Artificial Intelligence and Human-Computer Interaction
Y. Ye and P. Siarry (Eds.)

doi:10.3233/FAIA240151

Research on Stock Index Prediction Model Based on Deep Reinforcement Learning

Chuanwei YU [1] and Liang ZHANG
Institute of Information Science and Technology Hangzhou Normal University Hangzhou, China
ORCiD ID: Chuanwei Yu https://orcid.org/0009-0008-5992-8440

Abstract. Stock market prediction and trading strategies have been extensively studied in finance and AI. Due to market volatility, it's challenging for investors to achieve high returns. To address this, we explore deep reinforcement learning using LSTM-DQN and FC-DQN models for stock trading. LSTM-DQN combines Recurrent Neural Networks and Q-learning to capture stock data patterns, while FC-DQN uses a fully connected neural network. After applying discrete wavelet transform to the raw stock data for denoising and smoothing purposes, experiments were conducted. When comparing these two models in terms of accuracy and cumulative returns, the LSTM-DQN model outperforms the FC-DQN model. It's evident that the LSTM-DQN model generates greater profits in terms of investment returns, making it a more suitable choice for investors. Finally, this paper conducts an analysis and discussion of the experimental results.

Keywords. Deep reinforcement learning, stock market, denoise

1. Introduction

The stock market is a complex and dynamic system that attracts wide attention from investors and researchers [1]. Predicting stock prices and formulating effective trading strategies are crucial tasks in the field of finance. Traditional stock trading methods often rely on technical indicators and statistical models, which often struggle to capture the complex patterns and temporal dependencies in stock price data [2]. Deep learning models have shown promising results in various fields due to their ability to learn hierarchical representations and capture complex patterns. In the domain of reinforcement learning algorithms, the Q-learning algorithm is widely used for training intelligent agents in decision-making problems. The combination of deep learning and Q-learning has led to the development of deep reinforcement learning models, which have demonstrated outstanding performance in tasks such as gaming, robotics, and natural language processing. In the context of stock trading, deep reinforcement learning models offer a hopeful approach to address the limitations of traditional methods [3]. Weng et al. combine different classifiers to make predictions on data [4]. Barra et al. researched the applications of deep learning and time series for image encoding [5]. Labate et al. use signal processing methods to analyze stock data and make predictions [6].

[1] Corresponding Author: Chuanwei Yu, Institute of Information Science and Technology Hangzhou Normal University, Hangzhou, China. Email: y_chuanwei@foxmail.com.

It can be observed that current research mostly analyzes and predicts the stock market by comparing single and simple models. There is a lack of in-depth and multi-perspective processing of stock data, as well as the utilization of hybrid models for stock analysis and prediction. Based on this, the goal of future research is to process stock data from multiple angles, explore different stock features, and employ hybrid models to improve the accuracy of stock prediction.

2. Theoretical analysis and research hypothesis

2.1. The integration of deep reinforcement learning theory with the stock market

Traditional deep learning models do face certain challenges, particularly when dealing with high-dimensional features, such as in the case of analyzing stock market historical data. Due to the massive volume of stock data and the presence of noise, neural networks often struggle to achieve satisfactory learning outcomes [7]. Therefore, there is a need for a model that can extract patterns from extensive data and capture the underlying rules of stock market behavior.

Therefore, this experiment utilizes a deep neural network-based reinforcement learning model. In this model, a neural network is used as the reinforcement learning agent, leveraging its powerful learning capabilities to extract relationships among stock data. The model is enhanced by employing reinforcement learning algorithms to maximize the return on investment. Deep reinforcement learning consists of several critical components, including state, next state, reward, action, experience replay [8]. The flowchart of the reinforcement learning algorithm is as Figure 1:

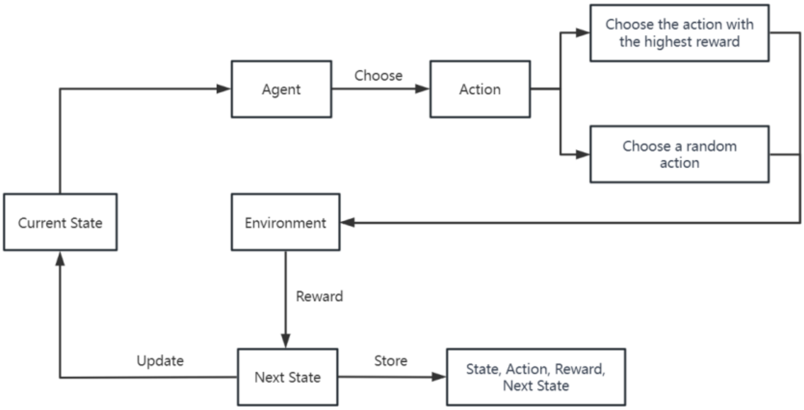

Figure 1. Deep reinforcement learning algorithm flowchart.

In reinforcement learning algorithms, the process begins with the environment providing the current state of the intelligent agent. Then, based on the results obtained from the trained neural network, the agent selects an appropriate action. At this point, there are two options: random exploration or choosing the action with the highest value each time. The chosen action is then fed back to the environment, which updates the state to the next state and outputs the action value. This feedback is then given to the intelligent agent, and the training loop continues until the network is optimized to its best performance.

2.2. The Integration of Discrete Wavelet Transform and the Stock Market

Discrete Wavelet Transform (DWT) is a mathematical tool that breaks down a signal into different frequency components and time intervals, allowing for a more detailed analysis of both high and low-frequency patterns. Combining DWT with the stock market involves applying this technique to stock price or trading volume data to extract meaningful insights.

In the context of the stock market, DWT can be utilized to decompose the original time series data into various scales or levels, with each level representing a different frequency range. Subsequently, it is possible to remove the corresponding levels and recombine the signals to achieve the effect of denoising and smoothing the original time series data.

This study aims to employ discrete wavelet denoising to eliminate significant fluctuations from stock data. Following denoising, reinforcement learning principles will be utilized to predict stock market data, with the objective of achieving maximal returns and higher accuracy.

3. Research design

3.1. Sample selection and data sources

In this experiment, the dataset was obtained from publicly available online sources. Several influential stock data were primarily selected. The specific stock data and their corresponding dates are listed in the following Table 1:

Table 1. The selection of stock data includes its daily opening price, closing price, highest price, lowest price, and trading volume.

Name	Time period
SSEC	2003/03/03~2023/05/08
HSI	2001/07/09~2023/05/08

3.2. Feature selection

The selection of features can be categorized into fundamental data and technical factors. The specific selections are as follows Table 2:

Table 2. The selected features and description.

Name	Full Name or Description
MA	Moving Average.
EMA	Exponential Moving Average.
MACD	Moving Average Convergence Divergence.
BR	The technical indicator that measures buying and selling willingness.
DEA	Divergence and Convergence Average.
CDP	Central Pivot Point.
BBI	Bull and Bear Index.
KDJ	Stochastic Oscillator.
RSI	Relative Strength Index.
BOLL	Bollinger Bands.
EXPMA	Exponential Moving Average.
VR	Volume Rate of Change.

3.3. Feature engineering

- Data cleaning: Raw stock data often contain missing values, outliers, and errors. To prepare the data for neural network input, we address these issues. Forward filling replaces missing values, while outliers and errors are removed.
- Data smoothing: Wavelet transform handles non-smooth time series data by analyzing specific frequency components. Using wavelet transform to address stock data noise makes it smoother. See Figure 2 for denoising comparison.
- Normalization: Unnormalized stock data often exhibits volatility, leading to significant price discrepancies. Failure to normalize the data can impede the convergence of neural networks and impact training outcomes [9]. Therefore, in this experiment, we will compress the data into a specific range. Firstly, we will split the dataset into a training set and a testing set, calculate the mean μ and the standard deviation σ of the training set, and apply the following formula for standardization. Here, y* represents the standardized data. To prevent information leakage, we will use the mean and standard deviation of the training set to standardize both the training and testing sets. Normalization is beneficial for prediction [10].

$$y^{*}\sim=\frac{y-\mu}{\sigma} \tag{1}$$

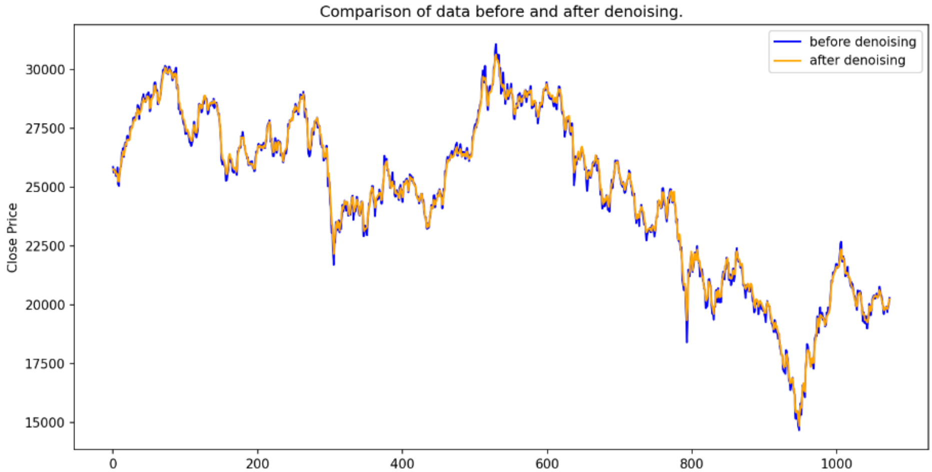

Figure 2. Comparison of data before and after denoising.

3.4. Evaluation metrics

In this experiment, a total of three evaluation metrics has been specified: accuracy, cumulative return, and maximum return. The experiment utilizes the closing prices of stock data and compares the closing price of the current day with that of the next trading day. If the closing price of the next trading day is greater than that of the current day, it is labeled as an increase (1), and if it is lower, it's labeled as a decrease (-1). These labels are used during training and testing.

Accuracy: The number of times the agent selects the option with the maximum reward divided by the total number of selections.

Cumulative Return: When the prediction of stock price movement is correct, the gain of that day is obtained. Conversely, when the prediction is incorrect, the loss of that day is incurred.

$$Cumulative\ Return = \frac{cumulative\ profit}{initial\ investment} * 100\% \tag{2}$$

Maximum Return: The highest value of cumulative profit obtained during model testing.

3.5. Deep reinforcement model

This section will introduce the key details of deep reinforcement learning model.

The deep reinforcement learning model of this experiment utilizes two Long Short-Term Memory (LSTM) networks or two Fully Connected (FC) layers with consistent parameters as the prediction network and the target network. The LSTM consists of 3 layers with 64 nodes each. The final layer is a dense layer with 2 nodes, which outputs the value for the two possible actions that can be taken.

Setting transaction fees to zero, the initial capital for the experiment is 10,000 units. The learning rate is set to 0.001, and the decay rate for each iteration is 0.95. The training process for the agent continues for 500 episodes.

The action design for the model is as follows: The available actions are "buy" and "sell". When the model chooses to "buy", the agent will spend all available cash to purchase stocks. When the model chooses to "sell" and holds stocks, the agent will sell all of its owned stocks.

Reward Value Design for the Model: Based on the agent's selection of different actions, two distinct rewards are designed, enabling the agent to be trained according to the reward values.

The reward for the action "sell" is as follows:

$$R_s = \frac{M_t^c - M_{t+day}^c}{M_t^c} \tag{3}$$

The reward for the action "buy" is as follows:

$$R_b = \frac{M_{t+day}^c - M_t^c}{M_t^c} \tag{4}$$

Where M_t^c represents the closing price of the current day, and M_{t+day}^c represents the closing price of the predicted day, which is day days ahead.

4. Empirical results and analysis

This experiment is divided into two groups: In the first group, a traditional single model is employed for stock market prediction, and its results are compared with those of a hybrid model. In the second group, two hybrid models are utilized for stock market prediction, and the performance of these two models is compared. Finally, the results of both experiments are combined for discussion to validate the effectiveness of utilizing wavelet denoising and reinforcement learning algorithms.

Table 3. The experimental results of LSTM-DQN and FC-DQN model.

Data	Model	Cumulative Return	Accuracy
SSEC	LSTM	2.01%	53.1%
SSEC	FC	-1.07%	52.9%
SSEC	LSTM-DQN	20.12%	61.1%
SSEC	FC-DQN	9.27%	54.2%
HSI	LSTM	-4.76%	50.0%
HSI	FC	-10.29%	48.9%
HSI	LSTM-DQN	15.6%	58.1%
HSI	FC-DQN	6.41%	54.0%

From the table above, it can be observed that using individual models for stock market prediction, where LSTM achieves a prediction accuracy of 53.1% for SSEC, and a prediction accuracy of 50.0% for HSI. On the other hand, FC achieves a prediction accuracy of 52.9% for SSEC, and a prediction accuracy of 48.9% for HSI. The experimental results indicate that the prediction accuracy of LSTM-DQN is 61.1%, the FC-DQN is 54.2%. This suggests that the predictive performance of individual models is generally inferior to that of hybrid models.

From Table 4, denoising with wavelet transform improves cumulative returns, accuracy, and model stability, underscoring the importance of denoising stock data. Comparing LSTM-DQN and FC-DQN models, LSTM-DQN outperforms FC-DQN in handling time series data.

Table 4. The experimental results of LSTM-DQN and FC-DQN model.

Data	Denoise	Model	Cumulative Return	Maximum Return	Accuracy	Recall
SSEC	NO	LSTM-DQN	6.01%	55.1%	52.7%	32.7%
SSEC	YES	LSTM-DQN	20.12%	30.22%	61.1%	11.1%
HSI	NO	LSTM-DQN	14.9%	36.07%	56.1%	22.4%
HSI	YES	LSTM-DQN	15.6%	21.3%	58.1%	10.7%
SSEC	YES	FC-DQN	9.27%	23.7%	54.2%	23.1%
HSI	YES	FC-DQN	6.41%	25.4%	54.0%	15.7%

In Figure 3, denoising enhances LSTM-DQN's returns for SSEC and HSI stocks. SSEC's returns rise from 6.01% to 20.12%, and HSI's improve from 14.9% to 15.6%. This underscores LSTM-DQN's consistent performance. Figure 4 compares LSTM-DQN and FC-DQN models, illustrating LSTM-DQN's superiority due to historical information retention. This aligns with Table 3, reinforcing LSTM's stock prediction outperformance.

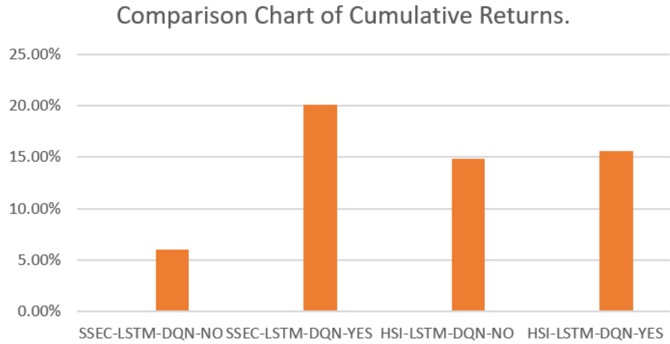

Figure 3. Chart of cumulative returns.

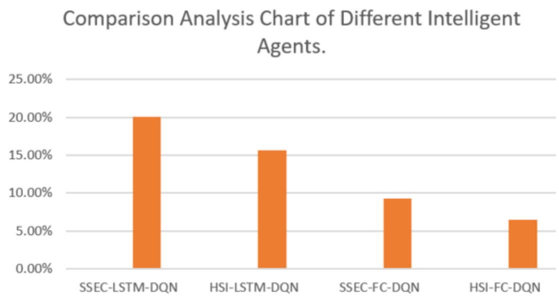

Figure 4. Chart of different intelligent agents.

5. Conclusion

This study employed wavelet transformation for refining stock data, followed by feature engineering and prediction training. A hybrid model, integrating the DQN algorithm, yielded favorable outcomes for training and predicting stock market data. Comparative analysis favored the hybrid model over individual prediction models. Similarly, a comparison between LSTM-DQN and FC-DQN models indicated LSTM-DQN's superiority within the chosen dataset, implying its applicability to stock data prediction.

Nonetheless, the study has limitations. Factors like news and geopolitics weren't considered in stock price analysis. Exploring varied model combinations and neural networks could yield enhanced results. Future research should leverage these findings, incorporating feature selection and multiple neural networks to boost accuracy and profitability in stock market prediction.

References

[1] Henrique, Y. B. M., Sobreiro, V. A., & Kimura, H. (2019). Machine learning techniques applied to financial market prediction. Expert Systems with Applications, 124, 226–251.

[2] Baek, Y., & Kim, H. Y. (2018). ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. Expert Systems with Applications, 113, 457–480.

[3] Ding, G., & Qin, L. (2020). Study on the prediction of stock price based on the associated network model of LSTM. International Journal of Machine Learning and Cybernetics, 11(6), 1307–1317.

[4] Weng, B., Lu, L., Wang, X., Megahed, F. M., & Martinez, W. (2018). Predicting short-term stock prices using ensemble methods and online data sources. Expert Systems with Applications, 112, 258–273.

[5] Barra, S., Carta, S., Corriga, A., Podda, A. S., & Recupero, D. R. (2020). Deep learning and time series-to-image encoding for financial forecasting. IEEE/CAA Journal of Automatica Sinica.

[6] Labate, D., La Foresta, F., Occhiuto, G., Morabito, F. C., Lay-Ekuakille, A., & Vergallo, P. (2013). Empirical mode decomposition vs. wavelet decomposition for the extraction of respiratory signal from single-channel ECG: A comparison. IEEE Sensors Journal, 13(7), 2666–2674.

[7] Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. Expert Systems with Applications, 197, Article 116659.

[8] Liang, X., Luo, L., Hu, S., & Li, Y. (2022). Mapping the knowledge frontiers and evolution of decision making based on agent-based modeling. Knowledge-Based Systems, 250.

[9] Yang, S. G. (2021). A novel study on deep learning framework to predict and analyze the financial time series information. Future Generation Computer Systems, 125, 812–819.

[10] Tang, Z., Zhang, T., Wu, J., Du, X., & Chen, K. (2020). Multistep-ahead stock price forecasting based on secondary decomposition technique and extreme learning machine optimized by the differential evolution algorithm. Mathematical Problems in Engineering, 2020.