

Article

Not peer-reviewed version

Research on Option Pricing Prediction Based on Deep Learning

[Linnea Whitford](#)*

Posted Date: 28 July 2025

doi: 10.20944/preprints202507.2204.v1

Keywords: Option pricing; Deep learning; Black-Scholes pricing model; BP neural network; LSTM neural network



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Research on Option Pricing Prediction Based on Deep Learning

Linnea Whitford

Eastern Washington University, Cheney, Washington, USA; whitfordlinnea@gmail.com

Abstract

As a financial derivative with high leverage, options have prominent hedging and arbitrage capabilities. Traditional pricing methods for European options, such as the Black–Scholes pricing model, the Merton model, and the Heston model, rely on strict assumptions and simulate option price trends using stochastic processes. However, due to the discrepancies between these assumptions and actual market conditions, these traditional models often fail to reflect real-world option pricing accurately. Therefore, this paper adopts a data-driven approach using deep learning algorithms to simulate the option pricing process. Based on the classical Black–Scholes pricing theory, we explore the feasibility of using BP neural networks and LSTM neural networks for option pricing prediction. Using historical data of the SSE 50ETF options, we build two predictive models and use MSE, MAE, and R-squared as evaluation metrics to assess prediction accuracy. Experimental results show that the LSTM model significantly outperforms others in predicting the price of SSE 50ETF options.

Keywords: option pricing; deep learning; Black–Scholes pricing model; BP neural network; LSTM neural network

1. Introduction

1.1. Research Background and Significance

In the development of financial markets, financial derivatives have gained attention due to their functions in risk transfer and hedging. Options, in particular, are known for their high leverage, nonlinear payoffs, and fat-tailed characteristics, which make them attractive tools for risk management, value discovery, and hedging.

The options market continues to grow rapidly, and theoretical research has deepened over time. Regarding option pricing, Fischer Black and Myron Scholes proposed the first complete Black-Scholes option pricing model in 1973, based on a series of idealized assumptions about market efficiency, which laid the foundation for modern option pricing theory [1]. However, real trading environments often fail to meet these assumptions, resulting in notable deviations when applied in practice.

With the rise of big data, researchers have shifted to deep learning theories driven by historical data. Machine learning and neural networks allow models to be trained on real market data, avoiding rigid assumptions and enabling better pricing accuracy.

1.2. Research Approach and Framework

This study focuses on option pricing. Based on the B-S model, it integrates deep learning theory to construct a predictive model for future option prices.

First, the paper fully analyzes the B-S model and then uses deep learning theory to build neural network models combined with the B-S formula. The model predicts future option prices based on historical data.

Second, BP and LSTM neural networks are employed, and prediction results are evaluated using multiple metrics to assess model effectiveness.

1.3. Literature Review

1.3.1. Option Pricing Models

Early option pricing theory began in 1900, when Louis Bachelier proposed a stochastic model for stock prices based on the “Theory of Speculation”, now considered a cornerstone of modern financial mathematics [2].

In 1973, Black and Scholes proposed the first complete option pricing model under strict assumptions—the B-S model, which became the foundation for later theoretical developments. Merton extended the model to better reflect real market conditions [3–5].

In practical fields such as real asset pricing, Long [6] applied the B-S model to consumer credit pricing, using option theory to replace traditional pricing based on interest rates. Ma [7] introduced a binomial tree method into railway freight pricing, solving volatility issues in fixed contracts.

1.3.2. Deep Learning-Based Option Price Forecasting Models

Hutchinson et al. [8] were among the first to apply neural networks to option pricing using S&P 500 index options. Their trained neural networks showed improved accuracy over the B-S model.

Anders et al. [9] compared B-S and neural models on DAX 30 options, showing neural networks offer better pricing precision. Liu et al. [10] introduced the Self-Attention mechanism to improve option pricing under complex conditions.

2. Related Work

In recent years, deep learning techniques have been extensively adopted in financial data modeling, yielding significant advancements in pricing, prediction, and risk control. Convolutional neural networks (CNNs) have shown utility in financial text analysis and classification tasks, supporting auditing and risk evaluation functions [11]. Reinforcement learning models, particularly those tailored to dynamic market environments, offer novel solutions for risk control and asset management through nested and temporal frameworks [12], while multimodal factor models have demonstrated effectiveness in enhancing stock market forecasting [13]. Federated learning techniques, emphasizing cross-domain data privacy, are particularly relevant in financial systems involving multi-institutional collaboration [14]. Meanwhile, advances in generative diffusion models and UI automation extend the interface and usability spectrum for financial analytics systems [15]. Deep reinforcement learning architectures, such as QTRAN, optimize portfolio strategies under complex market constraints [16], and capsule networks enhance structured data mining through adaptive feature abstraction [17]. For anomaly detection in high-frequency trading, deep learning-based approaches have enabled precise identification of outlier patterns in real time [18], supported further by probabilistic user behavior modeling frameworks based on mixture density networks [19]. Sparse high-dimensional financial datasets are now more tractable with diffusion-transformer hybrid frameworks that improve feature extraction and predictive accuracy [20].

Building on these foundations, the application of recurrent and attention-based models has been expanded in temporal dependency modeling and sequential data interpretation. Hybrid LSTM-GRU networks improve predictive accuracy in credit risk forecasting tasks [21], while ensemble sampling controlled by reinforcement learning enhances the exploration of complex data structures [22]. BiLSTM-CRF models integrated with domain features have shown improved performance in textual boundary detection, contributing indirectly to financial sentiment extraction and interpretation [23]. Reinforcement learning continues to refine structured preference modeling and optimization in financial decision systems [24], as does joint graph convolution with sequential modeling for network traffic forecasting, which finds analogy in financial transaction prediction [25]. Cross-domain forecasting using meta-learned representations supports flexible load and demand modeling in market contexts [26], and diffusion-based unsupervised learning frameworks enhance anomaly detection in structured financial data [27]. Additionally, entity-aware GNNs assist in extracting financial information structures effectively [28], while hybrid LSTM-CNN-transformer architectures improve volatility

prediction across various asset classes [29]. Heterogeneous network learning helps uncover implicit corporate relationships with potential implications for credit and investment assessments [30], and graph-based performance risk models enable proactive analysis in structured financial query systems [31]. Unified encoding and coordination techniques for multi-task learning support the development of efficient financial language models [32], while semantic modeling using capsule networks enhances user intent comprehension in financial dialogue systems [33]. Internal knowledge routing mechanisms and autonomous resource management via reinforcement learning further enable system-level optimization and performance enhancement in financial technology platforms [34,35]. Deep learning also continues to support advanced modeling in financial document analysis and macroeconomic forecasting. Structured text factor integration and dynamic windows have enhanced asset return forecasting models [36], and causal modeling techniques bolster fault detection frameworks in financial cloud services [37]. Transformer-based architectures have enabled contextual risk classification in financial policy documents [38], and federated meta-learning has facilitated distributed failure detection in multi-node infrastructures [39]. Vision-driven transformer models and cross-domain fusion have broadened the reach of deep learning into adjacent forecasting domains [40,41]. Applications of deep Q-networks in cache management and bootstrapped prompting in analogical reasoning further illustrate how reinforcement and transformer techniques are reshaping financial AI [42,43]. Temporal graph representation learning is now foundational in capturing user behavior dynamics, particularly in transactional systems [44], while multi-agent reinforcement learning frameworks serve increasingly in portfolio optimization under high market volatility [45]. Reinforcement learning-based adaptive interfaces and transaction graph-integrated risk monitors signal future directions for user-centered and regulatory applications in financial AI [46,47]. Moreover, transformer-based risk monitoring frameworks that incorporate transaction graph integration have emerged as a robust solution for anti-money laundering tasks, further highlighting the intersection between deep learning and financial compliance [49].

3. Fundamentals of Related Theories

3.1. B-S Option Pricing Theory

The Black-Scholes (B-S) model is designed for pricing European options and is based on the following assumptions:

1. The market is frictionless (no taxes or transaction costs).
2. The market is complete; short selling is allowed, and arbitrage is absent.
3. The price of the underlying follows a geometric Brownian motion with constant return and volatility.
4. The underlying pays no dividends during the option life.
5. The underlying can be traded continuously in any quantity.
6. The risk-free rate is constant and continuous; borrowing and lending rates are equal.

Under these assumptions, the formula for a European call option is:

$$C = SN(d_1) - Xe^{-r(T-t)}N(d_2) \quad (1)$$

where:

$$d_1 = \frac{\ln(S/X) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}},$$

$$d_2 = d_1 - \sigma\sqrt{T-t}$$

Here, C is the call price, S is the current stock price, X is the strike price, r is the risk-free rate, and $N(\cdot)$ is the standard normal CDF.

Using put-call parity, the European put price is:

$$P = Xe^{-r(T-t)}N(-d_2) - SN(-d_1) \quad (2)$$

3.2. Deep Learning Theories

3.2.1. BP Neural Network

The BP network consists of:

1. Random initialization of weights and biases.
2. Forward propagation of inputs.
3. Compute error by comparing output with target.
4. Backpropagate errors through hidden layers.
5. Update weights using gradient descent.
6. Repeat until error falls below threshold.

3.2.2. LSTM Neural Network

The LSTM network enhances RNNs by introducing memory units and gating mechanisms:

1. **Forget gate:** Decides what information to discard from the cell state.
2. **Input gate:** Updates the cell state with new information.
3. **Output gate:** Controls what information is output at each time step.

LSTM effectively captures long-term dependencies and is suitable for time-series prediction tasks such as option pricing.

4. Model Construction for Option Price Prediction

4.1. Sample Selection

4.1.1. Sample Product Description

Based on the favorable characteristics of the CSI 50ETF, this paper selects the 50ETF call option as the main research object. The underlying asset is the CSI 50ETF, which is an ETF tracking the SSE 50 Index, composed of 50 constituent stocks.

4.1.2. Sample Data Range

The data covers the period from January 1, 2022 to December 31, 2022. After removing holidays and missing data caused by trading suspensions, a total of 13,771 sets of data were collected from the Wind database. Of these, 11,657 samples from January 1, 2022 to September 30, 2022 were used for training, and 2,114 samples from October 10, 2022 to December 31, 2022 were used for testing.

4.2. Variable Definition

4.2.1. Input and Output Variables

According to the B-S pricing formula, we selected five known variables: remaining time to maturity, current price of the underlying, strike price, risk-free rate, and historical volatility (σ , denoted as HV). We also included four trading indicators: open price, settlement price (SP), volume (deal), and open interest (host). In total, 9 input variables were used. The output variable is the option's closing price P .

4.2.2. Data Preprocessing

- **Time Conversion:** Remaining time to maturity T is converted to years by:

$$T = \frac{\Delta T}{242}$$

where ΔT is the number of trading days.

- **Normalization:** All input data, including risk-free rate and HV, are standardized using z-score normalization.

4.3. Option Price Prediction Model Construction

This paper uses Python 3.8.0 to build neural network models.

4.3.1. BP Neural Network Model

(1) Network Design and Parameters:

A 3-layer BP neural network is constructed using `tansig` and `purelin` as activation functions. The training function is `traingdm`, with MSE as the loss function. Max allowable error is set to 0.01, training iterations to 100, and learning rate to 0.01.

(2) Prediction Results Analysis:

The predicted option prices using BP neural network are shown in Figure 1.

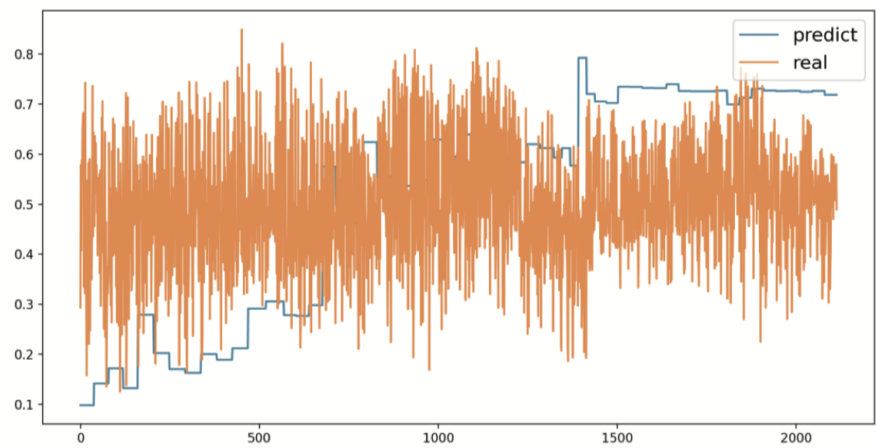


Figure 1. Option Price Prediction Results Using BP Neural Network.

Partial normalized prediction results are shown in Table 1.

Table 1. Partial Normalized Prediction Results (BP Network).

Index	True Price	BP Predicted Price
1	0.459045	0.651990
2	0.262803	0.329410
3	0.150352	0.112909
4	0.012373	0.026878
5	0.028034	0.097869

4.3.2. LSTM Neural Network Model

(1) Network Design and Parameters:

Data is split the same as above. Activation functions for the three gates are `sigmoid`, and `tanh` is used for hidden and output states. MSE is the loss function. RMSProp is the optimizer with 30 epochs and batch size set to 512.

(2) Prediction Results Analysis:

The prediction results using the LSTM model are shown in Figure 2.

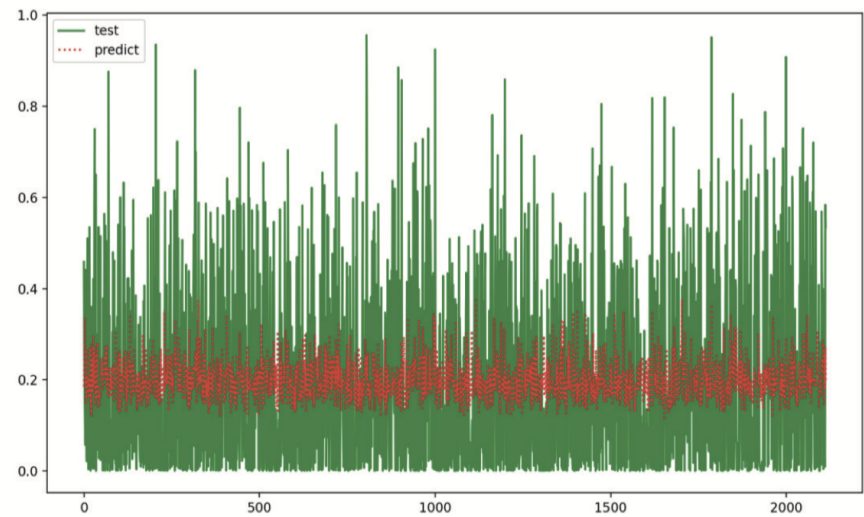


Figure 2. Option Price Prediction Results Using LSTM Model.

Partial normalized prediction results are shown in Table 2.

Table 2. Partial Normalized Prediction Results (LSTM Model).

Index	True Price	LSTM Predicted Price
1	0.459045	0.485363
2	0.262803	0.236087
3	0.150352	0.136311
4	0.012373	0.014152
5	0.028034	0.069612

4.4. Experimental Analysis and Comparison

To evaluate both models, we use three performance metrics:

1. Mean Squared Error (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

2. Mean Absolute Error (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

3. R-Squared (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Table 3 shows the comparison results:

Table 3. Comparison of BP and LSTM Prediction Results.

Metric	BP Network	LSTM Network
MSE	0.088528	0.035709
MAE	0.372881	0.053793
R^2	0.6409727	0.8516187

As seen from the table, the LSTM model demonstrates significantly higher accuracy than the BP model. It has better nonlinear learning capabilities and captures the high-dimensional features of financial data more effectively, making it better suited for predicting 50ETF option prices.

5. Conclusion

5.1. Summary of Findings

This paper applied BP neural networks and LSTM neural networks to build option price prediction models using CSI 50ETF option data. The main conclusions are as follows:

As a data-driven option pricing model, the LSTM neural network achieved higher prediction accuracy than the BP neural network. Among the three evaluation metrics, the LSTM model showed significant advantages in terms of prediction error. Due to its unique architecture and strong nonlinear computational capacity, the LSTM model can accurately capture option prices when trained on large-scale datasets.

5.2. Limitations and Future Research

This study has explored deep learning-based option pricing prediction and yielded promising results. However, there are still several limitations that suggest avenues for future research:

1. Beyond the 9 input variables selected in this paper, there may be other influential features worth exploring, which could improve the predictive power of future models.
2. The correlations among the 9 selected input features were not analyzed in depth. Future studies may investigate inter-variable correlations and their impact on model performance.
3. The model uses historical volatility from the B-S formula as one of the input features. According to prior studies, using implied volatility instead of historical volatility may further reduce model error.

References

1. Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3), 637–654.
2. Bachelier, L. (1900). Théorie de la spéculation. *Annales Scientifiques de l'École Normale Supérieure*, 17(1), 21–86.
3. Merton, R. C. (1973). Theory of rational option pricing. *The Bell Journal of Economics and Management Science*, 4(1), 141–183.
4. Merton, R. C. (1974). On the pricing of corporate debt: The risk structure of interest rates. *The Journal of Finance*, 29(2), 449–470.
5. Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1–2), 125–144.
6. Guo, J., Xie, Z., & Li, Q. (2020). Stackelberg game model of railway freight pricing based on option theory. *Discrete Dynamics in Nature and Society*, Article 6436729. <https://doi.org/10.1155/2020/6436729>
7. Zhang, X., Zhang, P., & Jiang, C. (2022). Option pricing and capacity allocation of China Railway Express considering capital constraint and spot market. In *Proceedings of [Conference]*. <https://doi.org/10.1061/9780784479384.008>
8. Hutchinson, J., Lo, A., & Poggio, T. (1994). A nonparametric approach to pricing and hedging derivative securities via learning. *The Journal of Finance*, 49, 851–889.
9. Anders, U., Korn, O., & Schmitt, C. (1998). Improving the pricing of options: A neural network approach. *Journal of Forecasting*, 17(5–6), 369–388.
10. Liu, J., Yang, Q., Xu, Y., et al. (2021). Attention-augmented neural network for option pricing. *Neurocomputing*, 454, 1–9.
11. Du, X. (2025, February). Financial text analysis using 1D-CNN: Risk classification and auditing support. In *Proceedings of the 2025 International Conference on Artificial Intelligence and Computational Intelligence* (pp. 515–520).
12. Yao, Y. (2025). Time-series nested reinforcement learning for dynamic risk control in nonlinear financial markets. *Transactions on Computational and Scientific Methods*, 5(1).
13. Liu, J. (2025). Multimodal data-driven factor models for stock market forecasting. *Journal of Computer Technology and Software*, 4(2).
14. Zhang, Y., Liu, J., Wang, J., Dai, L., Guo, F., & Cai, G. (2025). Federated learning for cross-domain data privacy: A distributed approach to secure collaboration. *arXiv preprint arXiv:2504.00282*.

15. Duan, Y., Yang, L., Zhang, T., Song, Z., & Shao, F. (2025, March). Automated UI interface generation via diffusion models: Enhancing personalization and efficiency. In *2025 4th ISCAIT* (pp. 780–783). IEEE.
16. Xu, Z., Bao, Q., Wang, Y., Feng, H., Du, J., & Sha, Q. (2025). Reinforcement learning in finance: QTRAN for portfolio optimization. *Journal of Computer Technology and Software*, 4(3).
17. Lou, Y. (2024). Capsule network-based AI model for structured data mining with adaptive feature representation. *Transactions on Computational and Scientific Methods*, 4(9).
18. Bao, Q., Wang, J., Gong, H., Zhang, Y., Guo, X., & Feng, H. (2025, March). A deep learning approach to anomaly detection in high-frequency trading data. In *2025 ISCAIT* (pp. 287–291). IEEE.
19. Dai, L., Zhu, W., Quan, X., Meng, R., Chai, S., & Wang, Y. (2025). Deep probabilistic modeling of user behavior for anomaly detection via mixture density networks. *arXiv preprint arXiv:2505.08220*.
20. Cui, W., & Liang, A. (2025). Diffusion-transformer framework for deep mining of high-dimensional sparse data. *Journal of Computer Technology and Software*, 4(4).
21. Sheng, Y. (2024). Temporal dependency modeling in loan default prediction with hybrid LSTM-GRU architecture. *Transactions on Computational and Scientific Methods*, 4(8).
22. Liu, J. (2025). Reinforcement learning-controlled subspace ensemble sampling for complex data structures.
23. Zhao, Y., Zhang, W., Cheng, Y., Tian, Y., & Wei, Z. (2025). Entity boundary detection in social texts using BiLSTM-CRF with integrated social features.
24. Zhu, L., Guo, F., Cai, G., & Ma, Y. (2025). Structured preference modeling for reinforcement learning-based fine-tuning of large models. *Journal of Computer Technology and Software*, 4(4).
25. Jiang, N., Zhu, W., Han, X., Huang, W., & Sun, Y. (2025). Joint graph convolution and sequential modeling for scalable network traffic estimation. *arXiv preprint arXiv:2505.07674*.
26. Yang, T. (2024). Transferable load forecasting and scheduling via meta-learned task representations. *Journal of Computer Technology and Software*, 3(8).
27. Xin, H., & Pan, R. (2025). Unsupervised anomaly detection in structured data using structure-aware diffusion mechanisms. *Journal of Computer Science and Software Applications*, 5(5).
28. Wang, Y. (2024). Entity-aware graph neural modeling for structured information extraction in the financial domain. *Transactions on Computational and Scientific Methods*, 4(9).
29. Sha, Q. (2024). Hybrid deep learning for financial volatility forecasting: An LSTM-CNN-transformer model. *Transactions on Computational and Scientific Methods*, 4(11).
30. Liu, Z., & Zhang, Z. (2024). Graph-based discovery of implicit corporate relationships using heterogeneous network learning. *Journal of Computer Technology and Software*, 3(7).
31. Gao, D. (2025). Deep graph modeling for performance risk detection in structured data queries. *Journal of Computer Technology and Software*, 4(5).
32. Zhang, W., Xu, Z., Tian, Y., Wu, Y., Wang, M., & Meng, X. (2025). Unified instruction encoding and gradient coordination for multi-task language models.
33. Wang, S., Zhuang, Y., Zhang, R., & Song, Z. (2025). Capsule network-based semantic intent modeling for human-computer interaction. *arXiv preprint arXiv:2507.00540*.
34. Wu, Q. (2024). Internal knowledge adaptation in LLMs with consistency-constrained dynamic routing. *Transactions on Computational and Scientific Methods*, 4(5).
35. Zou, Y., Qi, N., Deng, Y., Xue, Z., Gong, M., & Zhang, W. (2025). Autonomous resource management in microservice systems via reinforcement learning. *arXiv preprint arXiv:2507.12879*.
36. Su, X. (2024). Forecasting asset returns with structured text factors and dynamic time windows. *Transactions on Computational and Scientific Methods*, 4(6).
37. Wang, H. (2024). Causal discriminative modeling for robust cloud service fault detection. *Journal of Computer Technology and Software*, 3(7).
38. Qin, Y. (2024). Deep contextual risk classification in financial policy documents using transformer architecture. *Journal of Computer Technology and Software*, 3(8).
39. Wei, M. (2024). Federated meta-learning for node-level failure detection in heterogeneous distributed systems. *Journal of Computer Technology and Software*, 3(8).
40. Cui, W. (2024). Vision-oriented multi-object tracking via transformer-based temporal and attention modeling. *Transactions on Computational and Scientific Methods*, 4(11).
41. Lin, Y., & Xue, P. (2025). Multi-task learning for macroeconomic forecasting based on cross-domain data fusion. *Journal of Computer Technology and Software*, 4(6).
42. Sun, Y., Meng, R., Zhang, R., Wu, Q., & Wang, H. (2025). A deep Q-network approach to intelligent cache management in dynamic backend environments.

43. Xing, Y. (2024). Bootstrapped structural prompting for analogical reasoning in pretrained language models. *Transactions on Computational and Scientific Methods*, 4(11).
44. Liu, X., Xu, Q., Ma, K., Qin, Y., & Xu, Z. (2025). Temporal graph representation learning for evolving user behavior in transactional networks.
45. Fang, Z., Deng, Y., & Duan, Y. (2025). Dynamic portfolio optimization using multi-agent reinforcement learning in volatile markets.
46. Sun, Q., Xue, Y., & Song, Z. (2024). Adaptive user interface generation through reinforcement learning. *arXiv preprint arXiv:2412.16837*.
47. Jiang, N., Zhu, W., Han, X., Huang, W., & Sun, Y. (2025). Joint graph convolution and sequential modeling.
48. Xin, H., & Pan, R. (2025). Structure-aware diffusion mechanisms in anomaly detection.
49. Wu, Y., Qin, Y., Su, X., & Lin, Y. (2025). Transformer-based risk monitoring for anti-money laundering with transaction graph integration.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.