. Next, this paper will detail the composition of the auxiliary neural network and the main neural network, respectively.
FIGURE 1. Adversarial game neural network framework.

As shown in Figure 1, the auxiliary neural network consists of multi-layer perceptions (also called feed-forward layers) that perform nonlinear transformations on the input. The main neural network is more complicated than the auxiliary neural network. It mainly consists of 3 parts: a LSTM layer [14], an attention block and a linear layer.

Recurrent neural networks (RNNs) play a good role in short-sequence input prediction problems, but they are prone to problems such as vanishing gradients on long-sequence problems [15]. LSTM solves the gradient problem in RNNs, and its special gating mechanism enhances the fitting ability of the model and becomes a powerful tool for time series prediction problems. The attention block consists of 3 parts: a self-attention layer (self-attention) [16], the skip connection [17] with layer normalization [18] (Add & Norm), and a feed-forward layer (Feed Forward). Self-attention is a special attention calculation. The skip connection adds an identity mapping after the network propagation, which can stabilize the update of the gradient during the neural network learning process, and its gradient of the loss function and self-attention are shown in Equation (2). Layer normalization can accelerate the convergence of the network.

\[
\text{Self-Attention}(XW_q, XW_k, XW_v) = \text{Softmax} \left( \frac{XW_q W_k^T X}{\sqrt{d_k}} \right) XW_v
\]

\[
x_{l+1} = x_l + F(x_l, W_l)
\]

\[
\frac{\partial L}{\partial x_l} = \frac{\partial L}{\partial y_2} \left( 1 + \frac{\partial L}{\partial y_1} \right) F(x_l, W_l)
\]

where \(W_q, W_k, W_v\) are learnable parameters, \(d_k\) is the dimension of \(W_k\), \(l\) is the location of the network layer, \(L\) is the unit cumulative feature sum of each residual block, \(F\) is the mapping of the network at layer \(l\), \(\epsilon\) is the loss function, and the second equation of Equation (2) illustrates the skip connection.

According to the universal approximation theorem [19], a neural network with a linear output layer and at least one hidden layer can theoretically approximate any continuous function on a compact subset of \(R^n\) with high accuracy, where \(R^n\) is the real number space with \(n\) dimensions. Therefore, the introduction of feed-forward layers can increase the fitting ability of the network.

Finally, extracted features after stacking \(N\) attention blocks are passed through the linear layer to obtain the output value as part of the auxiliary network input.

The auxiliary neural network and main neural network form a closed circuit for mutual adversarial game training. The main steps of Algorithm 1 are as follows: (1) Denote the output values of the auxiliary network and main network as \(y_1\) and \(y_2\) at each epoch. (2) The main network updates the parameters by minimizing the loss function, and the final output value \(y_2\) gradually converges to \(label - y_1\). (3) The output value \(y_1\) of the auxiliary network gradually converges to \(label - y_2\). (4) The auxiliary network and main network finally reach an equilibrium state with \(label = y_1 + y_2\). At this time, the output value of the main network is the factor value after excluding exposure factors. We proposed
the new model named attention long short-term memory (ATTLSTM), which combines self-attention and LSTM. This paper uses LSTM as a feature extraction model, and trains low-level features through LSTM at each time step to extract more abstract high-level features. LSTM is the beginning part of the main network of the adversarial game neural network. For time series features, such as the opening price, closing price and trading volume of a period of time, we normalize and input them into LSTM, extract the last hidden state of LSTM, and use it as the input of the attention mechanism.

Algorithm 1 Adversarial Game Neural Networks Training

Require:
1. Samples \( \{(x_i, label_i)\}_{i=1}^{N} \);
2. Initialized main neural network;
3. Initialized auxiliary neural network.

Ensure:
1. Trained main neural network
2. The auxiliary neural network makes predictions and gets the prediction \( y_1^i \);
3. Using \( (x_i, label_i - y_1^i) \) to train the main neural network;
4. The main neural network makes predictions and gets the prediction \( y_2^i \);
5. Using \( (x_i, label_i - y_2^i) \) to train the auxiliary neural network;
6. \( \text{end while} \)

C. MS-WRSE LOSS

Because stock forecasting is not the pursuit of accurate forecasting of returns, it is more about the consistency of stock forecast ranking and real stock ranking. Therefore, we construct a loss function based on MSE and ranking, which is named the mean square-weighted ranking error (MS-WRSE) loss function. The MS-WRSE loss function is defined as follows.

\[
\text{MS-WRSE Loss} = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 + \gamma \frac{1}{m^2} \sum_{i=1}^{m} \alpha_i (y_i - \hat{y}_i)^2
\]

where \( \alpha_i = \frac{(R(x_i) - R(\hat{x}_i))^2 - \min[(R(x_i) - R(\hat{x}_i))^2]_{i=1}^{m}}{\max[(R(x_i) - R(\hat{x}_i))^2]_{i=1}^{m} - \min[(R(x_i) - R(\hat{x}_i))^2]_{i=1}^{m}} \)

The MSE-WRSE loss function has two items: the MSE and the weighted ranking score error. In the second item, \( R(x_i) \) is the position after ascending (or descending) order of \( x_i \) in the sequence \( \{x_i\}_{i=1}^{m} \), and \( \alpha_i \) is the normalized weighting coefficient of the ranking error, which aims to impose larger penalties on samples with large ranking gaps between true and predicted values. \( \gamma \geq 0 \) is the weight coefficient that weighs the two losses. \( y_i \) and \( \hat{y}_i \) are the true and predicted values of the \( i^{th} \) sample, respectively.

D. EVALUATION MEASURES

This study uses both statistical and economic indicators to choose and evaluate the model. We use selection criteria based on akaike information criterion (AIC), bayesian information criterion (BIC), and hannan-quinn information criterion (HQIC).

\[
\begin{align*}
\text{AIC} &= 2k - 2 \ln(\hat{L}) \\
\text{BIC} &= k \ln n - 2 \ln(\hat{L}) \\
\text{HQIC} &= -2\hat{L} + 2k \ln \ln(n)
\end{align*}
\]

where \( k \) is the number of estimated parameters in the model, \( \hat{L} \) is the maximum value of the likelihood function for the model, and \( n \) is the number of observations.

For statistical indicators, we use the variance accounted for (VAF), relative average absolute error (RAAE), root means absolute error (RMAE), root means squared error (RMSE), RMSE-observations standard deviation ratio (RSR), mean absolute percentage error (MAPE), weighted mean absolute percent error (WMAPE) and Nash Sutcliffe efficiency (NSE).

\[
\begin{align*}
\text{VAF} &= 1 - \frac{\text{var}(y_i - \hat{y}_i)}{\text{var}(y_i)} \\
\text{RAAE} &= \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{\text{max}(\text{abs}(y_i), 1)} \\
\text{RMSE} &= \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \\
\text{RSR} &= \frac{\text{RMSE}}{\text{RMS} - \text{obs}} \\
\text{MAPE} &= \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{y_i} \\
\text{WMAPE} &= \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{y_i} \\
\text{NSE} &= 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i^\text{pred})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}
\end{align*}
\]

where \( y_i \) is the observed value (or true value) of the \( i^{th} \) sample, \( \hat{y}_i \) is the estimated value, \( \hat{y}_i^\text{pred} \) is the predicted value, \( \bar{y} \) is the mean, and \( \text{var} \) is the sample variance.

Economic indicators include the investment return ratio (IRR) \[4\] and Sharpe ratio (SR) \[9\], which are used to measure the stock selection ability of the model.

\[
\begin{align*}
\text{IRR} &= \prod_{i=1}^{n} (1 + r_i) - 1 \\
\text{SR} &= \frac{E(\text{R}_\text{P}) - \text{R}_f}{\sigma_p}
\end{align*}
\]

where \( y_i \) is the true value, \( \hat{y}_i \) is the predicted value, and \( r_i \) is the profit return ratio on the \( i^{th} \) day. \( E(\text{R}_\text{P}) \) is the expected annual remuneration rate of the investment portfolio. \( R_f \) is the annual risk-free interest rate, and \( \sigma_p \) is the standard deviation of the annual remuneration rate of the investment portfolio. \( R_f \) is taken as 5% in this experiment.

III. DATA AND EXPERIMENT

A. DATASET

The data come from the daily k-line data of the RMB common stock market in the Wind database (https://www.wind.com.cn/) and include the opening price, closing price, highest price, lowest price, trading volume, and trading amount. Then, we eliminate stocks with missing data. All samples are divided into three parts. The four-year data
from 2015 to 2018 are used as the training data. The data in 2019 are the cross-validation dataset, which is used for the model selection under different epochs and parameters. The data from 2020 are the test dataset, which is used to evaluate the model. The past 40 days of each stock are selected as the input, and the z-score normalization is used to transform each origin feature distribution to a standard normal distribution. The z-score normalization is calculated as

\[
    z = \frac{x - \text{mean}(x)}{\text{std}(x)}
\]

where \(\text{mean}(x)\) and \(\text{std}(x)\) are the mean and standard deviation of each stock, respectively. The target value \(Y_t\) is calculated as:

\[
    Y_t = \frac{\text{close}_{t+1} - \text{open}_{t+1}}{\text{open}_{t+1}} - Y_{\text{market}}^{t+1}
\]

where \(Y_{\text{market}}^{t+1}\) is the average of all stocks at the \((t+1)^{th}\) day.

**B. EXPERIMENTAL SETTING**

The profit of stock trading comes from the price difference at different moments. By buying stocks at low prices and selling them at high prices, traders can obtain profit. In this study, the market data of each stock in the past 40 days are used as the input data, and the return rate of each stock in the future 1 day is used as the training target of the model. Assume that the transaction fee generated by the transaction and the impact of the transaction on the market are not considered. First, the prediction results of the models for stocks are sorted from large to small and equally divided into 5 groups, and stocks within each group are allocated equal weights. When the total number of stocks is not divisible by 5, an approximate method is used. Suppose that stocks whose prices are predicted to rise at the close of the market are bought at the market opening moment and sold before the market closing moment; then, the returns brought by the stocks in each group are separately calculated.

The baseline models in this experiment are LSTM and ATTLSTM, and three comparative experiments are performed. Experiment 1: The LSTM and ATTLSTM networks are used to learn the relationship between the market data of the past 40 days and the return rate in the next day. Experiment 2: First, we use linear regression to eliminate the relationship between circulating market value and the return rate on the next day to obtain the residual sequence. Then, we use the LSTM and ATTLSTM networks to learn the relationship between the market data and the residual sequence. Experiment 3: LSTM and ATTLSTM are used as the main network, and a deep neural network (DNN) is used as the auxiliary network. The input of DNN is the circulating market value, and we construct adversarial game long short-term memory (AG-LSTM) and adversarial game attention long short-term memory (AG-ATTLSTM). Then, we use Algorithm 1 to train the model and use the main network to make predictions. The purpose is to obtain a model that can select stocks and has a small relationship between stock picking and circulating market values. The output value of the model is used as a trading signal to calculate the profitability of each group.

The neural network uses the adaptive moment estimation (Adam) optimizer [20], and the learning rate is set to 0.001. The decay rates of the first-order moment and second-order moment are set to the default values, which are 0.9 and 0.999, respectively. We set the batch size to 64 and the iteration epoch to 100. The model uses dropout regularization [21] to reduce the possibility of model overfitting. DNN uses a leaky rectified linear unit (Leaky-ReLU) activation function [22]. By using selection criteria, parameters are listed in Table 1.

### TABLE 1. Model parameters.

| Parameter Name                        | Value |
|--------------------------------------|-------|
| learning rate                        | 0.001 |
| first-order moment decay order        | 0.9   |
| second-order moment decay order       | 0.999 |
| batch size                           | 64    |
| iteration epoch                       | 100   |

Then, the model with the smallest MSE is chosen when the loss function is stable on the validation dataset. We calculate the rate of return of each group on the test dataset and draw the curve of each group. For the largest return rate sequence, we calculate the annualized return (Top@1-IRR), Sharpe ratio (TOP@1-SR), long-short annualized return (Top@1-Top@n-IRR) and long-short Sharpe ratio (Top@1-Top@n-SR).

**IV. RESULTS AND DISCUSSION**

We rank the prediction values of stocks in each day in descending order and divide them into 5 groups (top@1 to top@5), where top@1 represents the group consisting of the top 20% stocks with the largest output values, and top@5 represents the group consisting of the top 20% stocks with the smallest output value. Then, we calculate the average excess return rate of each group, which is denoted as the return of each stock minus the mean return of the entire market and the average change in circulating market value in each group.

Experiment 1 uses LSTM and ATTLSTM to train and predict the original return rate data, and changes in the cumulative return rate sequences are shown in Figure 2. LSTM and ATTLSTM, which directly train the return data, have obvious differences in return between different groups, and the return sequence has greater fluctuations. Among the 5 different groups, the average return of the LSTM model from top@1 to top@5 shows a decreasing trend of 2.0132%, 1.1541%, -0.2429%, -1.2935% and -1.6224%, respectively, while the average returns of the ATTLSTM model are 1.1450%, 0.5742%, -0.2097%, 0.1804% and -1.7070%, respectively. For top@1, ATTLSTM has less volatility and more stable returns than LSTM.

Then, we perform Experiment 2. First, we use OLS linear regression to obtain the residual sequences. We test the
significance of the linear regression between the circulating market value sequences and the residual sequences, and there is no correlation between them. The influence of this variable on the residual sequence can be considered eliminated. Then, we train LSTM and ATTLSTM using 6 standardized features with a residual sequence. The models are denoted as LSTM-Res and ATTLSTM-Res, respectively. The cumulative return rate sequences of the two models are shown in Figure 3. Figure 3 shows that models trained by the residual sequences cannot learn the consistency of the ranking very well. For LSTM-Res and ATTLSTM-Res, the return rate of top@1 with the highest predicted value is the lowest. It is difficult to maintain the real ranking relationship by directly fitting the neutralized residual sequence, which is contrary to the original task.

Finally, we perform Experiment 3. LSTM, ATTLSTM, and DNN are used to construct adversarial game neural networks denoted by AG-LSTM and AG-ATTLSTM, respectively. Among the 5 different groups, the average returns of the AG-LSTM model from top@1 to top@5 are 2.7196%, −0.2465%, −0.4420%, −0.9653% and −1.0484%, respectively, with a decreasing trend. The average returns of the AG-ATTLSTM model from top@1 to top@5 are 3.18% −0.4700%, −1.0831% and −2.1664%, respectively, with a greater decreasing trend than AG-LSTM. In the comparison, top@1 of AG-ATTLSTM is 0.6287% higher than top@1 of AG-LSTM, while top@5 is 1.1180% lower than AG-LSTM, which indicates that AG-ATTLSTM has a stronger stock picking ability. In addition, from Figure 4, the curve trend of return sequentially decreases, and the curve of AG-ATTLSTM is clearer and more discriminative, which proves that AGNN, especially AG-ATTLSTM, has a strong ability to predict stock rankings. Through further analysis in the same market conditions, the retracement interval of return in the top@1 group trained by AGNN becomes shorter, which is significantly improved compared to Experiment 1.

Figure 5 shows 30 types of circulating market average values with 5 groups and 6 models. The horizontal axis in the figure represents the five groups with descending predicted values, and the vertical axis represents the average value of the circulating market values (100 million yuan). The circulating market values of each model in the 5 groups are quite different. For the top@1 group with the largest return sequence, ATTLSTM-Res has the highest average market value (approximately 176), while AG-LSTM has the smallest average market value (approximately 125), i.e., there is a range of approximately 50. The average circulating market value of the LSTM in the five groups successively increases, which indicates that the selection of stock portfolios with higher returns tends to be stocks with smaller market values, and stock portfolios with lower returns tend to be stocks with larger market values. In contrast, AG-ATTLSTM is approximately uniformly distributed, which indicates that the model picks stocks more dispersed and faces relatively lower unknown market risks.

Finally, we calculate the economic indicators of the models in the above experiments and show them in Table 2. Since the models obtained in Experiment 2 cannot guarantee the consistency between predicted ranking and real ranking, the model does not satisfy the declining state of cumulative return from group top@1 to top@5. Therefore, Table 2 will not include the results of Experiment 2. LSTM with attention blocks performs better than LSTM alone in
Experiments 1 and 3. The proposed AG-ATTLSTM has the best performance. Top@1-IRR of AG-ATTLSTM is 4.2743% higher than the market average return. And Top@1-Top@5 IRR of AG-ATTLSTM is 10.6045%, which is largely ahead of the other three models. For the Top@1-SR indicator, the value is negative, which implies that the return rate is lower
than the benchmark of 5%. However, whether it is facing a good or bad market, the superiority of AG-ATTLSTM is also reflected. To further explore the regression ability of regression models, this study calculated 9 statistical indicators of 4 models to evaluate the regression tasks, and the results are shown in Table 3.

As seen from Table 3, due to the low signal-to-noise ratio of the stock data, the regression accuracy of the four models is generally low. By comparison, the ATTLSTM model in this paper has slightly higher prediction accuracy than LSTM. For the adversarial game neural network, especially for AG-ATTLSTM, compared with ATTLSTM, it does not introduce additional large error while eliminating style exposure, and it is better than ATTLSTM on RMAE and WMAP. However, AG-LSTM has lower regression accuracy than LSTM, so the selection of the main network of the adversarial game neural network is one of the key factors that affect the final effect of the model. Further analysing the above experimental results, the adversarial game neural network proposed in this paper has the following characteristics. First, in the task of nonlinear elimination of style exposure factors, the adversarial game neural network can play a certain role, and this training paradigm can be extended to more meaningful tasks. In addition to excluding the impact of circulating market values in the stock market on the stock selection ability of the model, other factors can be selected, such as the industry to which the stock belongs and liquidity of funds. In addition, the regression results show that the selection of different main networks has a large difference in the training effect of the adversarial game network. Choosing ATTLSTM with a stronger representation ability as the main network introduces better results than LSTM training under this framework. Therefore, choosing a model with stronger representation power usually helps the training of game adversarial networks.

V. CONCLUSION
This research proposes a new model for the stock ranking prediction problem. The model additionally considers the influence of circulating market value during training, which improves compared with the traditional linear regression neutralization and classic LSTM. AGNN achieves the equilibrium state between main network and auxiliary network by using the consistency of the optimization goal. The AGNN has yielded good experimental results on stock prediction tasks that must consider the influence of additional factors on the model. The AGNN provides a new method to eliminate the influence of latent variables on target labels. In addition, we use the MS-WRSE loss function to express the ranking relationship between stocks, which is more meaningful than general stock price prediction tasks. On the IR index, the proposed model is 2.0223% higher than the benchmark model and 6.7848% higher on the long-short IRR index. Finally, this research uses regression errors and economic evaluation indicators to evaluate models and verifies the advantages of adversarial game neural networks in stock trading back-testing.

However, the adversarial game neural network has certain shortcomings. Due to the difference between main network and auxiliary network learning task, the output results of each round will affect each other. It is difficult to ensure that the two networks always provide positive promotion to each other. In addition, whether there is a potential connection between input features and target values in the auxiliary network cannot be determined by the learning results of the auxiliary network. Therefore, it is necessary to rely on human prior knowledge to find clear potential factors as network input and make experimental results more realistic.

In future research, the neural network framework for AGNN can be further improved. For the AGNN proposed in this paper, there is only one auxiliary neural network to train the main neural network. In the future, more auxiliary neural networks can be used to provide more information for the main neural network, and a better training state can be achieved through multi-party games. In addition, if we consider multiple related tasks, it is not sufficient to rely on one main neural network at this time. Researchers can design multiple main neural networks to better complete the model learning through the interconnection among main neural networks.

CONFLICT OF INTEREST
The authors declare no competing financial interests.

ACKNOWLEDGMENT
The funders had no role in the design of the study, the collection, analysis, and interpretation of data and in writing the manuscript.

REFERENCES
[1] S. S. Roy, D. Mittal, A. Basu, and A. Abraham, “Stock market forecasting using lasso linear regression model,” in Afro-European Conference for Industrial Advancement. Cham, Switzerland: Springer, 2015, pp. 371–381.
[2] S. M. Idrees, M. A. Alam, and P. Agarwal. “A prediction approach for stock market volatility based on time series data,” IEEE Access, vol. 7, pp. 17287–17298, 2019.
[3] A. Bose, C.-H. Hsu, S. S. Roy, K. C. Lee, B. Mohammadi-Ivatloo, and S. Abhimannan, “Forecasting stock price by hybrid model of cascading multivariate adaptive regression splines and deep neural network,” Comput. Electr. Eng., vol. 95, Oct. 2021, Art. no. 107405.
[4] Q. Chen, W. Zhang, and Y. Lou, “Forecasting stock prices using a hybrid deep learning model integrating attention mechanism, multi-layer perception, and bidirectional long-short term memory neural network,” IEEE Access, vol. 8, pp. 117365–117376, 2020.
[5] F. G. D. C. Ferreira, A. H. Gandomi, and R. T. N. Cardoso, “Artificial intelligence applied to stock market trading: A review,” IEEE Access, vol. 9, pp. 30898–30917, 2021.
[6] Q. Song, A. Liu, and S. Y. Yang, “Stock portfolio selection using learning-to-rank algorithms with news sentiment,” Neurocomputing, vol. 264, pp. 20–28, Nov. 2017.
[7] T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” Eur. J. Oper. Res., vol. 270, no. 2, pp. 654–669, 2018.
[8] S. S. Roy, R. Chopra, K. C. Lee, C. Spampinato, and B. Mohammadi-Ivatloo, “Random forest, gradient boosted machines and deep neural network for stock price forecasting: A comparative analysis on South Korean companies,” Int. J. Ad Hoc Ubiquitous Comput., vol. 33, no. 1, pp. 62–71, 2020.
[9] W. F. Sharpe, “The Sharpe ratio,” *J. Portfolio Manage.*, vol. 21, no. 1, pp. 49–58, 1994.

[10] T. Roughgarden, “Algorithmic game theory,” *Commun. ACM*, vol. 53, no. 7, pp. 78–86, 2010.

[11] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative adversarial networks: An overview,” *IEEE Signal Process.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.

[12] F. A. Oliehoek, R. Savani, J. Gallego-Posada, E. van der Pol, E. D. de Jong, and R. Groß, “GANs: Generative adversarial network games,” 2017, arXiv:1712.00679.

[13] D. He, W. Chen, L. Wang, and T.-Y. Liu, “A game-theoretic machine learning approach for revenue maximization in sponsored search,” 2014, arXiv:1406.0728.

[14] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] D. Wang, J. Fan, H. Fu, and B. Zhang, “Research on optimization of big data construction engineering quality management based on RNN-LSTM,” *Complexity*, vol. 2018, pp. 1–16, Jul. 2018.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[18] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” 2016, arXiv:1607.06450.

[19] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991.

[20] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T.-S. Chua, “Temporal relational ranking for stock prediction,” *ACM Trans. Inf. Syst.*, vol. 37, no. 2, pp. 1–30, Apr. 2019.

SHIZHAO WEI received the master’s degree from University of Science and Technology Beijing, where he is currently pursuing the Ph.D. degree in information and computing science.

He not only has many years of scientific research experience, but also a lot of engineering practice experience, and he is good at machine learning and deep learning algorithms. His current research interests include the application of machine learning and deep learning in quantitative trading, convex optimization problems in the financial field, and time-series data modeling.

SHUNGANG WANG received the B.S. degree in applied mathematics from University of Science and Technology Beijing, Beijing, China. He is currently pursuing the M.S. degree in applied statistics with Beijing Normal University. His previous work is about time series model for prediction.

His research interest includes the application of machine learning for analyzing financial market.

SIYI SUN received the B.S. degree in information and computational science from University of Science and Technology Beijing, Beijing, China. Since 2017, she started her university majoring in physics, and then she changed her major to information and computational science in 2019. She is currently pursuing the M.S. degree in computer and information science with New York University.

Her previous work is about fuzzy inference system for industry modeling. Her research interests include quantum computer science, mathematical modeling, data science, and time-series data modeling.

YAN XU received the B.S. and M.S. degrees in mathematics from Jilin University, Jilin, China, in 2000 and 2003, respectively, and the Ph.D. degree in mathematics from China Agricultural University, Beijing, China, in 2012.

She is currently a Professor with the Institute of Mathematics and Physics, University of Science and Technology Beijing, Beijing. Her research interests include the prediction of posttranslational modification, machine learning algorithms, and deep learning in bioinformatics and economics.

Dr. Xu has got the grants from the Natural Science Foundation of China, the Ministry of Science and Technology of China, and the Fundamental Research Funds for the Central Universities. She also received excellence awards of national university student mathematical modeling competition, from 2008 to 2021.