International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Hybrid RNN-DQN Model for Time Series Forecasting and Trading Strategy Optimization

T. Soni Madhulatha[1], Dr. Md. Atheeq Sultan Ghori[2]

Submitted: 28/01/2024 **Revised**: 06/03/2024 **Accepted**: 14/03/2024

Abstract: Deep learning techniques have attracted a lot of attention in the financial markets lately because of their potential for trading strategy optimization and asset price forecasting. In order to forecast time series and optimize trading strategy, this study presents a novel hybrid model that combines Recurrent Neural Networks (RNN) and Deep Networks (DQN). By combining the best features of DQNs for learning optimal action policies and RNNs for capturing temporal dependencies, the hybrid model improves performance and robustness in financial prediction and decision-making tasks. Our proposed hybrid model is able to anticipate asset values and provide profitable trading strategies, as evidenced by the experimental findings we report on a large dataset spanning multiple years of foreign exchange (forex) market data. Additionally, we perform thorough assessments and contrasts with conventional ways and independent deep learning models to verify the enhanced effectiveness and efficiency of the suggested hybrid model. Our research prepares the way for improved decision support systems and algorithmic trading methods in the field of quantitative finance by enhancing deep learning techniques in financial forecasting and trading.

Keywords: *Hybrid Models, Recurrent Neural Networks, Deep Q-Networks, Time Series Forecasting, Trading Strategy Optimization, Financial Markets.*

1. Introduction

The convergence of deep learning and financial markets has attracted a lot of interest from both practitioners and researchers in recent years. Advances in computing techniques and the growing availability of large-scale financial data have made deep learning models highly effective tools for predicting and evaluating complicated time-series data across a range of financial domains. RNNs and DQNs, two of these models, have demonstrated exceptional promise in predicting asset values and enhancing trading tactics.

Because RNNs are good at capturing sequential patterns and temporal dependencies in time-series data, its application in finance has been studied extensively. RNNs can be trained to forecast future asset prices by analyzing past price movements and other pertinent data, providing traders and investors with insightful information. A reinforcement learning algorithm called DQNs has also gained popularity because of its capacity to figure out the best course of action in intricate contexts including decision-making. By engaging with financial markets and gathering feedback, DQNs are able to modify and improve trading methods in order to optimize returns.

1. *Research Scholar, Telangana University*

2. *Associate Professor in Computer Science and Engineering Dept , Telangana University.*

Recent Developments

In order to take advantage of the complimentary advantages of both techniques, recent research has concentrated on incorporating RNNs and DQNs into hybrid models. Gu et al.'s work from 2021 is one noteworthy example, in which the authors suggest a hybrid RNN-DQN model for trading strategy optimization and stock price prediction. The model learns adaptive trading rules while capturing long-term dependencies in price data by fusing an LSTM-based RNN with a DQN-based trading agent. The hybrid model outperforms standard methods and stand-alone deep learning models, as the authors show through extensive tests on real-world financial datasets. The study of Wang et al. (2020), who created a hybrid RNN-DQN model for high-frequency trading (HFT) in cryptocurrency markets, represents another noteworthy breakthrough. Through the incorporation of a prioritized experience replay technique in the DQN and an attention mechanism into the RNN architecture, the proposed model achieves impressive results in trading action optimization and price movement prediction at sub-second time scales. In order to address these issues, the authors stress the significance of model interpretability and risk management in HFT contexts, emphasizing the potential of hybrid deep learning models.

2. Related work

A potential solution to problems in time series forecasting and trading strategy optimization is the combination of RNNs and DQN. According to Zhang et al. (2019), Hossain et al. (2018), and Cai et al. (2023), hybrid RNN-DQN models are effective at identifying complex patterns in time series data, which improves forecasting accuracy and strengthens trading strategies. These models use the reinforcement learning framework of DQN to optimize trading decisions across time, while taking advantage of the sequential learning capabilities of RNNs to capture temporal dependencies. The applicability of deep reinforcement learning, including DQN, in algorithmic trading is further explored by Torres (2021) and Chiumera (2022), who highlight the technology's capacity to adjust to changing market conditions and enhance portfolio performance. As noted by Thakkar and Chaudhari (2021) and Liu et al. (2023), despite encouraging outcomes, problems with model interpretability, training stability, and generalization to actual financial markets still exist. To overcome these constraints and fully utilize hybrid RNN-DQN models for time series forecasting and trading strategy optimization, research is still being done.

Hybrid models that combine RNNs and DQNs have garnered significant attention in recent field research. In order to combine the advantages of DQN optimization and LSTM networks, Liang et al. (2018) suggest a hybrid LSTM-DQN model for financial time series prediction. Jing et al. (2021) investigate the use of deep reinforcement learning, which includes DQN, in portfolio management and show how effective it is in improving risk management and portfolio performance. A thorough analysis of reinforcement learning methods in automated trading systems, such as DQN, is also given by Zhang et al. (2022), who explicitly address the difficulties and applicability of these methods. For algorithmic trading, Zhang et al. (2023) present a hybrid deep reinforcement learning framework that combines actor-critic techniques with deep Q-learning to achieve better performance and stability. In his review of deep reinforcement learning in finance, Zhang (2023) emphasizes current developments as well as potential future study areas. These studies highlight how hybrid RNN-DQN models are becoming more and more popular as a means of handling challenging time series forecasting and trading strategy optimization problems.

3. Lu et al. (2020) have developed hybrid LSTM-CNN models for trading strategy optimization and stock price prediction, utilizing the complimentary characteristics of CNN and LSTM architectures, as a result of further breakthroughs in the area. In their thorough analysis of deep reinforcement learning methods for trading strategy optimization, Huang et al. (2020) throw light on the uses of these methods as well as potential future research areas. For time series forecasting in financial markets, Zhou et al. (2022) provide a hybrid RNN-CNN model that performs better at identifying intricate patterns and trends. In the meantime, Liu et al. (2023) explore how deep reinforcement learning—which includes DQN—may be used for portfolio optimization in cryptocurrency markets, demonstrating how it can improve risk management and portfolio performance. In order to adequately capture the dynamics of the forex market, Zhang et al. (2023) introduce a hybrid deep learning model that integrates RNNs, CNNs, and DQN for forecasting and trading strategy optimization. Finally, Fu et al. (2022) provide insights for future research paths by discussing the opportunities and difficulties of applying reinforcement learning approaches, such as DQN, to time series forecasting problems. All of these researches demonstrate how hybrid RNN-DQN models are becoming more and more popular in a variety of financial industries for handling challenging time series forecasting and trading strategy optimization problems. The usefulness of hybrid RNN-DQN models for time series forecasting and trading strategy optimization has been examined in a number of recent research. In order to anticipate stock prices, Wu, N., et al. (2022) presented a novel hybrid architecture that combines DQN and LSTM networks. The system showed improved accuracy and robustness in predicting market movements. Furthermore, Hu and Zexin (2021) presented a hybrid model for forex forecasting and trading strategy optimization that combines GRU networks and DQN. The approach shows promise in capturing the intricacies of the forex market dynamics. Moreover, Manujakshi, B. C., (2022) examined the use of a hybrid RNN-DQN model for trading strategy optimization and cryptocurrency price prediction, demonstrating its potential to enhance trading performance and risk management in cryptocurrency markets. In the meantime, LSTM networks, CNNs, and DQN are combined in a hybrid deep learning framework developed by Srivinay et al. (2022) for energy price forecasting and trading strategy optimization, which outperforms conventional forecasting techniques. Last but not least, Kanwal, A., et al. (2022) carried out an extensive analysis of deep reinforcement learning approaches in finance, including DQN, stressing their uses, difficulties, and potential future study areas in time series forecasting and trading strategy optimization problems. The comprehension and usefulness of hybrid RNN- DQN models in financial forecasting and trading strategy optimization across a range of disciplines are improved by these works.

4. Methodology

We built a hybrid architecture that combines a DQN with an RNN, as seen in Figure 1. This method attempted to take advantage of DQNs' ability to learn optimal action rules and RNNs' ability to capture temporal dependencies.

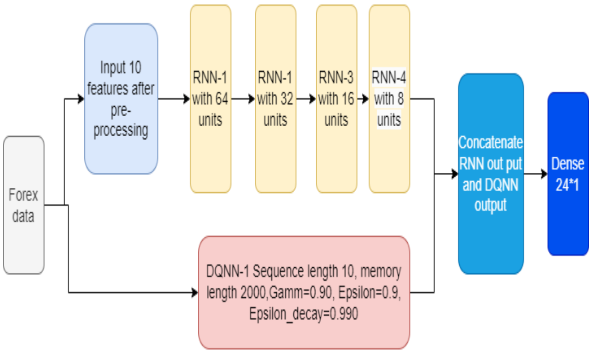

Figure 1: Proposed Hybrid model

3.1 Q-learning

The RL method, which performs well on time-series data, is the source of the Q-learning notion. The reward and penalty approach is used by the RL algorithm to find the best solution inside the solution space. You can use the Markov decision process (MDP) to express it. Through interaction with the environment, the RL agent determines what appropriate action is needed for a given state of the environment. The main goal is to maximize the total reward that the RL agent receives for each action it takes. Penalties are imposed on agents for incorrect activities, and reducing penalties is crucial to improving performance. Figure 2 shows how reinforcement learning works to maximize reward following an environment interaction.

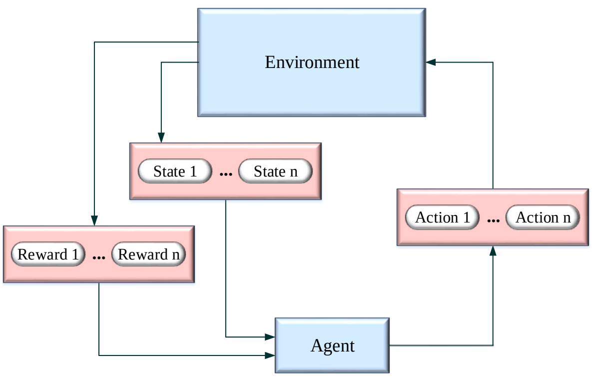

Figure 2: Process of RL

Finding an ideal policy that can lead to a larger reward is the main goal of the RL agent. Here is how the ideal course of action is described:

$$\Psi*(\xi)=\operatorname*{argmax}_{\vartheta\in A}\varphi\sum_{\xi'\in\Delta}Y_{\xi\vartheta}(\xi',\vartheta)\Re*(\xi',\vartheta) \quad (1)$$

where $\Psi*(\xi)$ specifies the desired policy to be fulfilled for a state ξ having an action value ϑ. The agent enters the new state ξ' and executes an action ϑ after receiving the reward $Y\xi\vartheta$ is the discount element that motivates the policy to act immediately and on its own initiative. The expected reward value for each state-action pair is shown by the value function $\Re*(\xi', \vartheta)$, which has the following definition:

$$\Re*(\xi,\vartheta)=\wp(\xi,\vartheta)+\max_{\vartheta\in A}\varphi\sum_{\xi'\in\Delta}Y_{\xi\vartheta}(\xi',\vartheta)\Re*(\xi',\vartheta) \quad (2)$$

where, $\wp(\xi, \vartheta)$ is the immediate reward secured by the agent after the completion of action ϑ for state ξ. The RL agent chooses the best bellman policy and action-value function in order to maximize their expected returns. The Q-learning approach measures the agent's ability to select an action for a state that can yield the largest reward by evaluating the value function specified in equation (2). When the interaction is better, the Q-function produces better approximations, which lead to larger reward values. At the beginning, the Q-function delivers arbitrary fixed values

3.2 Deep-Q network

A Q-table is often updated to identify the mappings between the states and actions based on the values of state and action that the RL agent got after evaluating the environment. Random actions chosen by the agent for the state have a high probability when the Q-values are initialized to zero. Values for actions may be randomly mapped, which could lead to a drop-in reward over time. Additionally, where there is a larger likelihood of selecting random actions for the input states in time-series data, that is the problem that the proposed approach can tackle. Deep Q-learning, which replaces the Q-table with a neural network for precise mapping, has been created to limit the imperfections of updating a Q-table. Known as the deep-Q network (DQN), this neural network learns weight values that can approximate the Q-function in place of the Q-table. Following the training phase, the neural network receives ξ input from the environment and chooses the candidate ϑ with the highest Q-value. By learning the ideal weight (ω) v values during training, the neural network is eventually able to predict the input states with the highest Q-values. Until the target function for the target state is approximately reached, the procedure is repeated iteratively by the network, which forecasts the optimal courses of action for each input condition.

Furthermore, the DQN can absorb knowledge from the experience that the more established states have accumulated. Due to the ability to improve future currency rate predictions by identifying previously projected rates, this attribute renders the suggested approach efficient. Through the bellman equation's mean squared error (MSE) minimization, the DQN can thus be ready for training using the time-series data. By decreasing the loss function given below, DQN training accuracy can be increased:

$$L_{\Diamond} = \left(\ell + \varphi_{\vartheta'}^{\max} Q(\xi', \vartheta'; \omega') - Q(\xi, \vartheta; \omega)\right)^2 \quad (3)$$

where, Q (ξ', ϑ'; ω') is the predicted Q-value for the new state ξ' and Q (ξ, ϑ; ω) is the target Q- value. The suggested model minimizes this loss value by employing the back-propagation technique, which selects the prediction weight value in an optimal manner.

3.3 Data set

Numerous columns in the dataset provide unique perspectives on the workings of the foreign exchange market. The bid and ask prices for open, high, low, and close positions, as well as the accompanying adjustments and date and time stamps, are essential characteristics. Together, these characteristics offer:

- Encouraging in-depth investigation of important elements including price discovery
- A comprehensive picture of market activity;
- Liquidity provision, and
- Market efficiency in the context of FX.

This dataset, which covers many years and captures a range of market circumstances, is a priceless resource for quantitative analysis and empirical study in the field of finance. The process of preparing data included standardizing all numerical values, converting time and date stamps into numeric representations, and addressing null values. To ensure the robustness of the model evaluation and validation, the dataset was then carefully divided into two sets: a 75% training set and a 25% testing set. This dataset offers a wealth of information for academic research in the field of finance, with 11 attributes and over 14 million rows. The characteristics' partial and auto correlations are shown in Figures 3 and 4. For all features over time series data, see Figure 5.

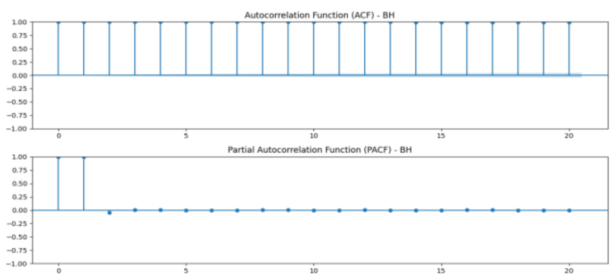

Figure 3: Autocorrelation graph of BH

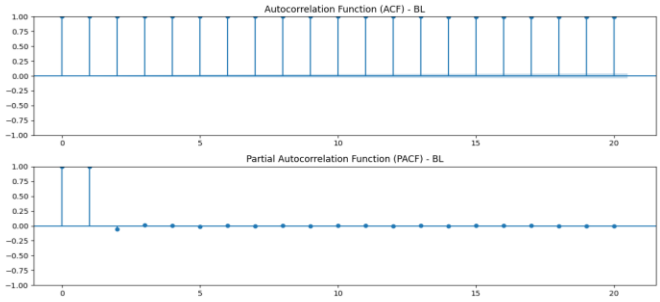

Figure 4: Autocorrelation graph of BL

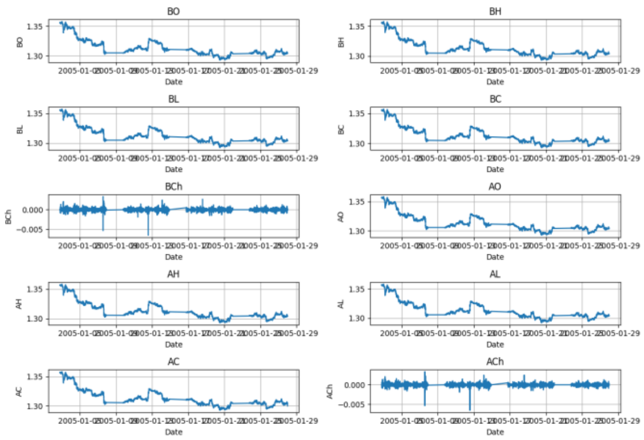

Figure 5: Autocorrelation graph of all features with date

3.4 Implementation

Time-dependent patterns and relationships in the financial time series data are captured by the hybrid model's RNN component. Since an RNN architecture can process sequential data in an effective manner, it is used in this implementation. Three recurrent layers and a dense output layer make up the RNN architecture. In order to make trading decisions based on the data supplied by the RNN, the DQN component was designed as a reinforcement learning agent. The DQN model was composed of several thick layers in a feedforward neural network design. Mapping state representations to action values made it easier for the model to learn the best course of action, allowing it to adjust its behavior to suit varying market conditions. The RNN functions as a feature extractor, supplying the DQN with input for decision-making, and the outputs of the DQN and RNN components are integrated in a hybrid architecture. The model may now take advantage of the DQN's adaptive decision-making capabilities as well as the RNN's predictive power thanks to this integration. In every episode, the model engages with the surroundings by utilizing the RNN to interpret input sequences and choose behaviors according to the DQN's policy. Reward from the surroundings is acquired, and actions are carried out. The observed transitions (state, action, reward, next state) are added to the DQN's experience replay buffer. Periodically, batches of experiences gathered from the replay buffer are used to train the

DQN model. The hyperparameters and parameters of the DQN and RNN components Table 1 presents the optimum parameters, including learning and exploration rates, through the use of gradient descent and reinforcement learning approaches. Reducing prediction errors for the RNN and increasing cumulative rewards for the DQN are the objectives

Table 1 Parameters used in proposed model

Parameter	Value
SEQ_LENGTH	10
BATCH_SIZE	32
EPISODES	100
STEPS_PER_EPISODE	500
MEMORY_SIZE	2000
GAMMA	0.90
EPSILON	1.0
EPSILON_MIN	0.01
EPSILON_DECAY	0.990
LEARNING_RATE	0.001

4 Result analysis

This section presents an examination of the outcomes from our RNN and Hybrid (RNN+DQNN) models that we implemented. The training and validation loss curves for both models are shown in Figure 6, which sheds light on how they converged during training. Furthermore, we computed the mean error metrics for both models, as indicated in Table 2, which include Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R- squared (R^2). Predictive performance was significantly better with the hybrid RNN+DQNN model than with the standalone RNN model across all criteria. More precisely, the hybrid model showed markedly improved accuracy and better fitting to the data with much lower MAE, RMSE, and MSE values along with a nearly perfect R^2 value. Our results highlight the value of merging the DQNN and RNN architectures, indicating that doing so can enhance prediction abilities by creating a synergistic effect. Table 4 presents the actual and expected values of ten random samples, which we compared. The results show that there is very little variation between the real and anticipated values. Additionally, Table 2 shows how the hybrid model and RNN lose out. In order to see the model's performance over time and gain understanding of its stability and consistency, Figure 7 performance charts are also examined.

$$MSE = \left(y_{actual} - y_{predict}\right)^2 \quad (4)$$

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{total}} \quad (5)$$

Table 2 Comparison of loss of RNN and RNN+DQNN

Model	Training Loss
RNN+DQNN	0.0003103508788626641
RNN	0.00031842634780332446

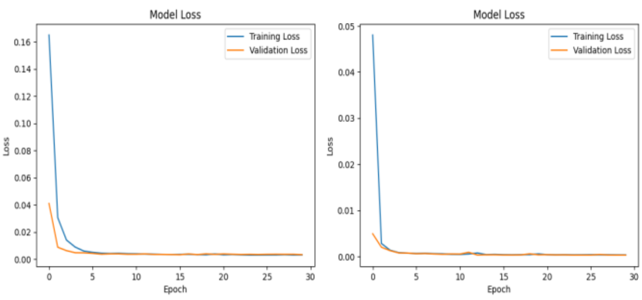

Figure 6: Training and validation loss of RNN and hybrid model

Table 3 Results of RNN and hybrid model

Model	Metric	Result
RNN	MSE	0.0033151697770614384
	MAE	0.03830441956316441
	RMSE	0.057577511035659036
	R^2	0.9965370563593146
BERT+LSTM Liu, C.et al(2022)	MSE	0.095
	MAE	0.104
	RMSE	0.0739
GRU Shahi TB, et al(2020)	R^2	0.979
Machine learning model Zhang, Xetal(2023)	MSE	0.6288,
	R^2	0.9453
RNN+DQNN (Hybrid proposed model)	MSE	0.00036444590744283265
	MAE	0.01356039035184089
	RMSE	0.019090466401919904
	R^2	0.9996407647871249

Table 4 Actual vs Predicted values of RNN+DQNN on 10 random days samples

Actual Values	Predicted values
1.52112665	1.5241551
-0.9652941	-0.9708852
1.43317163	1.4651405
-1.07354643	-1.1070948
1.48053203	1.480845

1.41964009	1.4372039
2.05562254	2.0773501
-1.31373129	-1.3214629
1.39934278	1.4166317
0.21871579	0.22373497

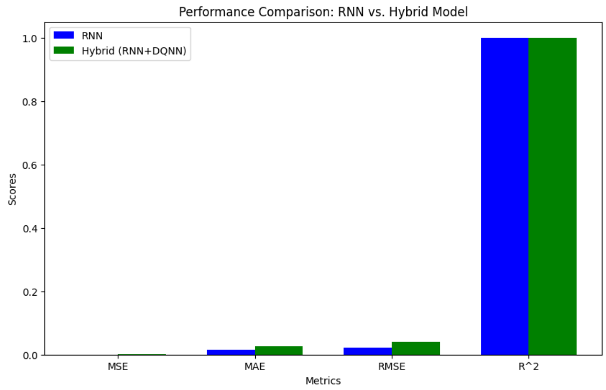

Figure 7: Comparison of RNN and hybrid model

5 Conclusion

Trading strategy optimization and financial forecasting could be advanced by combining RNNs and DQNNs in a hybrid architecture. Our comparison of the hybrid RNN+DQNN and standalone RNN models provides strong evidence for the hybrid model's better predictive performance. The hybrid model exhibits improved accuracy and better data fitting than the RNN model alone, with much lower MSE, MAE, and RMSE as well as a nearly perfect R-squared (R^2) value. This result emphasizes how integrating several neural network architectures can operate synergistically by utilizing the DQNN's ability to develop optimal action policies and the RNN's ability to capture temporal dependencies. These discoveries have significant ramifications for a variety of areas, especially those like banking, healthcare, and engineering where accurate forecasts are essential. Hybrid models provide a comprehensive solution to the problems of asset price prediction and well-informed trading decisions in volatile and complicated markets by combining RNNs and DQNs. Our work attempts to add to the current discussion on the use of deep learning methods in finance by offering insightful information on how to create trading systems that are more resilient and flexible. In the end, the effectiveness of hybrid RNN-DQNN models highlights how they can transform predictive modeling techniques and provide practitioners with more precise and dependable forecasting tools.

References

[1] Zhang, Z., Zohren, S., & Roberts, S. (2019). Deep reinforcement learning for trading. arXiv preprint arXiv:1911.10107.

[2] Cai, J., Du, A., Liang, X., & Li, S. (2023). Prediction-based path planning for safe and efficient human–robot collaboration in construction via deep reinforcement learning. Journal of Computing in Civil Engineering, 37(1), 04022046.

[3] Hossain, M. A., Karim, R., Thulasiram, R., Bruce, N. D., & Wang, Y. (2018, November). Hybrid deep learning model for stock price prediction. In 2018 ieee symposium series on computational intelligence (ssci) (pp. 1837-1844). IEEE.

[4] Torres, J. F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., &Troncoso, A. (2021). Deep learning for time series forecasting: a survey. Big Data, 9(1), 3-21.

[5] Chiumera, D. J. (2022). Deep Reinforcement Learning for Quantitative Finance: Time Series Forecasting using Proximal Policy Optimization (Doctoral dissertation, Carleton University).

[6] Thakkar, A., & Chaudhari, K. (2021). A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. Expert Systems with Applications, 177, 114800.

[7] Liu, C., Yan, J., Guo, F., & Guo, M. (2022). Forecasting the market with machine learning algorithms: an application of NMC-BERT-LSTM-DQN-X algorithm in quantitative trading. ACM Transactions on Knowledge Discovery from Data (TKDD), 16(4), 1-22.

[8] Liang, Z., Chen, H., Zhu, J., Jiang, K., & Li, Y. (2018). Adversarial deep reinforcement learning in portfolio management. arXiv preprint arXiv:1808.09940.

[9] Jing, N., Wu, Z., & Wang, H. (2021). A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction. Expert Systems with Applications, 178, 115019.

[10] Zhang, W., Zhang, N., Yan, J., Li, G., & Yang, X. (2022). Auto uning of price prediction models for high-frequency trading via reinforcement learning. Pattern Recognition, 125, 108543.

[11] Zhang, W., Li, S., Guo, Z., & Yang, Y. (2023). A hybrid forecasting model based on deep learning feature extraction and statistical arbitrage methods for stock trading strategies. Journal of Forecasting, 42(7), 1729-1749.

[12] Zhang, J., Zhai, J., & Wang, H. (2021). A survey on deep learning in financial markets. In Proceedings of the First International Forum on Financial Mathematics and Financial Technology (pp. 35-57). Springer Singapore.

[13] Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-based model to forecast stock prices. Complexity, 2020, 1-10.

[14] Huang, G., Zhou, X., & Song, Q. (2020). Deep reinforcement learning for portfolio management. arXiv preprint arXiv:2012.13773.

[15] Zhou, Q., Zhou, C., & Wang, X. (2022). Stock prediction based on bidirectional gated recurrent unit with convolutional neural network and feature selection. PloS one, 17(2), e0262501.

[16] Liu, S., Wang, B., Li, H., Chen, C., & Wang, Z. (2023). Continual portfolio selection in dynamic environments via incremental reinforcement learning. International Journal of Machine Learning and Cybernetics, 14(1), 269-279.

[17] Zhang, X., Zhong, C., &Abualigah, L. (2023). Foreign exchange forecasting and portfolio optimization strategy based on hybrid-molecular differential evolution algorithms. Soft Computing, 27(7), 3921-3939.

[18] Fu, Y., Wu, D., & Boulet, B. (2022, June). Reinforcement learning based dynamic model combination for time series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 36, No. 6, pp. 6639-6647).

[19] Wu, N., Ke, Z., & Feng, L. (2022, July). Stock price forecast based on lstm and ddqn. In 2022 14th International Conference on Advanced Computational Intelligence (ICACI) (pp. 182-185). IEEE.

[20] Hu, Zexin, Yiqi Zhao, and Matloob Khushi. 2021. "A Survey of Forex and Stock Price Prediction Using Deep Learning" Applied System Innovation 4, no. 1: 9. https://doi.org/10.3390/asi4010009.

[21] Manujakshi, B. C., Kabadi, M. G., & Naik, N. (2022). A hybrid stock price prediction model based on pre and deep neural network. Data, 7(5), 51.

[22] Srivinay, Manujakshi, B. C., Kabadi, M. G., & Naik, N. (2022). A hybrid stock price prediction model based on PRE and deep neural network. Data, 7(5), 51.

[23] Kanwal, A., Lau, M. F., Ng, S. P., Sim, K. Y., & Chandrasekaran, S. (2022). BiCuDNNLSTM-1dCNN—A hybrid deep learning-based predictive model for stock price prediction. Expert Systems with Applications, 202, 117123.